JUNIPER NETWORKS | Engineering Simplicity

IPsec VPN User Guide

Published
2026-09-08

JUNOS

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

IPsec VPN User Guide
Copyright © 2026 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | xviii

1 **IPsec Fundamentals**

Internet Key Exchange | 2

- Introduction to IKE | **2**
- IKE Versions | **3**
- Interaction Between IKE and IPSec | **3**
- IKEv1 Message Exchange | **4**
- Phase 1 of IKE Tunnel Negotiation | **4**
- Phase 2 of IKE Tunnel Negotiation | **6**
- IKEv2 Message Exchange | **7**
- Proxy ID | **9**
- Traffic Selectors | **9**
- IKE Authentication (Preshared Key and Certificate-Based Authentication) | **9**
- Network Address Translation-Traversal (NAT-T) | **10**
- Suite B and PRIME Cryptographic Suites | **11**

IPsec Basics | 12

- IPsec Overview | **12**
- IPsec Key Management | **14**
- IPsec Security Protocols | **16**
- IPsec Tunnel Negotiation | **18**
- Supported IPsec and IKE Standards | **19**
- Platform-Specific IPsec Tunnel Behavior | **21**
- Additional Platform Information | **21**

2

IPsec VPN in Junos OS

IKE for IPsec VPN | 24

IKE and IPsec Packet Processing | **25**

Introduction to IKE in Junos OS | **31**

IKE Proposal | **36**

IKE Policy | **37**

Rekeying and Reauthentication | **37**

IKE Authentication (Certificate-Based Authentication) | **39**

Signature Authentication in IKEv2 | **42**

IKE Protection from DDoS Attacks | **43**

Configure Protection Against IKE DDoS Attacks | **47**

- Configure the IKE Session for Half Open IKE SAs | **48**
- Configure the IKE Session for Full Open IKE SAs | **50**
- Configure the IKE Session Blocklists | **51**

Example: Configuring a Device for Peer Certificate Chain Validation | **54**

- Requirements | **54**
- Overview | **55**
- Configuration | **56**
- Verification | **64**
- IKE and IPsec SA Failure for a Revoked Certificate | **65**

IKEv2 Fragmentation | **67**

IKE Policy with Local Certificate | **68**

IKE Policy with a Trusted CA | **72**

IKE Policy with Trusted CA Group | **74**

Configuring Establish-Tunnel Responder-only in IKE | **77**

Platform-Specific IKEv2 Responder Only Behavior | **78**

IPsec VPN Overview | 79

- IPsec VPN Topologies on SRX Series Firewalls | **81**

Understanding IKE and IPsec Packet Processing | **81**

Distribution of IKE and IPsec Sessions Across SPUs | **84**

VPN Support for Inserting Services Processing Cards | **86**

IPsec VPN with iked Process | **87**

Cryptographic Acceleration Support | **89**

Routing Protocols Support on IPsec VPN Tunnels | **90**

Anti-Replay Window | **90**

Understanding Hub-and-Spoke VPNs | **91**

Platform-Specific IPsec VPN Behavior | **92**

Additional Platform Information | **94**

Inline IPsec | 103

3 **VPN Configuration Overview**

IPsec VPN Configuration Overview | 109

IPsec VPN with Autokey IKE Configuration Overview | **110**

Recommended Configuration Options for Site-to-Site VPN with Static IP Addresses | **111**

Recommended Configuration Options for Site-to-Site or Dialup VPNs with Dynamic IP Addresses | **112**

Understanding IPsec VPNs with Dynamic Endpoints | **114**

Understanding IKE Identity Configuration | **116**

Configuring Remote IKE IDs for Site-to-Site VPNs | **118**

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | **118**

Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Firewall | **120**

Requirements | **121**

Overview | **121**

Configuration | **122**

Verification | **126**

Configuring IPsec VPN Using the VPN Wizard | **127**

Chassis Cluster HA Control Link Encryption | 128

Configure IPsec VPN with OCSP for Certificate Revocation Status | 131

- Requirements | **132**
- Overview | **132**
- Configuration | **135**
- Verification | **146**

IPv6 IPsec VPNs | 153

- VPN Feature Support for IPv6 Addresses | **153**
- Understanding IPv6 IKE and IPsec Packet Processing | **159**
- IPv6 IPsec Configuration Overview | **166**
- Example: Configuring an IPv6 IPsec Manual VPN | **166**
  - Requirements | **167**
  - Overview | **167**
  - Configuration | **167**
  - Verification | **170**
- Platform-Specific IPv6 Tunnels Behavior | **170**
- Additional Platform Information | **171**

Quantum Safe IPsec VPN | 172

- Quantum Security Overview | **173**
- Junos Key Manager Overview | **173**
- Use Key Profile for Quantum Safe IPsec VPN | **174**
- Quantum Key Distribution | **174**
- Configure Static Key Profile for Junos Key Manager | **176**
- Example: Configure Static Keys Profile for Site-To-Site VPN | **178**
  - Example Prerequisites | **179**
  - Before You Begin | **179**
  - Functional Overview | **180**
  - Topology Overview | **183**

Topology Illustration | **184**

Step-By-Step Configuration on Firewall Devices | **184**

Verification | **187**

Appendix 1: Set Commands on all Devices | **192**

Appendix 2: Show Configuration Output on DUT | **195**

Example: Configure Static Keys Profile for AutoVPN | **203**

Example Prerequisites | **204**

Before You Begin | **204**

Functional Overview | **205**

Topology Overview | **209**

Topology Illustration | **211**

Step-By-Step Configuration on Hub | **211**

Step-By-Step Configuration on Spoke Devices | **214**

Verification | **217**

Appendix 1: Set Commands on all Devices | **226**

Appendix 2: Show Configuration Output on DUT | **230**

Configure Quantum Key Manager Key Profile for Junos Key Manager | **243**

Example: Configure Quantum Key Manager Key Profile for Site-To-Site IPsec VPN | **247**

Example Prerequisites | **248**

Before You Begin | **248**

Functional Overview | **249**

Topology Overview | **253**

Topology Illustration | **255**

Step-By-Step Configuration on Devices | **255**

Verification | **258**

Appendix 1: Set Commands on all Devices | **263**

Appendix 2: Show Configuration Output on DUT | **266**

Example: Configure Quantum-Secured IPsec AutoVPN Topology Using Quantum Key Manager Key Profile | **274**

Example Prerequisites | **275**

Before You Begin | **276**

Functional Overview | **276**

Topology Overview | **281**

Topology Illustration | **284**

Step-By-Step Configuration on Hub | **284**

Step-By-Step Configuration on Spoke Devices | **288**

Verification | **291**

Appendix 1: Set Commands on all Devices | **301**

Appendix 2: Show Configuration Output on DUT | **305**

4 **Route Based VPN**

Route-Based IPsec VPNs | 320

Understanding Route-Based IPsec VPNs | **320**

Example: Configuring a Route-Based VPN | **321**

Requirements | **321**

Overview | **322**

Configuration | **325**

Verification | **340**

Route-Based VPN with IKEv2 | 347

Example: Configuring a Route-Based VPN for IKEv2 | **348**

Requirements | **349**

Overview | **349**

Configuration | **353**

Verification | **368**

Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload | **374**

Requirements | **374**

Overview | **374**

Configuration | **380**

Verification | **401**

IKE Policy with a Trusted CA | **407**

Secure Tunnel Interface in a Virtual Router | 409

Understanding Virtual Router Support for Route-Based VPNs | **409**

Example: Configuring an st0 Interface in a Virtual Router | **411**

Requirements | **411**

Overview | **411**

Configuration | **414**

Verification | 420

Dual Stack Tunnels over an External Interface | 421

Understanding VPN Tunnel Modes | 422

Example: Configuring Dual-Stack Tunnels over an External Interface | 425

Requirements | 426

Overview | 426

Configuration | 430

Verification | 436

IPsec VPN Tunnels with Chassis Clusters | 440

Understanding Dual Active-Backup IPsec VPN Chassis Clusters | 440

Example: Configuring Redundancy Groups for Loopback Interfaces | 442

Requirements | 443

Overview | 443

Configuration | 445

Verification | 449

Traffic Selectors in Route-Based VPNs | 451

Understand Traffic Selectors in Route-Based VPNs | 451

Example: Configure Traffic Selectors in a Route-Based VPN | 458

Requirements | 459

Overview | 459

Configuration | 460

Verification | 475

Platform-Specific ARI for Traffic Selectors Behavior | 478

Migrate to Route-Based IPsec VPNs | 480

Comparison of Policy-Based VPNs and Route-Based VPNs | 480

Shared Point-to-Point st0 Interface | 483

Migrate Policy-Based VPNs to Route-Based VPNs | 485

5 **Class-of-Service Based VPN**

CoS-Based IPsec VPNs | 490

Understand CoS-Based IPsec VPNs with Multiple IPsec SAs | 490

Understand Traffic Selectors and CoS-Based IPsec VPNs | **493**

Example: Configure CoS-Based IPsec VPNs | **496**

- Requirements | **496**
- Overview | **497**
- Configuration | **500**
- Verification | **521**

Understand CoS Support on st0 Interfaces | **525**

Platform-Specific CoS-Based IPsec VPN Behavior | **527**

6 **NAT-T**

Route-Based VPNs with NAT-T | 530

Understanding NAT-T | **530**

Example: Configuring a Route-Based VPN with the Responder behind a NAT Device | **531**

- Requirements | **532**
- Overview | **532**
- Configuration | **538**
- Verification | **557**

Example: Configuring NAT-T with Dynamic Endpoint VPN | **568**

- Requirements | **568**
- Overview | **568**
- Configuration | **570**
- Verification | **586**

Platform-Specific NAT-T with IPsec VPN Behavior | **590**

7 **Group VPN**

Group VPNv1 | 593

Group VPNv1 Overview | **594**

Group VPNv1 Configuration Overview | **603**

Understanding IKE Phase 1 Configuration for Group VPNv1 | **604**

Understanding IPsec SA Configuration for Group VPNv1 | **604**

Understanding Dynamic Policies for Group VPNv1 | **605**

Understanding Antireplay for Group VPNv1 | **606**

Example: Configuring Group VPNv1 Server and Members | **607**

- Requirements | **607**
- Overview | **607**
- Configuration | **608**
- Verification | **633**

Example: Configuring Group VPNv1 Server-Member Communication for Unicast Rekey Messages | **636**

- Requirements | **636**
- Overview | **636**
- Configuration | **637**
- Verification | **637**

Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages | **638**

- Requirements | **638**
- Overview | **638**
- Configuration | **639**
- Verification | **641**

Example: Configuring Group VPNv1 with Server-Member Colocation | **641**

- Requirements | **642**
- Overview | **642**
- Configuration | **643**
- Verification | **653**

Group VPNv2 | 655

Group VPNv2 Overview | **655**

Group VPNv2 Configuration Overview | **661**

Understanding IKE Phase 1 Configuration for Group VPNv2 | **662**

Understanding IPsec SA Configuration for Group VPNv2 | **663**

Understanding Group VPNv2 Traffic Steering | **663**

Understanding the Group VPNv2 Recovery Probe Process | **665**

Understanding Group VPNv2 Antireplay | **666**

Example: Configuring a Group VPNv2 Server and Members | **666**

- Requirements | **667**
- Overview | **667**
- Configuration | **668**
- Verification | **704**

Example: Configuring Group VPNv2 Server-Member Communication for Unicast Rekey Messages | **713**

- Requirements | **713**
- Overview | **713**
- Configuration | **714**
- Verification | **714**

Group VPNv2 Server Clusters | 715

Understanding Group VPNv2 Server Clusters | **716**

Understanding Group VPNv2 Server Cluster Limitations | **720**

Understanding Group VPNv2 Server Cluster Messages | **721**

Understanding Configuration Changes with Group VPNv2 Server Clusters | **723**

Migrating a Standalone Group VPNv2 Server to a Group VPNv2 Server Cluster | **727**

Example: Configuring a Group VPNv2 Server Cluster and Members | **728**

- Requirements | **728**
- Overview | **729**
- Configuration | **731**
- Verification | **800**

8 **ADVPN**

Auto Discovery VPNs | 815

Understanding Auto Discovery VPN | **815**

Understanding Traffic Routing with Shortcut Tunnels | **823**

Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels | **825**

- Requirements | **826**
- Overview | **826**
- Configuration | **831**
- Verification | **856**

Example: Configuring ADVPN with OSPFv3 for IPv6 Traffic | **879**

- Requirements | **879**
- Overview | **880**
- Configuration | **883**
- Verification | **912**

Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established | **915**

Platform-Specific Multicast in ADVPN Behavior | **916**

9 **AutoVPN**

AutoVPN on Hub-And-Spoke Devices | 919

Understanding AutoVPN | **920**

Understanding Spoke Authentication in AutoVPN Deployments | **925**

AutoVPN Configuration Overview | **928**

Example: Configuring Basic AutoVPN with iBGP | **929**

- Requirements | **930**
- Overview | **930**
- Configuration | **934**
- Verification | **962**

Example: Configuring Basic AutoVPN with iBGP for IPv6 Traffic | **966**

- Requirements | **966**
- Overview | **967**
- Configuration | **970**
- Verification | **1001**

Example: Configuring AutoVPN with iBGP and ECMP | **1004**

- Requirements | **1004**
- Overview | **1005**
- Configuration | **1008**
- Verification | **1033**

Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels | **1037**

- Requirements | **1038**
- Overview | **1038**
- Configuration | **1042**

Verification | **1067**

Example: Configuring Basic AutoVPN with OSPF | **1074**

Requirements | **1074**

Overview | **1075**

Configuration | **1078**

Verification | **1104**

Example: Configuring AutoVPN with OSPFv3 for IPv6 Traffic | **1107**

Requirements | **1108**

Overview | **1108**

Configuration | **1112**

Verification | **1140**

Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors | **1144**

Requirements | **1144**

Overview | **1145**

Configuration | **1148**

Verification | **1161**

Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors | **1165**

Requirements | **1166**

Overview | **1166**

Configuration | **1169**

Verification | **1190**

Example: Configuring AutoVPN with Pre-Shared Key | **1193**

Configure Multicast Support on P2MP Infrastructure | **1198**

Platform-Specific AutoVPN Behavior | **1199**

Additional Platform Information | **1200**

10 **Remote Access VPN**

Juniper Secure Connect | 1204

11 **Monitoring VPN**

VPN Monitoring Overview | 1208

VPN Monitoring Methods | 1209

IPsec Datapath Verification | **1210**

Dead Peer Detection | **1212**

VPN Tunnel Monitoring | **1215**

Configure Dead Peer Detection | **1216**

Configure VPN Tunnel Monitoring | **1217**

Platform-Specific VPN Monitoring Behavior | **1219**

VPN Alarms, Audits, and Events | 1219

VPN Alarms and Audits | **1220**

VPN Tunnel Events | **1222**

Configure VPN Alarms | **1223**

Example: Configure an Audible Alert Notification | **1224**

Requirements | **1224**

Overview | **1224**

Configuration | **1224**

Verification | **1225**

Example: Configure Security Alarms Generation | **1225**

Requirements | **1226**

Overview | **1226**

Configuration | **1226**

Verification | **1229**

Platform-Specific VPN Alarms Behavior | **1230**

12 **Performance Tuning**

VPN Session Affinity | 1232

Understanding VPN Session Affinity | **1232**

Enabling VPN Session Affinity | **1234**

Accelerating the IPsec VPN Traffic Performance | **1236**

IPsec Distribution Profile | **1238**

Understanding the Loopback Interface for a High Availability VPN | **1239**

Platform-Specific High Availability VPN Loopback Interface Behavior | **1239**

PowerMode IPsec | 1241

Improving IPsec Performance with PowerMode IPsec | **1241**

Example: Configuring Behavior Aggregate Classifier in PMI | **1247**

Requirements | **1247**

Overview | **1248**

Configuration | **1248**

Verification | **1251**

Example: Configuring Behavior Aggregate Classifier in PMI for vSRX Virtual Firewall Instances | **1252**

Requirements | **1252**

Overview | **1253**

Configuration | **1254**

Verification | **1258**

Example: Configuring and Applying a Firewall Filter for a Multifield Classifier in PMI | **1259**

Requirements | **1259**

Overview | **1260**

Configuration | **1260**

Verification | **1265**

Example: Configuring and Applying Rewrite Rules on a Security Device in PMI | **1266**

Requirements | **1266**

Overview | **1267**

Configuration | **1267**

Verification | **1270**

Configure IPsec ESP Authentication-only Mode in PMI | **1271**

Platform-Specific PMI Behavior | **1272**

Additional Platform Information | **1273**

13 **Troubleshooting**

Troubleshoot a Flapping VPN Tunnel | 1277

Troubleshoot a VPN That Is Up But Not Passing Traffic | 1280

Troubleshoot a VPN Tunnel That is Down | 1284

Troubleshoot Issues in Quantum Safe IPsec VPN | 1286

Troubleshooting Summary | **1287**

Troubleshoot CA Certificate Load Failure Due to Multiple Certificates in a PEM File | **1289**

Troubleshoot QKD Key Retrieval Failure Due to Issues in Certificate Chain Validation | **1291**

Troubleshoot Key Retrieval Failure Due to SAE Mismatch | **1294**

Troubleshoot QKD Key Retrieval Failure Due to KME Server Error | **1296**

Other Common Issues in Quantum Safe IPsec VPN | **1298**

How to Analyze IKE Phase 2 VPN Status Messages | 1300

14 **Configuration Statements and Operational Commands**

Junos CLI Reference Overview | 1306

About This Guide

Use this guide to configure, monitor, and manage the IPsec VPN feature on Junos OS devices to enable secure communications across a public WAN such as the Internet.

Documentation Conventions

For information about the notice icons and text and syntax conventions used in the documentation, see Documentation Conventions.

RELATED DOCUMENTATION

- Learn About Secure VPNs
- Configuring a Small Office for High-Definition Videoconferencing
- Configuring Branch SRX Series for MPLS over GRE with IPsec Segmentation

1
CHAPTER

IPsec Fundamentals

IN THIS CHAPTER

- Internet Key Exchange | **2**
- IPsec Basics | **12**

Internet Key Exchange

SUMMARY

Read this topic to learn about IKE and its interaction with IPsec.

IN THIS SECTION

- Introduction to IKE | **2**
- IKE Versions | **3**
- Interaction Between IKE and IPSec | **3**
- IKEv1 Message Exchange | **4**
- Phase 1 of IKE Tunnel Negotiation | **4**
- Phase 2 of IKE Tunnel Negotiation | **6**
- IKEv2 Message Exchange | **7**
- Proxy ID | **9**
- Traffic Selectors | **9**
- IKE Authentication (Preshared Key and Certificate-Based Authentication) | **9**
- Network Address Translation-Traversal (NAT-T) | **10**
- Suite B and PRIME Cryptographic Suites | **11**

Introduction to IKE

Internet Key Exchange (IKE) is a secure key management protocol that is used to set up a secure, authenticated communications channel between two devices.

IKE does the following:

- Negotiates and manages IKE and IPsec parameters
- Authenticates secure key exchange
- Provides mutual peer authentication by means of shared secrets (not passwords) and public keys
- Provides identity protection (in main mode)

- Employs Diffie-Hellman methods and is optional in IPsec (the shared keys can be entered manually at the endpoints).

IKE Versions

Two versions of the IKE standards are available:

- IKE version 1 - IKE protocol defined in RFC 2409.
- IKE version 2 - IKE version 2 (IKEv2) is the latest version of the IKE protocol defined in RFC 7296.

Internet Key Exchange version 2 (IKEv2) is the latest version of the Internet Key Exchange (IKE) protocol defined in RFC 7296. A VPN peer is configured as either IKEv1 or IKEv2. When a peer is configured as IKEv2, it cannot fall back to IKEv1 if its remote peer initiates IKEv1 negotiation.

The advantages of using IKEv2 over IKEv1 are as follows:

- Replaces eight initial exchanges with a single four-message exchange.
- Reduces the latency for the IPsec SA setup and increases connection establishment speed.
- Increases robustness against DOS attacks.
- Improves reliability through the use of sequence numbers, acknowledgments, and error correction.
- Improves reliability, as all messages are requests or responses. The initiator is responsible for retransmitting if it does not receive a response.

Interaction Between IKE and IPSec

IPsec can establish a VPN in either of the following way:

- Internet Key Exchange (IKE) protocol— IPsec supports automated generation and negotiation of keys and security associations using the IKE protocol. Using IKE to negotiate VPNs between two endpoints provides more security than the manual key exchange.
- Manual key exchange—IPsec supports using and exchanging of keys manually (example: phone or email) on both sides to establish VPN.

IKEv1 Message Exchange

IKE negotiation includes two phases:

- Phase 1—Negotiate exchange of proposals for how to authenticate and secure the channel.
- Phase 2—Negotiate security associations (SAs) to secure the data that traverses through the IPsec tunnel.

Phase 1 of IKE Tunnel Negotiation

IN THIS SECTION

- Main Mode | 5
- Aggressive Mode | 5

Phase 1 of an AutoKey Internet Key Exchange (IKE) tunnel negotiation consists of the exchange of proposals for how to authenticate and secure the channel. The participants exchange proposals for acceptable security services such as:

- Encryption algorithms—Data Encryption Standard (DES), triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES). (See "IPsec Overview" on page 12.)
- Authentication algorithms—Message Digest 5 (MD5) and Secure Hash Algorithm (SHA). (See "IPsec Overview" on page 12.)
- Diffie-Hellman (DH) group. (See "IPsec Overview" on page 12.)
- Preshared key or RSA/DSA certificates. (See "IPsec Overview" on page 12.)

A successful Phase 1 negotiation concludes when both ends of the tunnel agree to accept at least one set of the Phase 1 security parameters proposed and then process them. Juniper Networks devices support up to four proposals for Phase 1 negotiations, allowing you to define how restrictive a range of security parameters for key negotiation you will accept. Junos OS provides predefined standard, compatible, and basic Phase 1 proposal sets. You can also define custom Phase 1 proposals.

Phase 1 exchanges can take place in either main mode or aggressive mode. You can choose your mode during IKE policy configuration.

This topic includes the following sections:

Main Mode

In main mode, the initiator and recipient send three two-way exchanges (six messages total) to accomplish the following services:

- First exchange (messages 1 and 2)—Proposes and accepts the encryption and authentication algorithms.
- Second exchange (messages 3 and 4)—Executes a DH exchange, and the initiator and recipient each provide a pseudorandom number.
- Third exchange (messages 5 and 6)—Sends and verifies the identities of the initiator and recipient.

The information transmitted in the third exchange of messages is protected by the encryption algorithm established in the first two exchanges. Thus, the participants' identities are encrypted and therefore not transmitted "in the clear."

Aggressive Mode

In aggressive mode, the initiator and recipient accomplish the same objectives as with main mode, but in only two exchanges, with a total of three messages:

- First message—The initiator proposes the security association (SA), initiates a DH exchange, and sends a pseudorandom number and its IKE identity.

  When configuring aggressive mode with multiple proposals for Phase 1 negotiations, use the same DH group in all proposals because the DH group cannot be negotiated. Up to four proposals can be configured.

- Second message—The recipient accepts the SA; authenticates the initiator; and sends a pseudorandom number, its IKE identity, and, if using certificates, the recipient's certificate.
- Third message—The initiator authenticates the recipient, confirms the exchange, and, if using certificates, sends the initiator's certificate.

Because the participants' identities are exchanged in the clear (in the first two messages), aggressive mode does not provide identity protection.

Main and aggressive modes applies only to IKEv1 protocol. IKEv2 protocol does not negotiate using main and aggressive modes.

SEE ALSO

Understanding IKE Phase 1 Configuration for Group VPNv1 | 604

Phase 2 of IKE Tunnel Negotiation

IN THIS SECTION

- Proxy IDs | 6
- Perfect Forward Secrecy | 6
- Replay Protection | 7

After the participants have established a secure and authenticated channel, they proceed through Phase 2, in which they negotiate security associations (SAs) to secure the data to be transmitted through the IPsec tunnel.

Similar to the process for Phase 1, the participants exchange proposals to determine which security parameters to employ in the SA. A Phase 2 proposal also includes a security protocol—either Encapsulating Security Payload (ESP) or Authentication Header (AH)—and selected encryption and authentication algorithms. The proposal can also specify a Diffie-Hellman (DH) group, if Perfect Forward Secrecy (PFS) is desired.

Regardless of the mode used in Phase 1, Phase 2 always operates in quick mode and involves the exchange of three messages.

This topic includes the following sections:

Proxy IDs

In Phase 2, the peers exchange proxy IDs. A proxy ID consists of a local and remote IP address prefix. The proxy ID for both peers must match, which means that the local IP address specified for one peer must be the same as the remote IP address specified for the other peer.

Perfect Forward Secrecy

PFS is a method for deriving Phase 2 keys independent from and unrelated to the preceding keys. Alternatively, the Phase 1 proposal creates the key (the SKEYID_d key) from which all Phase 2 keys are derived. The SKEYID_d key can generate Phase 2 keys with a minimum of CPU processing. Unfortunately, if an unauthorized party gains access to the SKEYID_d key, all your encryption keys are compromised.

PFS addresses this security risk by forcing a new DH key exchange to occur for each Phase 2 tunnel. Using PFS is thus more secure, although the rekeying procedure in Phase 2 might take slightly longer with PFS enabled.

Replay Protection

A replay attack occurs when an unauthorized person intercepts a series of packets and uses them later either to flood the system, causing a denial of service (DoS), or to gain entry to the trusted network. Junos OS provides a replay protection feature that enables devices to check every IPsec packet to see if it has been received previously. If packets arrive outside a specified sequence range, Junos OS rejects them. Use of this feature does not require negotiation, because packets are always sent with sequence numbers. You simply have the option of checking or not checking the sequence numbers.

SEE ALSO

Understanding IPsec SA Configuration for Group VPNv2 | **663**

policy (Security IPsec)

IKEv2 Message Exchange

IN THIS SECTION

- IKEv2 Configuration Payload | **8**
- IKEv2 Rekeying and Reauthentication | **8**
- IKEv2 Fragmentation | **8**
- Traffic Selectors for IKEv2 | **9**

IKE version 2 is the successor to the IKEv1 method. It provides a secure VPN communication channel between peer VPN devices and defines negotiation and authentication for IPsec security associations (SAs) in a protected manner.

IKEv2 does not include phase 1 and phase 2 similar to IKEv1, but there are four message exchanges occur to negotiate an IPsec tunnel with IKEv2. The message exchange in IKEv2 are:

- Negotiates the security attributes to establish the IPsec tunnel. This includes exchanging the protocols/parameters used, and Diffie-Hellman groups.

- Each peer establishes or authenticates their identities while the IPsec tunnel is established.
- Peers to create additional security associations between each other.
- Peers perform liveliness detection, removing SA relationships, and reporting error messages.

IKEv2 Configuration Payload

Configuration payload is an IKEv2 option offered to propagate provisioning information from a responder to an initiator. IKEv2 configuration payload is supported with route-based VPNs only.

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*, defines 15 different configuration attributes that can be returned to the initiator by the responder.

IKEv2 Rekeying and Reauthentication

With IKEv2, rekeying and reauthentication are separate processes.

Rekeying establishes new keys for the IKE security association (SA) and resets message ID counters, but it does not reauthenticate the peers.

Reauthentication verifies that VPN peers retain their access to authentication credentials. Reauthentication establishes new keys for the IKE SA and child SAs; rekeys of any pending IKE SA or child SA are no longer needed. After the new IKE and child SAs are created, the old IKE and child SAs are deleted.

IKEv2 reauthentication is disabled by default. You enable reauthentication by configuring a reauthentication frequency value between 1 and 100. The reauthentication frequency is the number of IKE rekeys that occurs before reauthentication occurs. For example, if the configured reauthentication frequency is 1, reauthentication occurs every time there is an IKE rekey. If the configured reauthentication frequency is 2, reauthentication occurs at every other IKE rekey. If the configured reauthentication frequency is 3, reauthentication occurs at every third IKE rekey, and so on.

IKEv2 Fragmentation

When certificate-based authentication is used, IKEv2 packets can exceed the path MTU if multiple certificates are transmitted. If the IKE message size exceeds the path MTU, the messages are fragmented at the IP level. Some network equipment, such as NAT devices, does not allow IP fragments to pass through, which prevents the establishment of IPsec tunnels.

IKEv2 message fragmentation, as described in RFC 7383, Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation, allows IKEv2 to operate in environments where IP fragments might be blocked and peers would not be able to establish an IPsec security association (SA). IKEv2 fragmentation splits a large IKEv2 message into a set of smaller ones so that there is no fragmentation at the IP level. Fragmentation takes place before the original message is encrypted and authenticated, so that each

fragment is separately encrypted and authenticated. On the receiver, the fragments are collected, verified, decrypted, and merged into the original message.

Traffic Selectors for IKEv2

You can configure traffic Selectors in IKEv2 used during IKE negotiation. A traffic selector is an agreement between IKE peers to permit traffic through a VPN tunnel if the traffic matches a specified pair of local and remote addresses. Only the traffic that conforms to a traffic selector is permitted through the associated security association (SA). Traffic selectors are used during the tunnel creation to set up the tunnel and to determine what traffic is allowed through the tunnel.

Proxy ID

A proxy-ID is used during phase 2 of Internet Key Exchange (IKE) Virtual Private Network (VPN) negotiations. Both ends of a VPN tunnel either have a proxy-ID manually configured (route-based VPN) or just use a combination of source IP, destination IP, and service in a tunnel policy. When phase 2 of IKE is negotiated, each end compares the configured local and remote proxy-ID with what is actually received.

Traffic Selectors

Proxy ID is supported for route-based VPNs. The multi-proxy ID is supported for route-based VPNs. The multi-proxy ID is also known as traffic selector. A traffic selector is an agreement between IKE peers to permit traffic through a tunnel, if the traffic matches a specified pair of local and remote addresses. You define a traffic selector within a specific route-based VPN, which can result in multiple Phase 2 IPsec SAs. Only traffic that conforms to a traffic selector is permitted through an SA. The traffic selector is commonly required when remote gateway devices are non-Juniper Networks devices.

IKE Authentication (Preshared Key and Certificate-Based Authentication)

The IKE negotiations provides the ability to establish a secure channel over which two parties can communicate. You can define how the two parties authenticate each other using a preshared key authentication or certificate based authentication.

Preshared Key Authentication	Certificate-Based Authentication
Common way to establish a VPN connection.	Secure way to establish VPN connection.
• Preshared key is a password that is the same for both the parties. This password is exchanged in advance using a phone, through a verbal exchange, or through less secure mechanisms, even e-mail. • Preshared key must consist of at least 8 characters (12 or more is recommended) using a combination of letters, numbers, and nonalphanumeric characters, along with different cases for the letters. • Preshared key must not use a dictionary word.	Certificates are composed of a public and private key, and can be signed by a primary certificate known as a certificate authority (CA)
The parties authenticate each other by encrypting the preshared key with the peer's public key, which is obtained in the Diffie-Hellman exchange.	The parties check certificates to confirm if they are signed by a trusted CA.
Preshared keys are commonly deployed for site-to-site IPsec VPNs, either within a single organization or between different organizations.	Certificates are also far more ideal in larger scale environments with numerous peer sites that should not all share a preshared key.

Network Address Translation-Traversal (NAT-T)

Network Address Translation-Traversal (NAT-T) is a method for getting around IP address translation issues encountered when data protected by IPsec passes through a NAT device for address translation.

Any changes to the IP addressing, which is the function of NAT, causes IKE to discard packets. After detecting one or more NAT devices along the data path during Phase 1 exchanges, NAT-T adds a layer of User Datagram Protocol (UDP) encapsulation to IPsec packets so they are not discarded after address translation. NAT-T encapsulates both IKE and ESP traffic within UDP with port 4500 used as both the source and destination port. Because NAT devices age out stale UDP translations, keepalive messages are required between the peers.

The location of a NAT device can be such that:

- Only the IKEv1 or IKEv2 initiator is behind a NAT device. Multiple initiators can be behind separate NAT devices. Initiators can also connect to the responder through multiple NAT devices.
- Only the IKEv1 or IKEv2 responder is behind a NAT device.
- Both the IKEv1 or IKEv2 initiator and the responder are behind a NAT device.

Suite B and PRIME Cryptographic Suites

Suite B is a set of cryptographic algorithms designated by the U.S. National Security Agency to allow commercial products to protect traffic that is classified at secret or top secret levels. Suite B protocols are defined in RFC 6379, *Suite B Cryptographic Suites for IPsec*. The Suite B cryptographic suites provide Encapsulating Security Payload (ESP) integrity and confidentiality and should be used when ESP integrity protection and encryption are both required. Protocol Requirements for IP Modular Encryption (PRIME), an IPsec profile defined for public sector networks in the United Kingdom, is based on the Suite B cryptographic suite, but uses AES-GCM rather than AES-CBC for IKEv2 negotiations.

The following cryptographic suites are supported:

- Suite-B-GCM-128
  - ESP: Advanced Encryption Standard (AES) encryption with 128-bit keys and 16-octet integrity check value (ICV) in Galois Counter Mode (GCM).
  - IKE: AES encryption with 128-bit keys in cipher block chaining (CBC) mode, integrity using SHA-256 authentication, key establishment using Diffie-Hellman (DH) group 19, and authentication using Elliptic Curve Digital Signature Algorithm (ECDSA) 256-bit elliptic curve signatures.
- Suite-B-GCM-256
  - ESP: AES encryption with 256-bit keys and 16-octet ICV in GCM for ESP.
  - IKE: AES encryption with 256-bit keys in CBC mode, integrity using SHA-384 authentication, key establishment using DH group 20, and authentication using ECDSA 384-bit elliptic curve signatures.
- PRIME-128
  - ESP: AES encryption with 128-bit keys and 16-octet ICV in GCM.
  - IKE: AES encryption with 128-bit keys in GCM, key establishment using DH group 19, and authentication using ECDSA 256-bit elliptic curve signatures.
- PRIME-256

- ESP: AES encryption with 256-bit keys and 16-octet ICV in GCM for ESP.
- IKE: AES encryption with 256-bit keys in GCM, key establishment using DH group 20, and authentication using ECDSA 384-bit elliptic curve signatures.

Suite-B cryptographic suites support IKEv1 and IKEv2. PRIME cryptographic suites only support IKEv2.

IPsec Basics

SUMMARY

Read this topic to learn about IPsec key management, security protocols, tunnel negotiation, and IPsec and IKE standards.

IN THIS SECTION

- IPsec Overview | **12**
- IPsec Key Management | **14**
- IPsec Security Protocols | **16**
- IPsec Tunnel Negotiation | **18**
- Supported IPsec and IKE Standards | **19**
- Platform-Specific IPsec Tunnel Behavior | **21**
- Additional Platform Information | **21**

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific IPsec Tunnel Behavior" on page 21 section for notes related to your platform.

See the "Additional Platform Information" on page 21 section for more information.

IPsec Overview

IN THIS SECTION

- Security Associations | **13**

IPsec is a suite of related protocols for cryptographically securing communications at the IP Packet Layer. IPsec provides methods for manual and automatic negotiation of security associations (SAs) and key distribution. The domain of interpretation (DOI) gathers every relevant attribute. The IPsec DOI is a document that defines all security parameters needed for successful VPN tunnel negotiation—essentially, every attribute required for SA and IKE negotiations. See RFC 2407 and RFC 2408 for more information.

To use IPsec security services, you create SAs between hosts. A SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. IPsec supports two types of SAs: manual and dynamic.

IPsec supports two modes of security (transport mode and tunnel mode).

Security Associations

A security association (SA) is a unidirectional agreement between the VPN participants regarding the methods and parameters to use in securing a communication channel. Full bidirectional communication requires at least two SAs, one for each direction. Through the SA, an IPsec tunnel can provide the following security functions:

- Privacy (through encryption)
- Content integrity (through data authentication)
- Sender authentication and—if using certificates—nonrepudiation (through data origin authentication)

The security functions you employ depend on your needs. If you need only to authenticate the IP packet source and content integrity, you can authenticate the packet without applying any encryption. On the other hand, if you are concerned only with preserving privacy, you can encrypt the packet without applying any authentication mechanisms. Optionally, you can both encrypt and authenticate the packet. Most network security designers choose to encrypt, authenticate, and replay-protect their VPN traffic.

An IPsec tunnel consists of a pair of unidirectional SAs—one SA for each direction of the tunnel. A SA specifies the security parameter index (SPI), destination IP address, and security protocol such as Authentication Header (AH) or Encapsulating Security Payload (ESP). A SA groups the following components for securing communications:

- Security algorithms and keys.
- Protocol mode, either transport or tunnel. Junos OS devices always use tunnel mode. See "Packet Processing in Tunnel Mode" on page 81.
- Key-management method, either manual key or AutoKey IKE.
- SA lifetime.

For inbound traffic, Junos OS looks up the SA by using the following triplet:

- Destination IP address.
- Security protocol, either AH or ESP.
- SPI value.

For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel.

IPsec Key Management

IN THIS SECTION

- Manual Key | **14**
- AutoKey IKE | **15**
- Diffie-Hellman Exchange | **15**

The distribution and management of keys are critical to using VPNs successfully. Junos OS supports IPsec technology for creating VPN tunnels with three kinds of key creation mechanisms:

- Manual key
- AutoKey IKE with a preshared key (PSK) or a certificate

You can choose your key creation mechanism—also called authentication method—during Phase 1 and Phase 2 proposal configuration. See "Internet Key Exchange" on page 2.

This topic includes the following sections:

Manual Key

With manual keys, administrators at both ends of a tunnel configure all the security parameters. Manual key is a viable technique for small, static networks where the distribution, maintenance, and tracking of keys are not difficult. However, safely distributing manual-key configurations across great distances poses security issues. Aside from passing the keys face-to-face, you cannot be completely certain that unauthorized parties have not compromised the keys in transit. Also, whenever you want to change the key, you are faced with the same security issues as when you initially distributed it.

AutoKey IKE

When you need to create and manage numerous tunnels, you need a method that does not require you to configure every element manually. IPsec supports the automated generation and negotiation of keys and SA using the Internet Key Exchange (IKE) protocol. Junos OS refers to such automated tunnel negotiation as AutoKey IKE and supports AutoKey IKE with preshared keys and AutoKey IKE with certificates.

- AutoKey IKE with preshared keys—Using AutoKey IKE with preshared keys to authenticate the participants in an IKE session, each side must configure and securely exchange the PSK in advance. In this regard, the issue of secure key distribution is the same as that with manual keys. However, once distributed, an autokey, unlike a manual key, can automatically change its keys at predetermined intervals using the IKE protocol. Frequently changing keys greatly improves security, and automatically doing so greatly reduces key-management responsibilities. However, changing keys increases traffic overhead; therefore, changing keys too often can reduce data transmission efficiency.

  A PSK is a key for both encryption and decryption, which both participants must have before initiating communication.

- AutoKey IKE with certificates—When using certificates to authenticate the participants during an AutoKey IKE negotiation, each side generates a public-private keypair and acquires a certificate. As long as both the sides trust the issuing certificate authority (CA), the participants can retrieve the peer's public key and verify the peer's signature. You need not keep track of the keys and SAs; IKE does it automatically.

Diffie-Hellman Exchange

A Diffie-Hellman (DH) exchange allows participants to produce a shared secret value. The strength of the technique is that it allows participants to create the secret value over an unsecured medium without passing the secret value through the wire. The size of the prime modulus used in each group's calculation differs as shown in the below table. The device performs DH exchange operations either in software or in hardware. The following Table 1 on page 15 lists different Diffie Hellman (DH) groups and specifies whether the operation performed for that group is in the hardware or in software.

Table 1: Diffie Hellman (DH) groups and their exchange operations performed

Diffie-Hellman (DH) Group	Prime Module Size
DH Group 1	768-bit
DH Group 2	102-bit

Table 1: Diffie Hellman (DH) groups and their exchange operations performed *(Continued)*

Diffie-Hellman (DH) Group	Prime Module Size
DH Group 5	1536-bit
DH Group 14	2048-bit
DH Group 15	3072-bit
DH Group 16	4096-bit
DH Group 19	256-bit elliptic curve
DH Group 20	384-bit elliptic curve
DH Group 21	521-bit elliptic curve
DH Group 24	2048-bit with 256-bit prime order subgroup

We do not recommend the use of DH groups 1, 2, and 5.

Because the modulus for each DH group is a different size, the participants must agree to use the same group.

IPsec Security Protocols

IN THIS SECTION

- IPsec Authentication Algorithms (AH Protocol) | **17**
- IPsec Encryption Algorithms (ESP Protocol) | **17**

IPsec uses two protocols to secure communications at the IP layer:

- Authentication Header (AH)—A security protocol for authenticating the source of an IP packet and verifying the integrity of its content
- Encapsulating Security Payload (ESP)—A security protocol for encrypting the entire IP packet and authenticating its content

You can choose your security protocols—also called *authentication and encryption algorithms*—during Phase 2 proposal configuration. See "Internet Key Exchange" on page 2.

For each VPN tunnel, both AH and ESP tunnel sessions are installed on Services Processing Units (SPUs) and the control plane. The device updates the tunnel sessions with the negotiated protocol once the negotiation is completed. The `show security flow session` and `show security flow cp-session` operational mode commands display the details of ESP and AH tunnel sessions.

This topic includes the following sections:

IPsec Authentication Algorithms (AH Protocol)

The Authentication Header (AH) protocol provides a means to verify the authenticity and integrity of the content and origin of a packet. You can authenticate the packet by the checksum calculated through a Hash Message Authentication Code (HMAC) using a secret key and either MD5 or SHA hash functions.

- Message Digest 5 (MD5)—An algorithm that produces a 128-bit hash (also called a *digital signature* or *message digest*) from a message of arbitrary length and a 16-byte key. The resulting hash is used, like a fingerprint of the input, to verify content and source authenticity and integrity.
- Secure Hash Algorithm (SHA)—An algorithm that produces a 160-bit hash from a message of arbitrary length and a 20-byte key. SHA is more secure than MD5 because of the larger hashes it produces. Because the ASIC performs the computational processing, the performance cost is negligible.

For more information about MD5 hashing algorithms, see RFC 1321 and RFC 2403. For more information about SHA hashing algorithms, see RFC 2404. For more information about HMAC, see RFC 2104.

IPsec Encryption Algorithms (ESP Protocol)

The ESP protocol provides a means to ensure privacy (encryption) and source authentication and content integrity (authentication). ESP in tunnel mode encapsulates the entire IP packet (header and payload) and then appends a new IP header to the now-encrypted packet. This new IP header contains the destination address needed to route the protected data through the network. See "Packet Processing in Tunnel Mode" on page 81.

With ESP, you can both encrypt and authenticate, encrypt only, or authenticate only. For encryption, you can choose one of the following encryption algorithms:

- Data Encryption Standard (DES)—A cryptographic block algorithm with a 56-bit key.
- Triple DES (3DES)—A more powerful version of DES in which the original DES algorithm is applied in three rounds, using a 168-bit key. DES provides significant performance savings but is considered unacceptable for many classified or sensitive material transfers.
- Advanced Encryption Standard (AES)—An encryption standard which offers greater interoperability with other devices. Junos OS supports AES with 128-bit, 192-bit, and 256-bit keys.
- ChaCha20-Poly1305 Authenticated Encryption with Associated Data—ChaCha20 stream cipher which supports Authenticated Encryption with Associated Data (AEAD) using Poly1305 authenticator.

For authentication, you can use either MD5 or SHA algorithms.

Even though it is possible to select NULL for encryption, studies demonstrate that IPsec might be vulnerable to attack under such circumstances. Therefore, we suggest that you choose an encryption algorithm for maximum security.

IPsec Tunnel Negotiation

The following two different modes determine how the VPN exchanges traffic.

- Tunnel mode—Protect traffic by encapsulating the original IP packet within another packet in the VPN tunnel. This mode uses preshared keys with IKE to authenticate peers or digital certificates with IKE to authenticate peers. Tunnel mode is most commonly used when hosts within separate private networks want to communicate over a public network. Both VPN clients and VPN gateways use tunnel mode to protect communications that come from or go to non-IPsec systems.
- Transport mode—Protect traffic by sending the packet directly between the two hosts that have established the IPsec tunnel. That is, when the communication endpoint and cryptographic endpoint are the same. In this mode, the encryption mechanism secures the data portion of the IP packet, but the IP header remains unencrypted. VPN gateways that provide encryption and decryption services for protected hosts cannot use transport mode for protected VPN communications. The IP addresses of the source or destination can be modified if the packet is intercepted. Because of its construction, transport mode can be used only when the communication endpoint and cryptographic endpoint are the same.

Supported IPsec and IKE Standards

On routers equipped with one or more MS-MPCs, MS-MICs, or DPCs, the Canada and U.S. version of Junos OS substantially supports the following RFCs, which define standards for IP Security (IPsec) and Internet Key Exchange (IKE).

- RFC 2085, *HMAC-MD5 IP Authentication with Replay Prevention*
- RFC 2401, *Security Architecture for the Internet Protocol* (obsoleted by RFC 4301)
- RFC 2402, *IP Authentication Header* (obsoleted by RFC 4302)
- RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*
- RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH* (obsoleted by RFC 4305)
- RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*
- RFC 2406, *IP Encapsulating Security Payload (ESP)* (obsoleted by RFC 4303 and RFC 4305)
- RFC 2407, *The Internet IP Security Domain of Interpretation for ISAKMP* (obsoleted by RFC 4306)
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)* (obsoleted by RFC 4306)
- RFC 2409, *The Internet Key Exchange (IKE)* (obsoleted by RFC 4306)
- RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*
- RFC 2451, *The ESP CBC-Mode Cipher Algorithms*
- RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*
- RFC 3193, *Securing L2TP using IPsec*
- RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*
- RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*
- RFC 3948, *UDP Encapsulation of IPsec ESP Packets*
- RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)*
- RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*
- RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*
- RFC 4301, *Security Architecture for the Internet Protocol*

- RFC 4302, *IP Authentication Header*
- RFC 4303, *IP Encapsulating Security Payload (ESP)*
- RFC 4305, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
- RFC 4306, *Internet Key Exchange (IKEv2) Protocol*
- RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*
- RFC 4308, *Cryptographic Suites for IPsec*

  Only Suite VPN-A is supported in Junos OS.
- RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*
- RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
- RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)* (obsoleted by RFC 7296)
- RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*
- RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*
- RFC 7634, *ChaCha20, Poly1305, and Their Use in the Internet Key Exchange Protocol (IKE) and IPsec*
- RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

Junos OS partially supports the following RFCs for IPsec and IKE:

- RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*
- RFC 5114, *Additional Diffie-Hellman Groups for Use with IETF Standards*
- RFC 5903, *Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2*

The following RFCs and Internet draft do not define standards, but provide information about IPsec, IKE, and related technologies. The IETF classifies them as “Informational.”

- RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*
- RFC 2412, *The OAKLEY Key Determination Protocol*
- RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*

- Internet draft draft-eastlake-sha2-02.txt, *US Secure Hash Algorithms (SHA and HMAC-SHA)* (expires July 2006)

SEE ALSO

> Services Interfaces Overview for Routing Devices
>
> MX Series 5G Universal Routing Platform Interface Module Reference
>
> Accessing Standards Documents on the Internet

Platform-Specific IPsec Tunnel Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behavior for your platform.

Table 2: Platform-Specific Behavior

Platform	Difference

SRX Series	• On SRX5400, SRX5600, and SRX5800 devices that support IPsec VPNs:
	  • Junos OS updates the negotiated protocol for the tunnel sessions on anchor SPUs.
	  • Non-anchor SPUs retain ESP and AH tunnel sessions.

Additional Platform Information

Use Feature Explorer to confirm platform and release support for specific features. Additional Platforms may be supported.

Table 3: Additional Platform Information

Feature	SRX300 SRX320 SRX340 SRX345 SRX380	SRX400 SRX440 SRX440-2 AC	SRX1500 SRX1600	SRX2300 SRX4120 SRX4100 SRX4200 SRX4300 SRX4600 SRX4700	SRX5400 SRX5600 SRX5800	vSRX Virtual Firewalls
Support for DH groups 15, 16, and 21	No	Yes	Yes	Yes	Yes	Yes for vSRX 3.0 with `junos-ike` package

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.2R1	Support for ChaCha20-Poly1305 algorithm added to SRX1600, SRX2300, SRX4300, SRX4600, SRX5400, SRX5600, SRX5800, and vSRX 3.0 in Junos OS Release 24.2R1.
20.3R1	Starting in Junos OS Release 20.3R1, vSRX Virtual Firewalls support DH groups 15, 16, and 21.
19.1R1	Starting in Junos OS Release 19.1R1, SRX Series Firewalls support DH groups 15, 16, and 21.

2

CHAPTER

IPsec VPN in Junos OS

IN THIS CHAPTER

- IKE for IPsec VPN | **24**
- IPsec VPN Overview | **79**
- Inline IPsec | **103**

IKE for IPsec VPN

SUMMARY

Learn about IKEv2 for IPsec VPN and its configuration in Junos OS.

IN THIS SECTION

- IKE and IPsec Packet Processing | **25**
- Introduction to IKE in Junos OS | **31**
- IKE Proposal | **36**
- IKE Policy | **37**
- Rekeying and Reauthentication | **37**
- IKE Authentication (Certificate-Based Authentication) | **39**
- Signature Authentication in IKEv2 | **42**
- IKE Protection from DDoS Attacks | **43**
- Configure Protection Against IKE DDoS Attacks | **47**
- Example: Configuring a Device for Peer Certificate Chain Validation | **54**
- IKEv2 Fragmentation | **67**
- IKE Policy with Local Certificate | **68**
- IKE Policy with a Trusted CA | **72**
- IKE Policy with Trusted CA Group | **74**
- Configuring Establish-Tunnel Responder-only in IKE | **77**
- Platform-Specific IKEv2 Responder Only Behavior | **78**

Internet Key Exchange version 2 (IKEv2) is an IPsec based tunneling protocol. IKEv2 provides a secure VPN communication channel between peer VPN devices and defines negotiation and authentication for IPsec security associations (SAs) in a protected manner.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific IKEv2 Responder Only Behavior" on page 78 section for notes related to your platform.

IKE and IPsec Packet Processing

IN THIS SECTION

- IKE Packet Processing | 25
- IPsec Packet Processing | 28

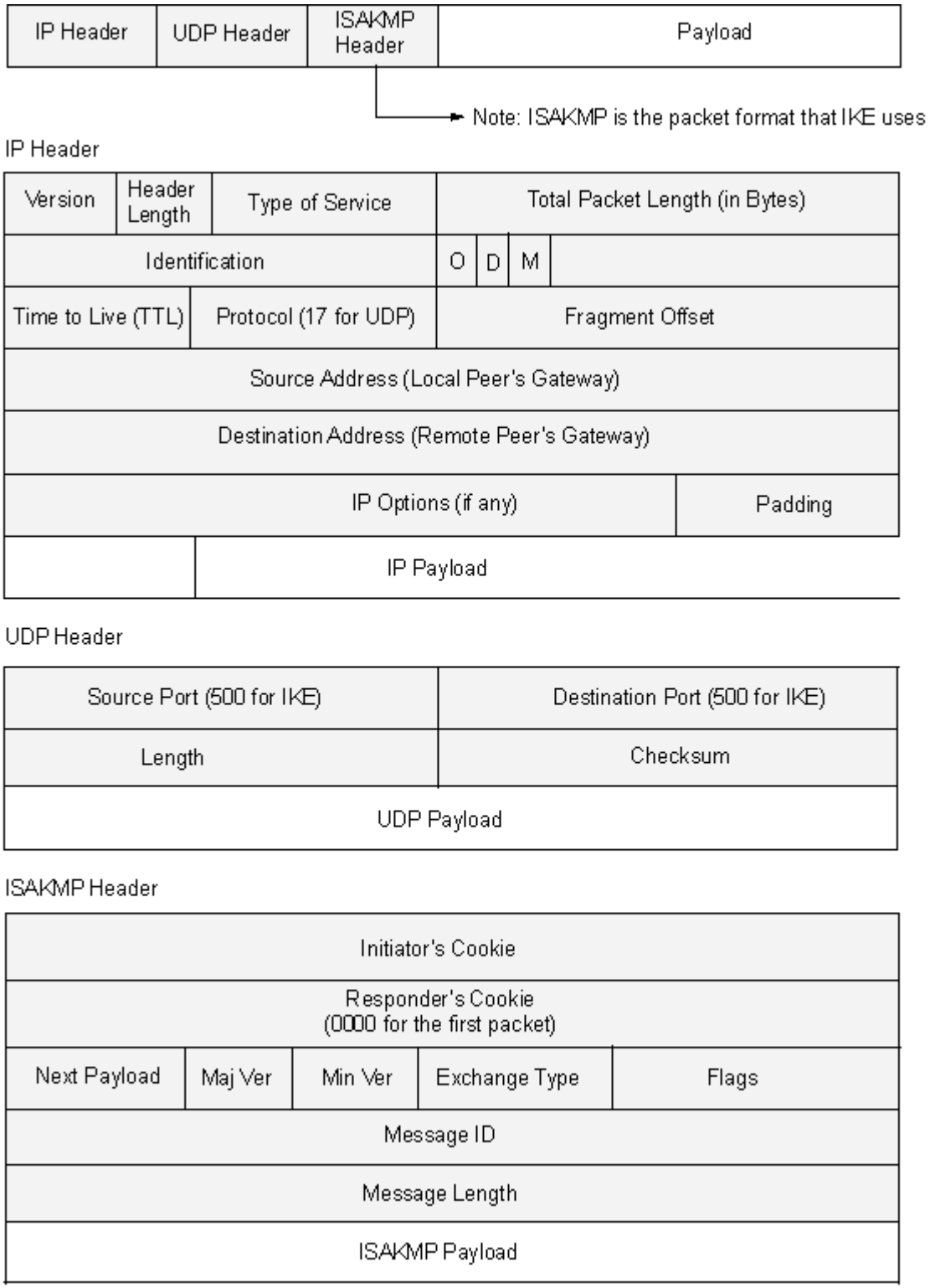
IKE provides tunnel management for IPsec and authenticates end entities. IKE performs a Diffie-Hellman (DH) key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the IP layer.

IKE Packet Processing

When a cleartext packet arrives on a Juniper Networks device that requires tunneling, and no active Phase 2 SA exists for that tunnel, Junos OS begins IKE negotiations and drops the packet. The source and destination addresses in the IP packet header are those of the local and remote IKE gateways, respectively. In the IP packet payload, there is a UDP segment encapsulating an ISAKMP (IKE) packet. The format for IKE packets is the same for Phase 1 and Phase 2. See Figure 1 on page 26.

Meanwhile, the source host has sent the dropped packet again. Typically, by the time the second packet arrives, IKE negotiations are complete, and Junos OS protects the packet and all subsequent packets in the session—with IPsec before forwarding it.

Figure 1: IKE Packet for Phases 1 and 2



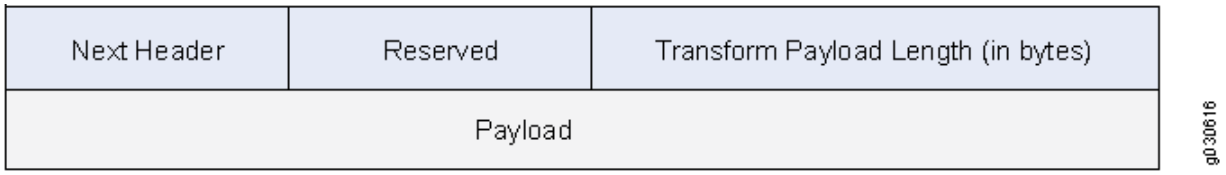
The Next Payload field contains a number indicating one of the following payload types:

- 0002—SA Negotiation Payload contains a definition for a Phase 1 or Phase 2 SA.
- 0004—Proposal Payload can be a Phase 1 or Phase 2 proposal.
- 0008—Transform Payload gets encapsulated in a proposal payload that gets encapsulated in a SA payload.
- 0010—Key Exchange (KE) Payload contains information necessary for performing a key exchange, such as a DH public value.
- 0020—Identification (IDx) Payload.
  - In Phase 1, IDii indicates the initiator ID, and IDir indicates the responder ID.
  - In Phase 2, IDui indicates the user initiator, and IDur indicates the user responder.

  The IDs are IKE ID types such as FQDN, U-FQDN, IP address, and ASN.1_DN.
- 0040—Certificate (CERT) Payload.
- 0080—Certificate Request (CERT_REQ) Payload.
- 0100—Hash (HASH) Payload contains the digest output of a particular hash function.
- 0200—Signature (SIG) Payload contains a digital signature.
- 0400—Nonce (Nx) Payload contains some pseudorandom information necessary for the exchange).
- 0800—Notify Payload.
- 1000—ISAKMP Delete Payload.
- 2000—Vendor ID (VID) Payload can be included anywhere in Phase 1 negotiations. Junos OS uses it to mark support for NAT-T.

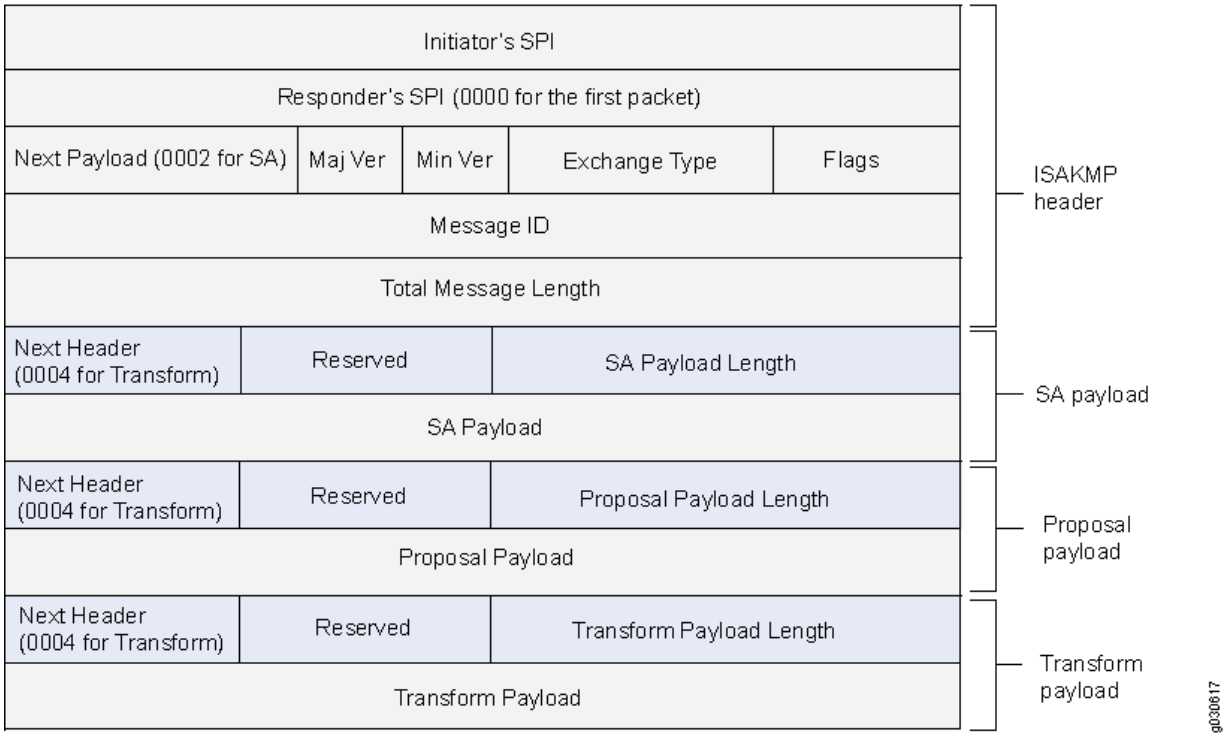
Each ISAKMP payload begins with the same generic header, as shown in Figure 2 on page 27.

Figure 2: Generic ISAKMP Payload Header



There can be multiple ISAKMP payloads chained together, with each subsequent payload type indicated by the value in the Next Header field. A value of **0000** indicates the last ISAKMP payload. See Figure 3 on page 28 for an example.

Figure 3: ISAKMP Header with Generic ISAKMP Payloads

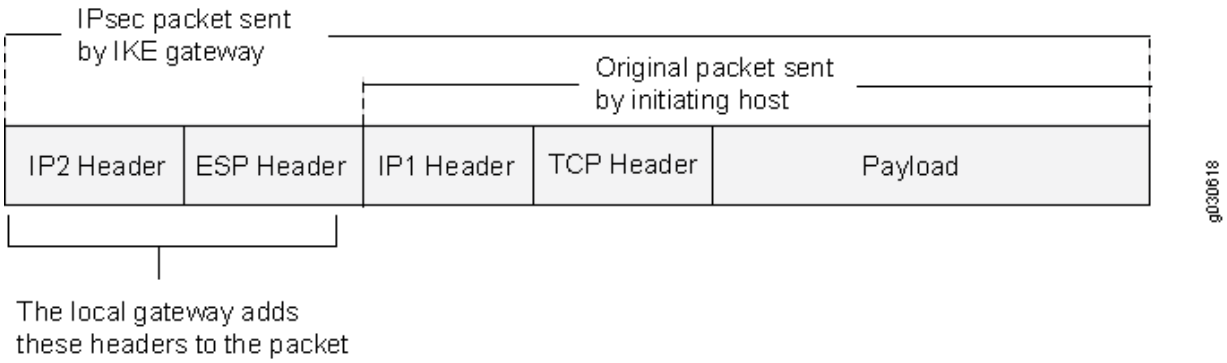


IPsec Packet Processing

After IKE negotiations complete and the two IKE gateways have established Phase 1 and Phase 2 SAs, all subsequent packets are forwarded using the tunnel. If the Phase 2 SA specifies the Encapsulating Security Protocol (ESP) in tunnel mode, the packet looks like the one shown in Figure 4 on page 29. The device adds two additional headers to the original packet that the initiating host sends.

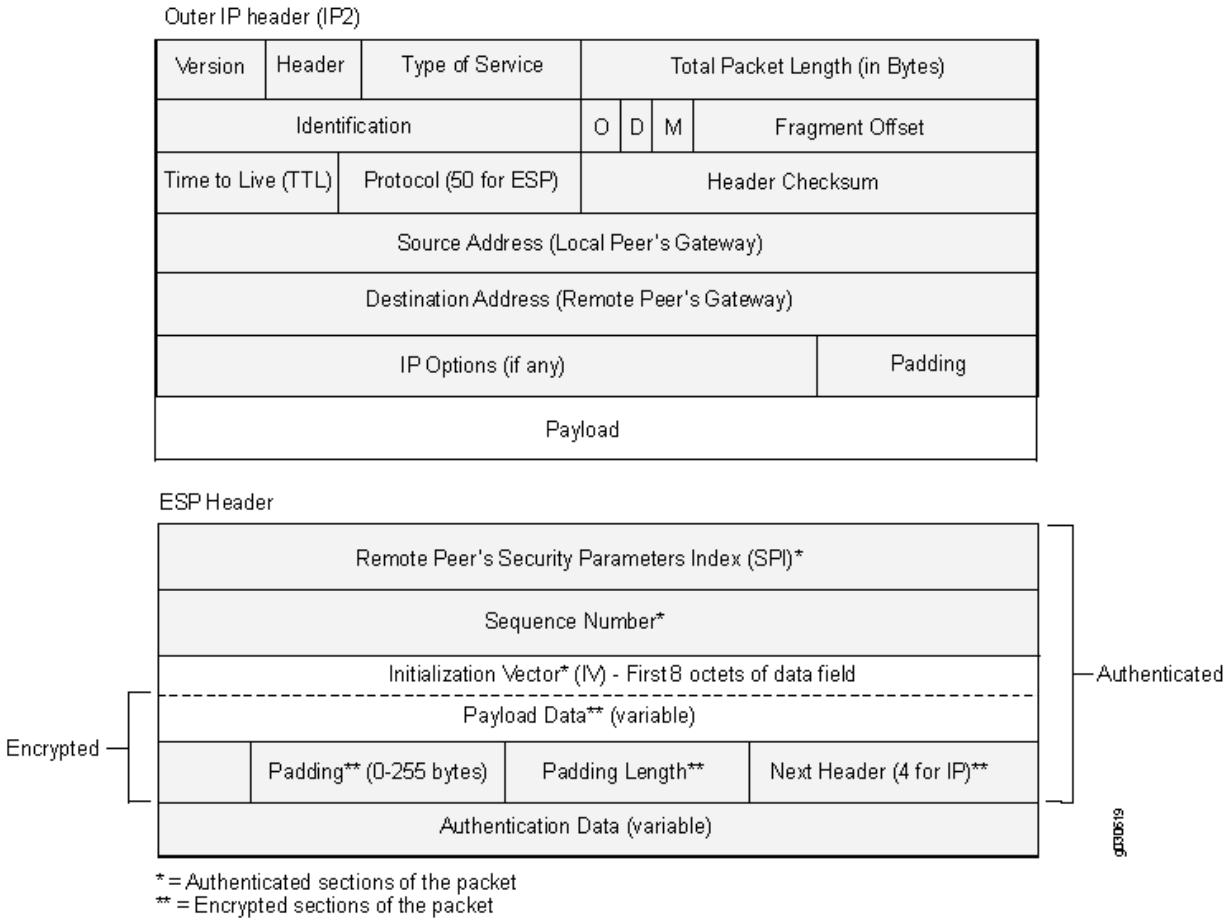
As shown in Figure 4 on page 29, the packet that the initiating host constructs includes the payload, the TCP header, and the inner IP header (IP1).

Figure 4: IPsec Packet—ESP in Tunnel Mode



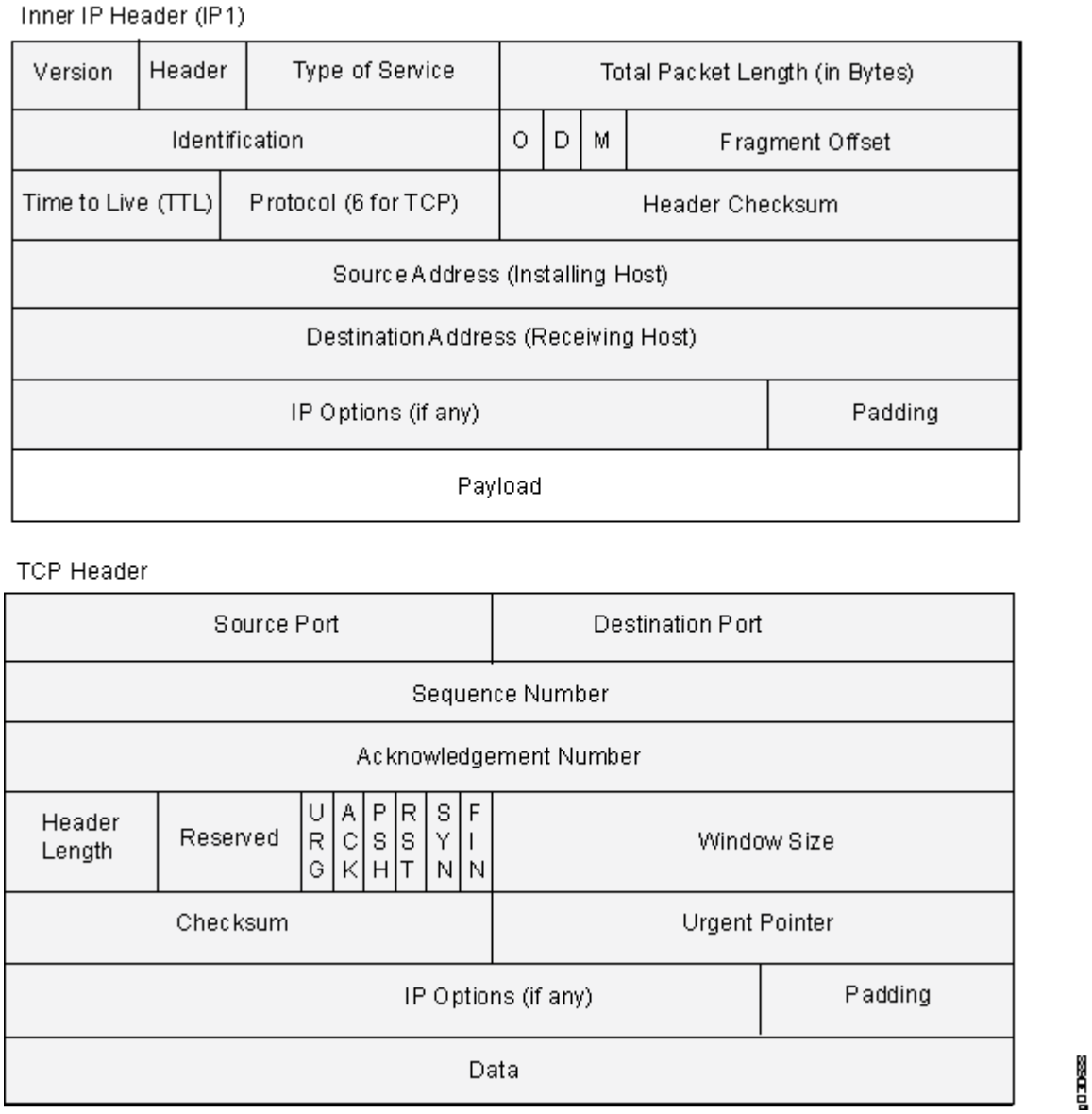
The router IP header (IP2), which Junos OS adds, contains the IP address of the remote gateway as the destination IP address and the IP address of the local router as the source IP address. Junos OS also adds an ESP header between the outer and inner IP headers. The ESP header contains information that allows the remote peer to properly process the packet when it receives it. See Figure 5 on page 29.

Figure 5: Outer IP Header (IP2) and ESP Header



The Next Header field indicates the type of data in the payload field. In tunnel mode, this value is 4, indicating an IP packet is contained within the payload. See Figure 6 on page 30.

Figure 6: Inner IP Header (IP1) and TCP Header



Introduction to IKE in Junos OS

IN THIS SECTION

- Configuring IKEv2 in Junos OS | 32
- Understanding IKEv2 Configuration Payload | 32
- Understanding Pico Cell Provisioning | 35

IKE provides ways to exchange keys for encryption and authentication securely over an unsecured medium such as the Internet. IKE enables a pair of security gateways to: Dynamically establish a secure tunnel over which security gateways can exchange tunnel and key information. Set up user-level tunnels or SAs, including tunnel attribute negotiations and key management. These tunnels can also be refreshed and terminated on top of the same secure channel. IKE employs Diffie-Hellman methods and is optional in IPsec (the shared keys can be entered manually at the endpoints).

IKEv2 includes support for:

- Route-based VPNs.
- Site-to-site VPNs.
- Dead peer detection.
- *Chassis cluster*.
- Pre-shared key authentication.
- Certificate-based authentication.
- Child SAs. An IKEv2 child SA is known as a Phase 2 SA in IKEv1. In IKEv2, a child SA cannot exist without the underlying IKE SA.
- AutoVPN.
- Dynamic endpoint VPN.
- EAP is supported for Remote Access using IKEv2.
- Traffic selectors.

IKEv2 does not support the following features:

- Policy-based VPN.

- VPN monitoring.
- IP Payload Compression Protocol (IPComp).

Configuring IKEv2 in Junos OS

A VPN peer is configured as either IKEv1 or IKEv2. When a peer is configured as IKEv2, it cannot fall back to IKEv1 if its remote peer initiates IKEv1 negotiation. By default, Juniper Networks security devices are IKEv1 peers.

Use the `version v2-only` configuration statement at the [`edit security ike gateway` *`gw-name`*] hierarchy level to configure IKEv2.

The IKE version is displayed in the output of the `show security ike security-associations` and `show security ipsec security-associations` CLI operational commands.

Juniper Networks devices support up to four proposals for Phase 2 negotiations, allowing you to define how restrictive a range of tunnel parameters you will accept. Junos OS provides predefined standard, compatible, and basic Phase 2 proposal sets. You can also define custom Phase 2 proposals.

Understanding IKEv2 Configuration Payload

Configuration payload is an Internet Key Exchange version 2 (IKEv2) option offered to propagate provisioning information from a responder to an initiator. IKEv2 configuration payload is supported with route-based VPNs only.

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*, defines 15 different configuration attributes that can be returned to the initiator by the responder. Table 4 on page 32 describes the IKEv2 configuration attributes supported on the firewalls.

Table 4: IKEv2 Configuration Attributes

Attribute Type	Value	Description	Length
INTERNAL_IP4_ADDRESS	1	Specifies an address on the internal network. Multiple internal addresses can be requested. The responder can send up to the number of addresses requested.	0 or 4 octets
INTERNAL_IP4_NETMASK	2	Specifies the internal network's netmask value. Only one netmask value is allowed in the request and response messages (for example, 255.255.255.0), and it must be used only with an INTERNAL_IP4_ADDRESS attribute.	0 or 4 octets

Table 4: IKEv2 Configuration Attributes *(Continued)*

Attribute Type	Value	Description	Length
INTERNAL_IP4_DNS	3	Specifies an address of a DNS server within the network. Multiple DNS servers can be requested. The responder can respond with zero or more DNS server attributes.	0 or 4 octets
INTERNAL_IP4_NBNS	4	Specifies an address of a NetBIOS name server (NBNS), for example, a WINS server, within the network. Multiple NBNS servers can be requested. The responder can respond with zero or more NBNS server attributes.	0 or 4 octets
INTERNAL_IP6_ADDRESS	8	Specifies an address on the internal network. Multiple internal addresses can be requested. The responder can send up to the number of addresses requested.	0 or 17 octets
INTERNAL_IP6_DNS	10	Specifies an address of a DNS server within the network. Multiple DNS servers can be requested. The responder can respond with zero or more DNS server attributes.	0 or 16 octets

For the IKE responder to provide the initiator with provisioning information, it must acquire the information from a specified source such as a RADIUS server. Provisioning information can also be returned from a DHCP server through a RADIUS server. On the RADIUS server, the user information should not include an authentication password. The RADIUS server profile is bound to the IKE gateway using the `aaa` *`access-profile profile-name`* configuration at the [`edit security ike gateway` *`gateway-name`*] hierarchy level.

Junos OS improved the IKEv2 configuration payload to support the following features:

- Support for IPv4 and IPv6 local address pool. You can also assign a fixed IP address to a peer.

  During IKE establishment, the initiator requests for an IPv4 address, IPv6 address, DNS address, or WINS address from the responder. After the responder has authenticated the initiator successfully, it assigns an IP address either from a local address pool or through RADIUS server. Depending on the configuration, this IP address is either assigned dynamically each time when a peer connects or assigned as a fixed IP address. If the RADIUS server responds with a framed pool, Junos OS assigns an IP address or information based on configuration from it's corresponding local pool. If you configure both local address pool and RADIUS server, the IP address allocated from RADIUS server takes precedence over the local pool. If you configure local IP address pool and the RADIUS server did not return any IP address, then local pool assigns the IP address to the request.

- Additional option, `none` introduced for `authentication-order`. See *authentication-order (Access Profile)*.

- RADIUS accounting start and stop messages inform the state of the tunnel or peer to the RADIUS server. These messages can be used for tracking purposes or notifications to subsystems such as a DHCP server.

  Ensure that the RADIUS server support accounting start or stop messages. Also ensure that both the firewall and the RADIUS server have appropriate settings to track these messages.

- Introduction of IPv6 support allows dual stack tunnels using configuration payload. During login process, IKE requests for both IPv4 and IPv6 addresses. AAA allow login only if all requested addresses have been allocated successfully. IKE terminates the negotiation if the requested IP is not allocated.

In a route-based VPN, secure tunnel (st0) interfaces operate in either point-to-multipoint or point-to-point mode. Address assignment through the IKEv2 configuration payload is now supported for point-to-multipoint or point-to-point mode. For point-to-multipoint interfaces, the interfaces must be numbered and the addresses in the configuration payload INTERNAL_IP4_ADDRESS attribute type must be within the subnetwork range of the associated point-to-multipoint interface.

You can configure a common password for IKEv2 configuration payload requests for an IKE gateway configuration. The common password in the range of 1 to 128 characters allows the administrator to define a common password. This password is used between the firewall and the RADIUS server when the firewall requesting an IP address on behalf of a remote IPsec peer using IKEv2 configuration payload. RADIUS server matches the credentials before it provides any IP information to the firewall for the configuration payload request. You can configure the common password using `config-payload-password` *`configured-password`* configuration statement at [`edit security ike gateway gateway-name aaa access-profile access-profile-name`] hierarchy level.

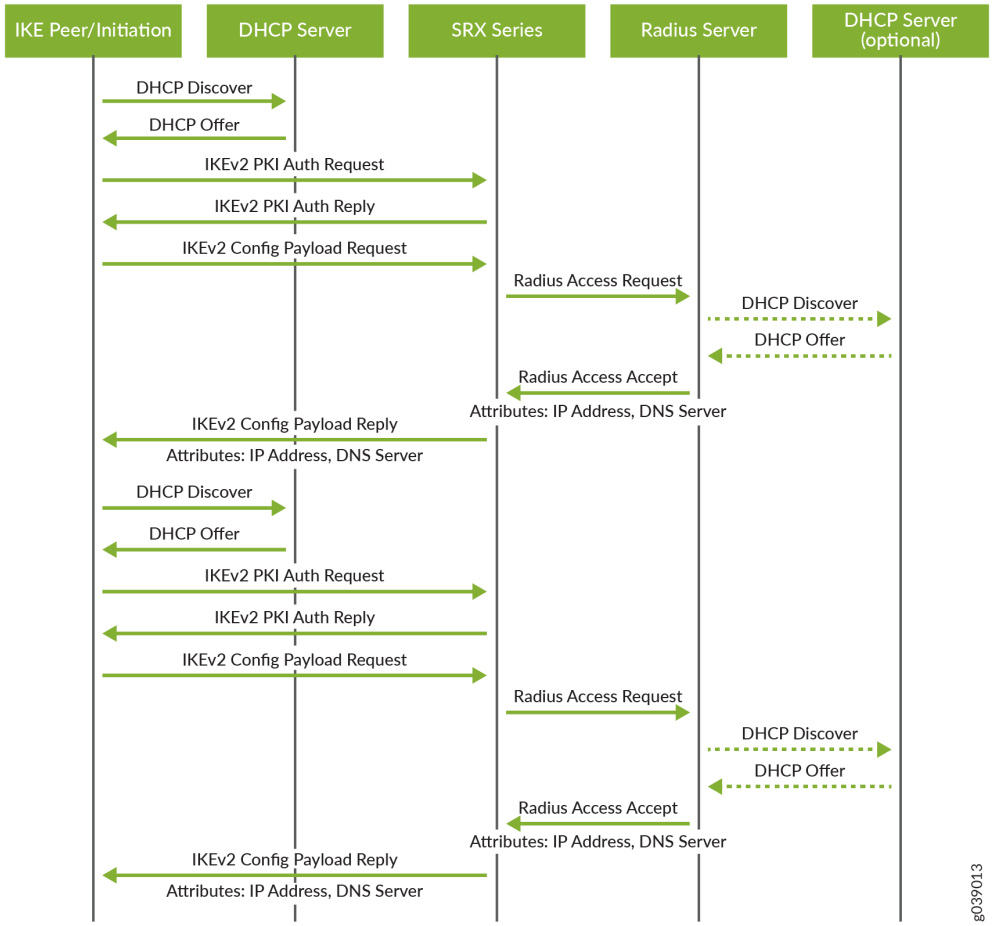
Both the firewall and the RADIUS server must have the same password configured and the radius server should be configured to use Password Authentication Protocol (PAP) as the authentication protocol. Without this, tunnel establishment will not be successful.



NOTE: The iked process does not depend on the DHCP pool or the address-allocation method. When the VPN tunnel is going down, Junos OS releases the assigned address.

Figure 7 on page 35 shows a typical workflow for a IKEv2 Configuration Payload.

Figure 7: Typical IKEv2 Configuration Payload Workflow



The IKEv2 configuration payload feature is supported for both point-to-multipoint secure tunnel (st0) interfaces and point-to-point interfaces. Point-to-multipoint interfaces must be numbered, and the addresses provided in the configuration payload must be within the subnetwork range of the associated point-to-multipoint interface.

Understanding Pico Cell Provisioning

IKEv2 configuration payload can be used to propagate provisioning information from an IKE responder, such as the firewall, to multiple initiators, such as LTE pico cell base stations in a cellular network. The pico cells ship from the factory with a standard configuration that allows them to connect to the firewall, but the pico cell provisioning information is stored on one or more provisioning servers within a protected network. The pico cells receive full provisioning information after establishing secure connections with the provisioning servers.

The workflow required to bootstrap and provision a pico cell and introduce it to service includes four distinct stages:

1. Initial addresses acquisition—The pico cell ships from the factory with the following information:

   - Configuration for the secure gateway tunnel to the firewall
   - Digital certificate issued by the manufacturer
   - Fully qualified domain name (FQDN) of the provisioning servers that lie within the protected network

   The pico cell boots up and acquires an address to be used for IKE negotiation from a DHCP server. A tunnel is then built to the secure gateway on the firewall using this address. An address for Operation, Administration, and Management (OAM) traffic is also assigned by the DHCP server for use on the protected network.

2. Pico cell provisioning—Using its assigned OAM traffic address, the pico cell requests its provisioning information—typically operator certificate, license, software, and configuration information—from servers within the protected network.

3. Reboot—The pico cell reboots and uses the acquired provisioning information to make it specific to the service provider's network and operation model.

4. Service provision—When the pico cell enters service, it uses a single certificate that contains distinguished name (DN) and subject alternative name values with a FQDN to build two tunnels to the secure gateway on the firewall: one for OAM traffic and the other for Third-Generation Partnership Project (3GPP) data traffic.

SEE ALSO

Example: Configuring NAT-T with Dynamic Endpoint VPN | **568**

IKE Proposal

The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

To configure an IKE proposal, include the `proposal` statement and specify a name at the [`edit security ike`] hierarchy level:

IKE Policy

An IKE policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address and the proposals needed for that connection. Depending on which authentication method is used, it defines the preshared key (PSK) for the given peer or the local certificate. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match. A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used. The configured PSK must also match its peer.

First, you configure one or more IKE proposals; then you associate these proposals with an IKE policy.

To configure an IKE policy, include the `policy` statement and specify a policy name at the [`edit security ike`] hierarchy level:

Rekeying and Reauthentication

IN THIS SECTION

- Overview | 37
- Supported Features | 38
- Limitations | 38

Overview

With IKEv2, rekeying and reauthentication are separate processes. Rekeying establishes new keys for the IKE security association (SA) and resets message ID counters, but it does not reauthenticate the peers. Reauthentication verifies that VPN peers retain their access to authentication credentials. Reauthentication establishes new keys for the IKE SA and child SAs; rekeys of any pending IKE SA or child SA are no longer needed. After the new IKE and child SAs are created, the old IKE and child SAs are deleted.

IKEv2 reauthentication is disabled by default. You enable reauthentication by configuring a reauthentication frequency value between 1 and 100. The reauthentication frequency is the number of IKE rekeys that occurs before reauthentication occurs. For example, if the configured reauthentication frequency is 1, reauthentication occurs every time there is an IKE rekey. If the configured

reauthentication frequency is 2, reauthentication occurs at every other IKE rekey. If the configured reauthentication frequency is 3, reauthentication occurs at every third IKE rekey, and so on.

You configure the reauthentication frequency with the `reauth-frequency` statement at the [`edit security ike policy` *`policy-name`*] hierarchy level. Reauthentication is disabled by setting the reauthentication frequency to 0 (the default). Reauthentication frequency is not negotiated by peers, and each peer can have its own reauthentication frequency value.

Supported Features

IKEv2 reauthentication is supported with the following features:

- IKEv2 initiators or responders
- Dead peer detection (DPD)
- Virtual routers and secure tunnel (st0) interfaces in virtual routers
- Network Address Translation traversal (NAT-T)
- Chassis clusters in active-active and active-passive mode
- In-service software upgrade (ISSU)
- Upgrade or insertion of a new Services Processing Unit (SPU) using the in-service hardware upgrade (ISHU) procedure

Limitations

Note the following caveats when using IKEv2 reauthentication:

- With NAT-T, a new IKE SA can be created with different ports from the previous IKE SA. In this scenario, the old IKE SA might not be deleted.
- In a NAT-T scenario, the initiator behind the NAT device can become the responder after reauthentication. If the NAT session expires, the NAT device might discard new IKE packets that might arrive on a different port. NAT-T keepalive or DPD must be enabled to keep the NAT session alive. For AutoVPN, we recommend that the reauthentication frequency configured on the spokes be smaller than the reauthentication frequency configured on the hub.
- Based on the reauthentication frequency, a new IKE SA can be initiated by either the initiator or the responder of the original IKE SA. Because Extensible Authentication Protocol (EAP) authentication and configuration payload require the IKE SA to be initiated by the same party as the original IKE SA, reauthentication is not supported with EAP authentication or configuration payload.

IKE Authentication (Certificate-Based Authentication)

IN THIS SECTION

- Multilevel Hierarchy for Certificate Authentication | 39

Multilevel Hierarchy for Certificate Authentication

Certificate-based authentication is an authentication method supported on firewalls during IKE negotiation. In large networks, multiple certificate authorities (CAs) can issue end entity (EE) certificates to their respective end devices. It is common to have separate CAs for individual locations, departments, or organizations.

When a single-level hierarchy for certificate-based authentication is employed, all EE certificates in the network must be signed by the same CA. All firewall devices must have the same CA certificate enrolled for peer certificate validation. The certificate payload sent during IKE negotiation only contains EE certificates.

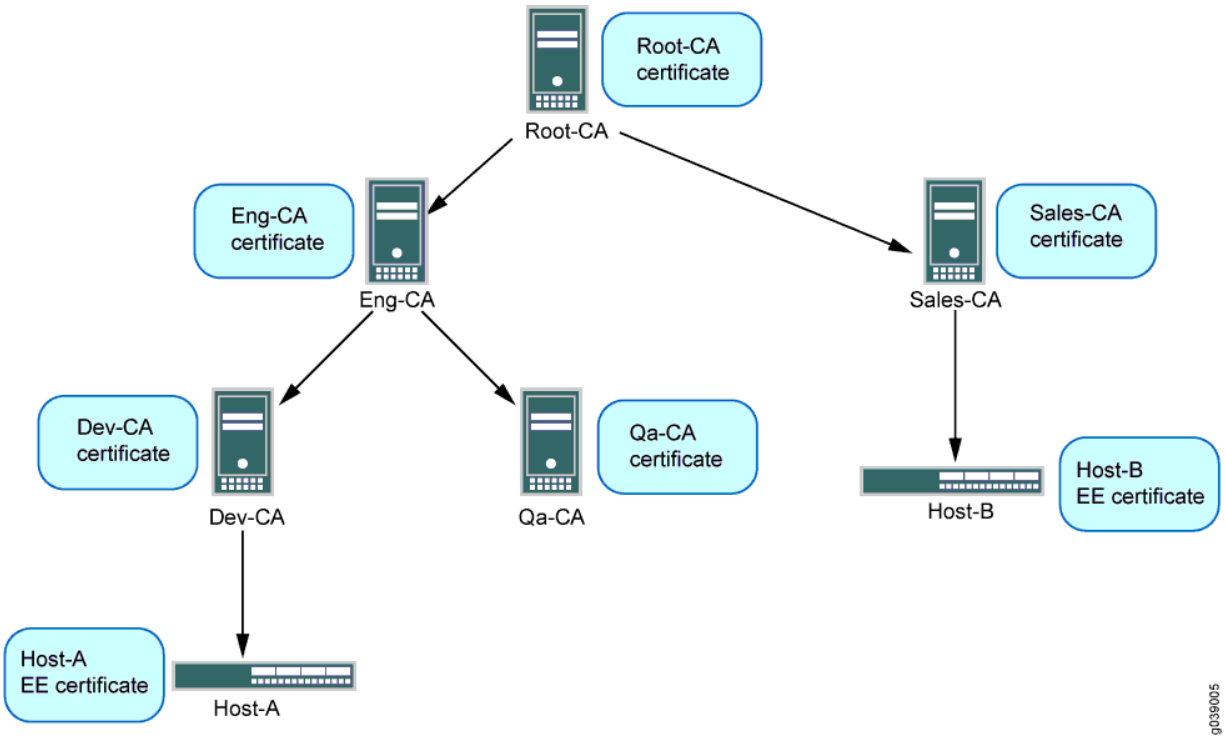
Alternatively, the certificate payload sent during IKE negotiation can contain a chain of EE and CA certificates. A *certificate chain* is the list of certificates required to validate a peer's EE certificate. The certificate chain includes the EE certificate and any CA certificates that are not present in the local peer.

The network administrator needs to ensure that all peers participating in an IKE negotiation have at least one common trusted CA in their respective certificate chains. The common trusted CA does not have to be the root CA. The number of certificates in the chain, including certificates for EEs and the topmost CA in the chain, cannot exceed 10.

Validation of a configured IKE peer can be done with a specified CA server or group of CA servers. With certificate chains, the root CA must match the trusted CA group or CA server configured in the IKE policy.

In the example CA hierarchy shown in Figure 8 on page 40, Root-CA is the common trusted CA for all devices in the network. Root-CA issues CA certificates to the engineering and sales CAs, which are identified as Eng-CA and Sales-CA, respectively. Eng-CA issues CA certificates to the development and quality assurance CAs, which are identified as Dev-CA and Qa-CA, respectively. Host-A receives its EE certificate from Dev-CA while Host-B receives its EE certificate from Sales-CA.

Figure 8: Multilevel Hierarchy for Certificate-Based Authentication



Each end device needs to be loaded with the CA certificates in its hierarchy. Host-A must have Root-CA, Eng-CA, and Dev-CA certificates; Sales-CA and Qa-CA certificates are not necessary. Host-B must have Root-CA and Sales-CA certificates. Certificates can be loaded manually in a device or enrolled using the Simple Certificate Enrollment Process (SCEP).

Each end device must be configured with a CA profile for each CA in the certificate chain. The following output shows the CA profiles configured on Host-A:

```
admin@host-A# show security
pki {
    ca-profile Root-CA {
        ca-identity Root-CA;
        enrollment {
            url “www.example.net/scep/Root/”;
        }
    }
    ca-profile Eng-CA {
        ca-identity Eng-CA;
        enrollment {
            url “www.example.net/scep/Eng/”;
        }
```

```
    }
    ca-profile Dev-CA {
        ca-identity Dev-CA;
        enrollment {
            url “www.example.net/scep/Dev/”;
        }
    }
}
```

The following output shows the CA profiles configured on Host-B:

```
admin@host-B# show security
pki {
    ca-profile Root-CA {
        ca-identity Root-CA;
        enrollment {
            url “www.example.net/scep/Root/”;
        }
    }
    ca-profile Sales-CA {
        ca-identity Sales-CA;
        enrollment {
            url “www.example.net/scep/Sales/”;
        }
    }
}
```

SEE ALSO

- Basic Elements of PKI in Junos OS
- Understanding Certificate Authority Profiles

Signature Authentication in IKEv2

SUMMARY

Read this topic to learn about the signature authentication method in IKEv2 and how it works in Junos OS.

IN THIS SECTION

- Implementation of Signature Authentication in IKEv2 | **42**
- Benefits | **43**

The Internet Key Exchange version 2 (IKEv2) protocol supports signature-based authentication that uses public key cryptography. In IKEv2, signature-based authentication supports one authentication method per signature algorithm. For example, the IKE peers use a separate authentication method for each of these digital signatures—RSA, Digital Signature Algorithm (DSA), and Elliptic Curve DSA (ECDSA). Each hash algorithm is tied to one signature for authentication. In IKEv2 proposal configuration, when you specify the authentication method, the device uses that method to authenticate the source of IKEv2 messages. It is difficult for the IKE peer to know which hash algorithm is associated with the signature. This process becomes even more cumbersome with the introduction of every new algorithm. See "Internet Key Exchange" on page 2 for more information about IKEv2.

You can address these challenges with the digital signature authentication method that is based on RFC 7427. This method is more generic compared to the signature-based authentication method, as the IKE peers can use any of the supported signature algorithms and also negotiate the signature hash algorithm. Read further to understand about the signature authentication in IKEv2.

Implementation of Signature Authentication in IKEv2

In addition to supporting signature-based authentication on a per algorithm basis, Junos OS also supports the digital signature authentication method described in RFC 7427. In this method, the IKEv2 authentication payload indicates not only the type of public key, but also the hash algorithm that the device uses to generate the signature. The device uses the SIGNATURE_HASH_ALGORITHM notification to notify its peer about the RFC 7427 support and provide the list of supported hash algorithms.

To use the feature, you must configure the digital signature authentication method on your device. For more information about the digital signature authentication method, see *proposal (Security IKE)*. You can use the configuration option `signature-hash-algorithm` to define the specific signature hash algorithms that must match in the hierarchical order with each of the received signature hash algorithms. If you do not specify the signature hash algorithm, the device matches the received signature hash algorithm from the default list of all the supported hash algorithms. See *Signature Hash Algorithm (Security IKE)*.

In Junos OS, the authentication method involves the following steps that are defined in the IKEv2 message flow. See RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)* for more details.

- Both the IKE peers initially notify the list of supported hash algorithms to each other. Your device sends the SIGNATURE_HASH_ALGORITHM payload in the IKE_SA_INIT message to the peer device and receives a response. The peers then negotiate a signature hash algorithm.
- In the IKE_AUTH message, the peers exchange the digital signature authentication method.

The device uses the default certificate authentication method in either of these scenarios:

- Responder doesn't support RFC 7427.
- Initiator doesn't support the received hash algorithm.

Benefits

- Flexible—Includes traditional and new digital signatures.
- Ease of use—Integrates into the existing public key infrastructure (PKI).
- Robust solution—Performs better identity verification compared to the signature-based authentication method, improving overall security and reliability of the IKE peers.

SEE ALSO

proposal (Security IKE)

IKE Protection from DDoS Attacks

IN THIS SECTION

- DDoS Vulnerabilities on IKE Implementations | 44
- Protection Against DDoS Attacks | 45
- Ways to Monitor DDoS Attacks | 46

Read this topic to understand how Juniper protects IPsec VPN IKE implementations from distributed denial-of-service (DDoS) attacks and monitors the implementations.

Denial of service (DoS) is one of the most common yet serious attacks in an insecure IPsec VPN network. A DoS attack provides a quick and easy way to grab the network as it doesn't require much toehold in the network infrastructure. Cyberattackers choose this method to take control of the network.

What happens in a DoS attack?

The attacker tries to flood and gradually crash the network with too much traffic, depleting the network resources, and further taking control of the device resources such as memory and CPU. If the attacker tries to control using multiple orchestrated systems, synchronously attacking a single target, it is called Distributed DoS (DDoS) attacks.

DDoS Vulnerabilities on IKE Implementations

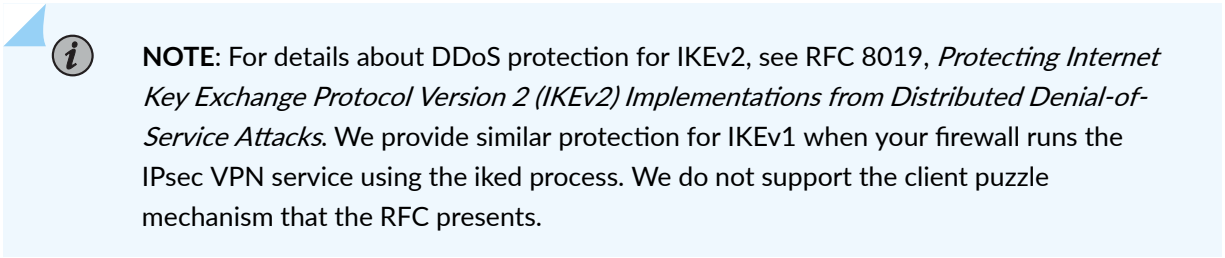
When the remote peer (initiator) sends an SA_INIT message, the local peer (responder) replies and allocates memory for the message structure. We call the session a *half-open IKE session* until authentication happens with the help of the IKE_AUTH message. After the peers establish an IKE security association (SA), the session becomes a *full-open IKE session*.

To understand the IKEv2 DDoS vulnerabilities, let's look at some of the ways an attacker can create an easy attack vector against the IKE SAs:

- Send a large number of SA_INIT messages (without IKE_AUTH messages) for which the attacker can create half-open IKE security association structures. The attack causes the device to utilize the resources and run out of memory.
- Send a large number of junk IKE_AUTH packets with the correct SPI_i and SPI_r on the initiator and the responder, respectively. The device runs out of memory while trying to decrypt the packets.
- Send SA_INIT packets continuously. The device runs out of memory while trying to generate keys for the encrypted packets.
- Send a large number of rekey requests per second during full open IKE sessions.
- Send a large number of messages with distinct message identifiers (IDs). The device queues all incoming IKE messages and runs out of memory.

How to safeguard the IKE implementations

We provide a robust infrastructure to mitigate and to monitor DDoS attacks for both IKEv1 and IKEv2 protocols. You can protect against the attacks on IKE implementations when your firewall runs the iked process (in the **junos-ike** package) for the IPsec VPN service.



NOTE: For details about DDoS protection for IKEv2, see RFC 8019, *Protecting Internet Key Exchange Protocol Version 2 (IKEv2) Implementations from Distributed Denial-of-Service Attacks*. We provide similar protection for IKEv1 when your firewall runs the IPsec VPN service using the iked process. We do not support the client puzzle mechanism that the RFC presents.

Protection Against DDoS Attacks

You can enable multiple defense mechanisms against the DDoS attacks during the IKE security association creation process. These mechanisms include configuration of rate limiting and a retention period for the half-open IKE SAs, and further managing the incoming exchange rates for the rekey requests. We provide the following measures to ensure protection against DDoS attacks on IKE SAs:

- Protection measures for half-open IKE SAs:
  - The responder does not allow configuration of half-open IKE SAs for a certain duration. You can set this limit so that the responder does not configure the SAs until it reaches the timeout duration. For more details, see *session (Security IKE)* for the option `timeout`.
  - You can set a limit on the maximum allowed half-open IKE SAs on the responder. When the total number of half-open IKE SAs reaches the maximum count, the responder rejects new connections for both IKEv1 and IKEv2 SAs. For more details, see the `max-count` option in *session (Security IKE)*.
  - The responder enforces threshold values on the session count for half-open IKE SAs. When the total number of half-open IKE SAs reaches the threshold values:
    - In case of IKEv2 SAs, the responder invokes cookie mechanism for any new connections.
    - In case of IKEv1 SAs, the responder rejects new connections.

    For more details, see the `thresholds`, `send-cookie` and `reduce-timeout` options in *session (Security IKE)*.
  - The responder can discard duplicate sessions. For more details, see the `discard-duplicate` option in *session (Security IKE)*.
  - You can set backoff timeouts for authentication failure and initiation failure phases. For more details, see *session (Security IKE)* for the options `backoff-timeouts`, `init-phase-failure` and `auth-phase-failure`.
- For full open IKE SAs:
  - You can configure maximum incoming rekey request rates to throttle the requests in a scaled scenario. For more details, see the `incoming-exchange-max-rates` option in *session (Security IKE)*.

- The responder can block an incoming IKE session from peers based on the peer IKE ID. For more details, see *blocklists (Security IKE)*.
- For dynamic gateways, you can set a limit for the number of connections at the IKE gateway configuration level using the option `connections-limit`. For more details, see gateway (Security IKE).

For further details on how to configure these options, see "Configure Protection Against IKE DDoS Attacks" on page 47.

We do not support:

- DDoS protection with IPsec VPN service based on the kmd process.
- Protection against *Hash and URL* certificate encoding attacks as we do not support these encoding types.

Ways to Monitor DDoS Attacks

We provide the following mechanisms to monitor the DDoS attacks:

- Use the `show security ike security-associations` command to list down all the matured and non-matured IKE security associations. For more details, see show security ike security-associations.
- Use the `show security ike stats` command to display global IKE statistics of the IPsec VPN tunnel, such as in-progress, established, expired statistics. For more details, see show security ike stats.
- Use the `show security ike active-peer` command to display details of the successful IKE negotiations with the remote peers. For more details, see show security ike active-peer.
- Use the `show security ike peers in-progress` command to display the details of in progress IKE SAs, including the half open IKE SAs. For more details, see *show security ike peers*. You can also see the details of the blocked, failed, and backoff peers using this command.
- Use the `clear security ike peers` command to clear the IKE peers that are backed off, blocked, failed or in progress. For more details, see *clear security ike peers*.
- To delete an existing IKE security association with peers that needs to be blocked from further communications, use the `clear security ike security-associations` command. For more details, see clear security ike security-associations.
- The iked system log (syslog) messages *IKE_GATEWAY_PEER_BLOCKED*, *IKE_GATEWAY_PEER_BACKOFF* and *IKE_GATEWAY_PEER_FAILED* provide details about the blocked, backed-off, and failed IKE negotiations with the remote peers, respectively.
- For additional security, Junos OS provides services to users for protection against packet-based system attacks such as filtering, session count, and rate limiting. For more details, see the show

firewall command and the ids-option statement at the [edit security screen ids-option *screen-name*] hierarchy level.

Configure Protection Against IKE DDoS Attacks

SUMMARY

See this section to understand how to configure protection against DDoS attacks on IKE protocol.

IN THIS SECTION

- Configure the IKE Session for Half Open IKE SAs | **48**
- Configure the IKE Session for Full Open IKE SAs | **50**
- Configure the IKE Session Blocklists | **51**

Preprequisites

Before configuring protection against the IKE DDoS attacks, ensure that you meet the following prerequisites:

- Firewall that supports junos-ike package to run the IPsec VPN service using the iked process.
- Firewall that serves as the local endpoint (the responder) is reachable to the remote IKE peer (the initiator).
- An IKE policy that can associate an IKE blocklist.

Following actions are involved in configuring protection against the IKE DDoS attacks:

- Manage the incoming half open IKE SAs.
- Manage the incoming full open IKE SAs.
- Configure multiple blocking methods in order to block the incoming IKE sessions from various peers and associate one of the blocklists with an IKE peer.

See the following tasks to configure these actions.

Configure the IKE Session for Half Open IKE SAs

IN THIS SECTION

- Overview | 48
- Configuration | 48

Overview

Ensure you meet all the prerequisites discussed above.

In this section, you'll see how to configure the timeouts, maximum count and thresholds for the half open IKE SAs. The configuration changes are applicable to the new sessions, while the existing sessions continue to use the default values when not configured explicitly earlier. The scope of these configurations at the [`edit security ike session half-open`] hierarchy level is applicable at global level and not per peer level.

Configuration

- To set the responder lifetime parameter using the option `timeout` *seconds*:

  ```
  [edit]
  user@host# set security ike session half-open timeout 150
  ```

  During this period, the responder does not allow the configuration of half open IKE SAs till it reaches the timeout duration. The initiator can continue to have 60 seconds timeout duration irrespective of the responder's configuration.

- To set the responder maximum count parameter using the option `max-count` *value*:

  ```
  [edit]
  user@host# set security ike session half-open max-count 1000
  ```

  The option sets the maximum numbers of half open IKE sessions on the responder. The default value is 300, if you do not specify. The `max-count` configuration disables all thresholds. In such cases, you need to explicilty configure thresholds to enforce them.

- Specify the different types of actions using the option `threshold` when the responder's session count reaches the limit.

- To set the minimum number of half open IKE sessions to enforce cookie action using the option `send-cookie` *`count`*:

```
[edit]
user@host# set security ike session half-open threshold send-cookie 500
```

  This specifies the threshold limit from which the responder requests the remote peers to retry the session initiation with a cookie sent back to the peer in the initial response. Here, when the half open IKE session count limit reaches 500, the iked process employs a cookie mechanism for the new IKE sessions.

- To set the minimum number of half open IKE sessions to enforce reduced timeout action using the option `reduced-timeout` *`count`* `timeout` *`seconds`*:

```
[edit]
user@host# set security ike session half-open threshold reduce-timeout 600 timeout 100
```

  This specifies the limit from which the iked process reduces the lifetime of the new half open IKE SAs. Once the half open responder IKE session count reduces below the threshold, the half open responder IKE sessions use the default timeout value again.

- To set the option `discard-duplicate` in order to discard the duplicate half open IKE sessions without sending response back to the initiator:

```
[edit]
user@host# set security ike session half-open discard-duplicate
```

  For a duplicate session initiation request (SA_INIT) that comes from the same peer, with a different initiator cookie for which there is no IKE SA while the negotiation is in progress, the responder discards the packet.

- You can set the backoff timeouts using the option `backoff-timeouts`.

  This gives some time for the remote peer to back off in the event of a session initiation failure, ensuring that the same peer cannot initiate a new session during that period. After the backoff timeout, the peer can initiate a new session. The session initiation can fail at two phases—the initialization phase and the authentication phase.

- To set the backoff timeout when there's a failure during the IKE_AUTH phase using the option `backoff-timeouts auth-phase-failure value`:

```
[edit]
user@host# set security ike session half-open backoff-timeouts auth-phase-failure 150
```

  When you configure `auth-phase-failure`, any remote peer that is blocklisted, backsoff even when you do not configure backoff as an action for the target blocklist rule. The timeout for the backoff is the one configured for `auth-phase-failure`. In this example, the device initiates a new session after 150 seconds. To overwrite this backoff timeout for a particular rule, you can explicitly configure the `backoff` action for the blocklist `rule` mentioning the timeout at the [`edit security ike blocklists blocklist1 rule rule-name then backoff timeout-value`] `hierarchy level`.

- To set the backoff timeout when there's a failure during the SA_INIT phase using the option `backoff-timeouts init-phase-failure value`:

```
[edit]
user@host# set security ike session half-open backoff-timeouts init-phase-failure 160
```

  In this example, the device initiates a new session after 160 seconds.

Configure the IKE Session for Full Open IKE SAs

IN THIS SECTION

- Overview | 50
- Configuration | 51

Overview

Ensure you meet all the prerequisites discussed above.

In this section, you'll see how to configure various incoming request rates for the full open IKE SAs using the option `incoming-exchange-max-rates` at the [`edit security ike session full-open`] hierarchy level.

Configuration

Configure the `incoming-exchange-max-rates` option to set the maximum rates for various exchanges initiated by the remote peer after establishing an IKE SA. You can configure three types of exchange rates—IKE rekey, IPsec rekey and keepalive (also known as Dead Peer Detection).

- To set the incoming peer initiated IKE rekey maximum rate using the option `incoming-exchange-max-rates ike-rekey` *value*:

  ```
  [edit]
  user@host# set security ike session full-open incoming-exchange-max-rates ike-rekey 200/60
  ```

  The option is applicable to IKEv2 rekey on a per peer basis with an existing peer where an IKE SA is already present.

- To set the incoming peer initiated IPsec SA rekey maximum rate using the option `incoming-exchange-max-rates ipsec-rekey` *value*:

  ```
  [edit]
  user@host# set security ike session full-open incoming-exchange-max-rates ipsec-rekey 100/60
  ```

  The limit is applicable on a per tunnel basis.

- To set the incoming peer initiated keepalive maximum rate using the option `incoming-exchange-max-rates keepalive` *value*:

  ```
  [edit]
  user@host# set security ike session full-open incoming-exchange-max-rates keepalive 60/60
  ```

  The limit is applicable on a per peer basis.

Configure the IKE Session Blocklists

IN THIS SECTION

- Overview | 52
- Configuration | 52

Overview

Ensure you meet all the prerequisites discussed above.

To block an incoming IKE session from peers based on the peer IKE ID, you need to configure one or more blocklists. Each blocklist contains one or more rules. Each rule has a match criteria and an action.

Consider the following criteria when configuring the blocklists:

- Blocklist rules are applicable on the new IKE SAs that are being negotiated and doesn't affect the existing IKE SAs. At the peer authentication stage, the device applies the blocklist rules.
- The order of application of the rules is dependent on the sequence in which these rules are listed.
- Configure the match criteria based on the role (initiator or responder), an ID type (IPv4 or IPv6 address, hostname, distinguished name, email ID, or a key ID) and an ID pattern which is a regular expression to match the IKE ID.
- You can configure each rule with an ID type. This allows you to configure different IKE IDs for different rules within the same blocklist.
- Configure an action to either discard or reject the peer connection. Based on the match, the device applies an action. Optionally, you can set the backoff timer with these actions.
- Refer the blocklist in the IKE policy which is associated to the IKE gateway.
- Each IKE gateway supports only one type of remote IKE ID type. If you attach a blocklist to a gateway and configure it with rules containing different IKE IDs, the gateway will apply and match only the rules whose IKE ID type is the same as the one configured for the IKE gateway.
- The tunnel setup rate metrices with blocklists that are attached to the IKE gateways is based on the number of rules configured in the blocklists.

In this section, you'll see how to configure blocklists and associate the blocklist to an IKE policy.

Configuration

- To create a blocklist with multiple rules:
  - Create a blocklist.

```
[edit]
user@host# set security ike blocklists blocklist1 description block_from_remote
```

- Create one or more rules.

```
[edit]
user@host# set security ike blocklists blocklist1 rule rule1 description rule_1
user@host# set security ike blocklists blocklist1 rule rule2 description rule_2
```

You can create multiple such blocklists and its rules.

- To configure the match criteria and specify the action:
  - Configure the match criteria for the *rule1* in the blocklist *blocklist1*.

```
[edit]
user@host# set security ike blocklists blocklist1 rule rule1 match role initiator
user@host# set security ike blocklists blocklist1 rule rule1 match id-type hostname
user@host# set security ike blocklists blocklist1 rule rule1 match id-pattern "peer.*
\.example\.net"
user@host# set security ike blocklists blocklist1 rule rule2 match role initiator
user@host# set security ike blocklists blocklist1 rule rule2 match id-type user-at-hostname
user@host# set security ike blocklists blocklist1 rule rule2 match id-pattern
"hr.example.com"
```

To configure blocklists using a group of IKE IDs or partial IKE IDs, use the `id-pattern` *`value`* with a suffix or prefix. For example, you can use the value *.example.com, when you need to match hr.example.com, finance.example.com, admin.example.com. In the rule *rule1*, the device looks for the hostname that matches the pattern *peer.*\.example\.net*. Here peer.example.net, peer.1.example.net, and peer.uplink.example.net are a few potential matches. In the rule *rule2*, the device looks for the email address that matches the pattern *hr.example.com*. Similarly, you can configure another match criteria for the other rules based on different `id-type` or `id-pattern`. These patterns use the standard regular expression.

  - Specify the action for a match:

```
[edit]
user@host# set security ike blocklists blocklist1 rule rule1 then reject
user@host# set security ike blocklists blocklist1 rule rule1 then backoff 60
user@host# set security ike blocklists blocklist1 rule rule2 then discard
user@host# set security ike blocklists blocklist1 rule rule2 then backoff 100
```

- To associate a blocklist with an IKE peer:

- Configure the blocklist *blocklist1* in the IKE policy *ike_policy1*.

```
[edit]
user@host# set security ike policy ike_policy1 blocklist blocklist1
```

RELATED DOCUMENTATION

IKE Protection from DDoS Attacks

session (Security IKE)

blocklists (Security IKE)

Example: Configuring a Device for Peer Certificate Chain Validation

IN THIS SECTION

- Requirements | 54
- Overview | 55
- Configuration | 56
- Verification | 64
- IKE and IPsec SA Failure for a Revoked Certificate | 65

This example shows how to configure a device for certificate chains used to validate peer devices during IKE negotiation.

Requirements

Before you begin, obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

Overview

IN THIS SECTION

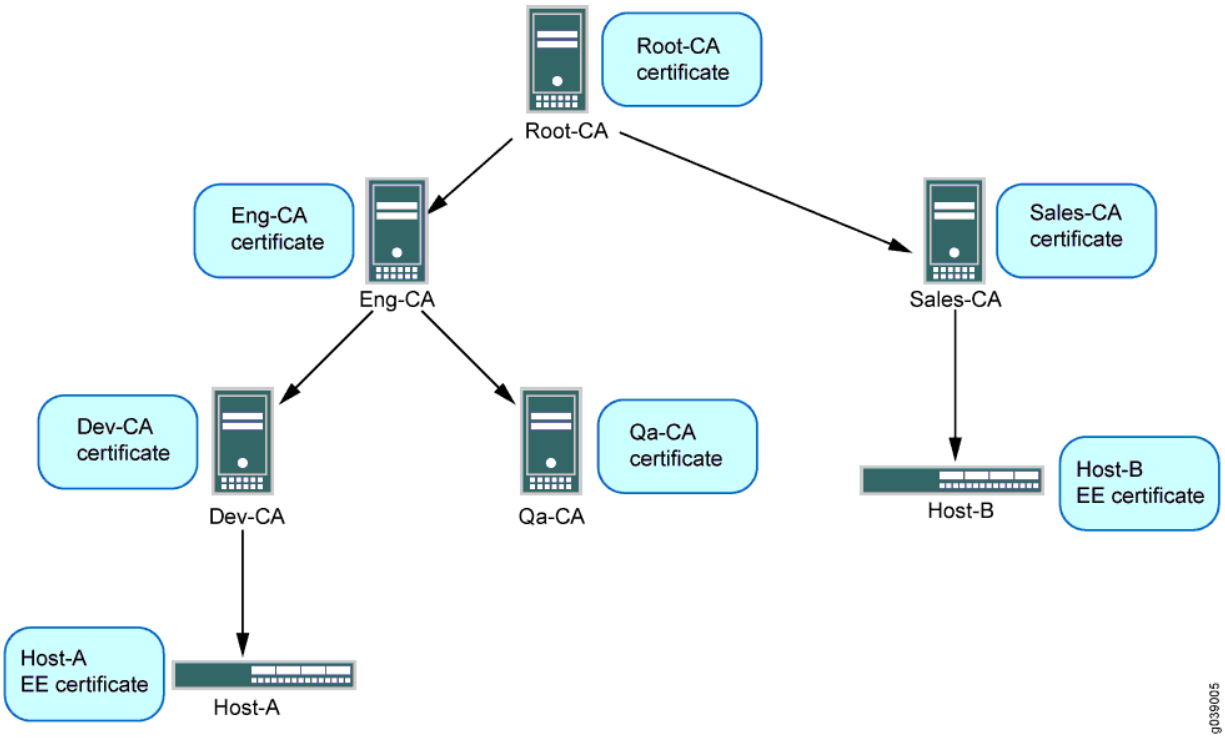
- Topology | 55

This example shows how to configure a local device for certificate chains, enroll CA and local certificates, check the validity of enrolled certificates, and check the revocation status of the peer device.

Topology

This example shows the configuration and operational commands on Host-A, as shown in Figure 9 on page 55. A dynamic CA profile is automatically created on Host-A to allow Host-A to download the CRL from Sales-CA and check the revocation status of Host-B's certificate.

Figure 9: Certificate Chain Example



The IPsec VPN configuration for Phase 1 and Phase 2 negotiation is shown for Host-A in this example. The peer device (Host-B) must be properly configured so that Phase 1 and Phase 2 options are successfully negotiated and security associations (SAs) are established. See "Configuring Remote IKE IDs for Site-to-Site VPNs" on page 118 for examples of configuring peer devices for VPNs.

Configuration

IN THIS SECTION

- Configure CA Profiles | 56
- Enroll Certificates | 58
- Configure IPsec VPN Options | 61

To configure a device for certificate chains:

Configure CA Profiles

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile Root-CA ca-identity CA-Root
set security pki ca-profile Root-CA enrollment url http://198.51.100.230:8080/scep/Root/
set security pki ca-profile Root-CA revocation-check crl
set security pki ca-profile Eng-CA ca-identity Eng-CA
set security pki ca-profile Eng-CA enrollment url http://198.51.100.230:8080/scep/Eng/
set security pki ca-profile Eng-CA revocation-check crl
set security pki ca-profile Dev-CA ca-identity Dev-CA
set security pki ca-profile Dev-CA enrollment url http://198.51.100.230:8080/scep/Dev/
set security pki ca-profile Dev-CA revocation-check crl
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure CA profiles:

1. Create the CA profile for Root-CA.

```
[edit security pki]
user@host# set ca-profile Root-CA ca-identity CA-Root
user@host# set ca-profile Root-CA enrollment url http://198.51.100.230:8080/scep/Root/
user@host# set ca-profile Root-CA revocation-check crl
```

2. Create the CA profile for Eng-CA.

```
[edit security pki]
user@host# set ca-profile Eng-CA ca-identity Eng-CA
user@host# set ca-profile Eng-CA enrollment url http://198.51.100.230:8080/scep/Eng/
user@host# set ca-profile Eng-CA revocation-check crl
```

3. Create the CA profile for Dev-CA.

```
[edit security pki]
user@host# set ca-profile Dev-CA ca-identity Dev-CA
user@host# set ca-profile Dev-CA enrollment url http://198.51.100.230:8080/scep/Dev/
user@host# set ca-profile Dev-CA revocation-check crl
```

Results

From configuration mode, confirm your configuration by entering the `show security pki` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security pki
ca-profile Root-CA {
    ca-identity Root-CA;
    enrollment {
        url "http:/;/198.51.100.230:8080/scep/Root/";
    }
    revocation-check {
        crl ;
    }
```

```
}
ca-profile Eng-CA {
    ca-identity Eng-CA;
    enrollment {
        url "http:/;/198.51.100.230:8080/scep/Eng/";
    }
    revocation-check {
        crl ;
    }
}
ca-profile Dev-CA {
    ca-identity Dev-CA;
    enrollment {
        url "http:/;/198.51.100.230:8080/scep/Dev/";
    }
    revocation-check {
        crl ;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Enroll Certificates

Step-by-Step Procedure

To enroll certificates:

1. Enroll the CA certificates.

```
user@host> request security pki ca-certificate enroll ca-profile Root-CA
```

```
user@host> request security pki ca-certificate enroll ca-profile Eng-CA
```

```
user@host> request security pki ca-certificate enroll ca-profile Dev-CA
```

Type **yes** at the prompts to load the CA certificate.

2. Verify that the CA certificates are enrolled in the device.

```
user@host> show security pki ca-certificate ca-profile Root-CA
Certificate identifier: Root-CA
        Issued to: Root-CA, Issued by: C = us, O = example, CN = Root-CA
        Validity:
          Not before: 08-14-2012 22:19
          Not after: 08-13-2017 22:19
        Public key algorithm: rsaEncryption(2048 bits)
```

```
user@host> show security pki ca-certificate ca-profile Eng-CA
Certificate identifier: Eng-CA
        Issued to: Eng-CA, Issued by: C = us, O = example, CN = Root-CA
        Validity:
          Not before: 08-15-2012 01:02
          Not after: 08-13-2017 22:19
        Public key algorithm: rsaEncryption(2048 bits)
```

```
user@host> show security pki ca-certificate ca-profile Dev-CA
        Certificate identifier: Dev-CA
        Issued to: Dev-CA, Issued by: C = us, O = example, CN = Eng-CA
        Validity:
          Not before: 08-15-2012 17:41
          Not after: 08-13-2017 22:19
        Public key algorithm: rsaEncryption(2048 bits)
```

3. Verify the validity of the enrolled CA certificates.

```
user@host> request security pki ca-certificate verify ca-profile Root-CA
CA certificate Root-CA verified successfully
```

```
user@host> request security pki ca-certificate verify ca-profile Eng-CA
CA certificate Eng-CA verified successfully
```

```
user@host> request security pki ca-certificate verify ca-profile Dev-CA
CA certificate Dev-CA verified successfully
```

4. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Host-A type rsa size 1024
```

5. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll certificate-id       Host-A ca-
profile Dev-CA challenge-password example domain-name host-a.example.net email host-
a@example.net subject DC=example,CN=Host-A,       OU=DEV,O=PKI,L=Sunnyvale,ST=CA,C=US
```

6. Verify that the local certificate is enrolled in the device.

```
user@host> show security pki local-certificate
Issued to: Host-A, Issued by: C = us, O = example, CN = Dev-CA
        Validity:
          Not before: 09-17-2012 22:22
          Not after: 08-13-2017 22:19
        Public key algorithm: rsaEncryption(1024 bits)
```

7. Verify the validity of the enrolled local certificate.

```
user@host> request security pki local-certificate verify certificate-id Host-A
Local certificate Host-A verification success
```

8. Check the CRL download for configured CA profiles.

```
user@host> show security pki crl
     CA profile: Root-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Root-CA
        Effective date: 09- 9-2012 13:08
        Next update: 09-21-2012 02:55

      CA profile: Eng-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Eng-CA
        Effective date: 08-22-2012 17:46
        Next update: 10-24-2015 03:33

      CA profile: Dev-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Dev-CA
        Effective date: 09-14-2012 21:15
        Next update: 09-26-2012 11:02
```

Configure IPsec VPN Options

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_cert_prop_01 authentication-method rsa-signatures
set security ike proposal ike_cert_prop_01 dh-group group5
set security ike proposal ike_cert_prop_01 authentication-algorithm sha1
set security ike proposal ike_cert_prop_01 encryption-algorithm aes-256-cbc
set security ike policy ike_cert_pol_01 mode main
set security ike policy ike_cert_pol_01 proposals ike_cert_prop_01
set security ike policy ike_cert_pol_01 certificate local-certificate Host-A
set security ike gateway ike_cert_gw_01 ike-policy ike_cert_pol_01
set security ike gateway ike_cert_gw_01 address 192.0.2.51
set security ike gateway ike_cert_gw_01 external-interface ge-0/0/1.0
set security ike gateway ike_cert_gw_01 local-identity 192.0.2.31
```

```
set security ipsec proposal ipsec_prop_01 protocol esp
set security ipsec proposal ipsec_prop_01 authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop_01 encryption-algorithm 3des-cbc
set security ipsec proposal ipsec_prop_01 lifetime-seconds 300
set security ipsec policy ipsec_pol_01 proposals ipsec_prop_01
set security ipsec vpn ipsec_cert_vpn_01 bind-interface st0.1
set security ipsec vpn ipsec_cert_vpn_01 ike gateway ike_cert_gw_01
set security ipsec vpn ipsec_cert_vpn_01 ike ipsec-policy ipsec_pol_01
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec VPN options:

1. Configure Phase 1 options.

```
[edit security ike proposal ike_cert_prop_01]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ike_cert_pol_01]
user@host# set mode main
user@host# set proposals ike_cert_prop_01
user@host# set certificate local-certificate Host-A
[edit security ike gateway ike_cert_gw_01]
user@host# set ike-policy ike_cert_pol_01
user@host# set address 192.0.2.51
user@host# set external-interface ge-0/0/1.0
user@host# set local-identity 192.0.2.31
```

2. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_prop_01]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 300
[edit security ipsec policy ipsec_pol_01]
```

```
user@host# set proposals ipsec_prop_01
[edit security ipsec vpn ipsec_cert_vpn_01]
user@host# set bind-interface st0.1
user@host# set ike gateway ike_cert_gw_01
user@host# set ike ipsec-policy ipsec_pol_01
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` and `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike_cert_prop_01 {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
    policy ike_cert_pol_01 {
        mode main;
        proposals ike_cert_prop_01;
        certificate {
            local-certificate Host-A;
        }
    }
    gateway ike_cert_gw_01 {
        ike-policy ike_cert_pol_01;
        address 192.0.2.51;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec_prop_01 {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 300;
}
    policy ipsec_pol_01 {
```

```
        proposals ipsec_prop_01;
    }
    vpn ipsec_cert_vpn_01 {
        bind-interface st0.1;
        ike {
            gateway ike_cert_gw_01;
            ipsec-policy ipsec_pol_01;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Phase 1 Status | **64**
- Verifying IPsec Phase 2 Status | **65**

If certificate validation is successful during IKE negotiation between peer devices, both IKE and IPsec security associations (SAs) are established.

The IKE SA is UP if the certificate is valid. The IKE SA is DOWN and IPSEC SA is formed if the certificate is revoked, only if revocation check is configured on the peer device

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

Enter the **show security ike security-associations** command from operational mode.

```
user@host> show security ike security-associations
    Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
    2090205 DOWN     285feacb50824495  59fca3f72b64da10  Main           192.0.2.51
```

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

Enter the **show security ipsec security-associations** command from operational mode.

```
user@host> show security ipsec security-associations
      Total active tunnels: 1
      ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
      <131073 ESP:3des/sha1 a4756de9 207/  unlim   -   root 500   192.0.2.51
      >131073 ESP:3des/sha1 353bacd3 207/  unlim   -   root 500   192.0.2.51
```

IKE and IPsec SA Failure for a Revoked Certificate

IN THIS SECTION

- Checking for Revoked Certificates | 65

Checking for Revoked Certificates

Problem

If certificate validation fails during IKE negotiation between peer devices, check to make sure that the peer's certificate has not been revoked. A dynamic CA profile allows the local device to download the CRL from the peer's CA and check the revocation status of the peer's certificate. To enable dynamic CA profiles, the `revocation-check crl` option must be configured on a parent CA profile.

Solution

To check the revocation status of a peer's certificate:

1. Identify the dynamic CA profile that will show the CRL for the peer device by entering the **show security pki crl** command from operational mode.

```
user@host> show security pki crl
     CA profile: Root-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Root-CA
        Effective date: 09- 9-2012 13:08
        Next update: 09-21-2012 02:55

      CA profile: Eng-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Eng-CA
        Effective date: 08-22-2012 17:46
        Next update: 10-24-2015 03:33

      CA profile: Dev-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Dev-CA
        Effective date: 09-14-2012 21:15
        Next update: 09-26-2012 11:02

      CA profile: dynamic-001
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Sales-CA
        Effective date: 09-14-2012 21:15
        Next update: 09-26-2012 11:02
```

The CA profile `dynamic-001` is automatically created on Host-A so that Host-A can download the CRL from Host-B's CA (Sales-CA) and check the revocation status of the peer's certificate.

2. Display CRL information for the dynamic CA profile by entering the **show security pki crl ca-profile dynamic-001 detail** command from operational mode.

   Enter

```
user@host> show security pki crl ca-profile dynamic-001 detail
    CA profile: dynamic-001
      CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Sub11
        Effective date: 09-19-2012 17:29
        Next update: 09-20-2012 01:49
```

```
        Revocation List:
          Serial number              Revocation date
          10647C84                   09-19-2012 17:29 UTC
```

Host-B's certificate (serial number 10647084) has been revoked.

SEE ALSO

Basic Elements of PKI in Junos OS

Understanding Certificate Authority Profiles

IKEv2 Fragmentation

IN THIS SECTION

- Message Fragmentation | 67
- Configuration | 68
- Caveats | 68

Message Fragmentation

IKEv2 message fragmentation, as described in RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*, allows IKEv2 to operate in environments where IP fragments might be blocked and peers would not be able to establish an IPsec security association (SA). IKEv2 fragmentation splits a large IKEv2 message into a set of smaller ones so that there is no fragmentation at the IP level. Fragmentation takes place before the original message is encrypted and authenticated, so that each fragment is separately encrypted and authenticated. On the receiver, the fragments are collected, verified, decrypted, and merged into the original message.

For IKEv2 fragmentation to occur, both VPN peers *must* indicate fragmentation support by including the IKEV2_FRAGMENTATION_SUPPORTED notification payload in the IKE_SA_INIT exchange. If both peers indicate fragmentation support, it is up to the initiator of the message exchange to determine whether or not IKEv2 fragmentation is used.

On firewalls, a maximum of 32 fragments are allowed per IKEv2 message. If the number of IKEv2 message fragments to be sent or received exceeds 32, the fragments are dropped and the tunnel is not established. Retransmission of individual message fragments is not supported

Configuration

On firewalls, IKEv2 fragmentation is enabled by default for IPv4 and IPv6 messages. To disable IKEv2 fragmentation, use the `disable` statement at the [`edit security ike gateway gateway-name fragmentation`] hierarchy level. You can also use the `size` statement to configure the size of the packet at which messages are fragmented; the packet size ranges from 500 to 1300 bytes. If `size` is not configured, the default packet size is 576 bytes for IPv4 traffic and 1280 bytes for IPv6 traffic. An IKEv2 packet that is larger than the configured packet size is fragmented.

After IKEv2 fragmentation is disabled or enabled or the packet fragment size is changed, the VPN tunnels that are hosted on the IKE gateway are brought down and IKE and IPsec SAs are renegotiated.

Caveats

The following features are not supported with IKEv2 fragmentation:

- Path MTU Discovery.
- SNMP.

SEE ALSO

Certificate Authority Profiles

IKE Policy with Local Certificate

SUMMARY

Read this topic to learn about IKE policy that uses local certificate for authentication during IKE negotiations.

IN THIS SECTION

- Local Certificate in IKE | 69
- Local Certificate Selection Criteria | 69
- Multiple Local Certificates | 70
- Role of IKE ID in Certificate Selection | 70

- Local Certificate Caching in IKE | 70
- Addition or Deletion of Local Certificates | 71
- Configure IKE Policy with Multiple Local Certificates | 71

Local Certificate in IKE

Internet Key Exchange (IKE) supports certificate-based authentication to securely establish trust between VPN peers. In this authentication method, both peers use digital certificates to verify each other's identities during IKE negotiation.

During the IKE authentication phase, the remote peer (initiator) sends a certificate request payload to the local device (responder). In response, the local device sends a certificate payload containing its local certificate. The initiator uses the received certificate to validate the responder's identity by verifying the certificate chain and checking the trust relationship with the issuing certificate authority (CA). This validation ensures that the responder is authorized to participate in the IKE negotiation.

Local Certificate Selection Criteria

The local certificate is the certificate that the responder presents to the initiator during IKE negotiation to prove its identity. A VPN peer includes CA identifiers in a certificate request to indicate the CAs it trusts. The iked process takes into account only the first eight identifiers and ignores any additional identifiers. The local device examines CA identifiers included in the peer's certificate request payload and selects the local certificate based on the criteria listed in the following table.

Table 5: Local Certificate Selection Criteria for IKE

IKE Version	Certificate Selection Criteria
IKEv2	Selects the local certificate based on the CA hash provided in the certificate request payload.
IKEv1	Selects the local certificate based on the CA subject distinguished name (DN) provided in the certificate request payload.
IKE fallback behavior	Selects the first certificate loaded in the local certificate list when no match is found.

Multiple Local Certificates

The device selects and manages local certificates based on whether it supports a single or multiple local certificates.

When the device supports only a single local certificate:

- Each new local certificate configuration replaces the existing certificate.
- The gateway uses a single certificate, even if it serves multiple remote peers. Some peers might not have the CA certificate required to validate the gateway certificate because the gateway might not present a matching local certificate in response to the peer's request.

When the device supports multiple local certificates with the iked process:

- You can configure multiple local certificates in the IKE policy.
- Each configured local certificate is appended to the existing list, rather than replacing it. Certificates remain available until they expire or they're explicitly removed.
- The device evaluates local certificates in their listed order. The device selects the first certificate whose chain—built to a trusted root CA—matches one of the peer CA identifiers. If no configured certificate matches the CA identifiers in the certificate request, the iked process selects a local certificate whose chain can be built to a trusted root CA. The process evaluates certificates sequentially and selects the first certificate with a valid chain to send to the peer.

Role of IKE ID in Certificate Selection

- The local IKE ID does not influence IKE certificate selection, except when the IKE ID type is a DN at the [`edit security ike gateway` *`name`* `remote-identity distinguished-name`] hierarchy level:
  - When the IKE ID is a DN, IKE performs a PKI lookup to retrieve the certificate that matches the DN without evaluating the listed order.
  - After retrieval, IKE caches the certificate for future use.

If the local IKE ID doesn't match the selected local certificate, then IKE handshake might fail on the peer device.

Local Certificate Caching in IKE

After a local certificate is selected, IKE caches the certificate to minimize future PKI lookups. Once cached in IKE:

- The iked process uses the cached certificate for subsequent connections, as long as it matches the incoming CA identifiers.

- If the cached certificate doesn't match the incoming CA identifiers, the iked process selects a new certificate and rebuilds the certificate chain from the configured local certificate list.
- The certificate order does not affect certificate selection.
- Reordering certificates (for example, moving the first certificate to another position) does not trigger a new PKI lookup or delete the existing IPsec connection.
- IKE continues to use the cached certificate as long as it meets the required selection criteria.
- If the local certificate is deleted from the device, the corresponding cached certificate is also cleared.
- Restarting the iked process clears the certificate cache.

Addition or Deletion of Local Certificates

- Addition of certificates does not delete the existing security associations (SAs).
- If a certificate is deleted from the certificate list, all IKE SAs that use the corresponding IKE policy are deleted.
- Deletion of a local certificate or a CA certificate from PKI does not cause commit error in IKE or delete the existing IPsec connection.

Configure IKE Policy with Multiple Local Certificates

To configure multiple local certificates, perform the following steps:

1. Configure a CA profile and load the CA certificate. See Certificate Authority.
2. Load an existing local certificate on the device or enroll a local certificate from the CA whose CA certificate was previously loaded. See Digital Certificates.

   Repeat the step for each local certificate in the IKE policy.
3. Associate the local certificates with the IKE policy. Use either of the following commands:

```
set security ike policy IKE-POL certificate local-certificate [cert1 cert2 cert3]
```

or

```
[edit]
set security ike policy IKE-POL certificate local-certificate cert1
```

```
set security ike policy IKE-POL certificate local-certificate cert2
set security ike policy IKE-POL certificate local-certificate cert3
```

Configure IKE policy, IKE proposal, and IKE gateway followed by IPsec policy and proposal for end-to-end IPsec VPN tunnel establishment.

To determine which local certificate is selected from the certificate list configured in an IKE policy, review the IKE trace logs. The logs display the certificate subject DN.

SEE ALSO

certificate

IKE Policy with a Trusted CA

This example shows how to bind a trusted CA server to an IKE policy of the peer.

Before you begin, you must have a list of all the trusted CAs you want to associate with the IKE policy of the peer.

You can associate an IKE policy to a single trusted CA profile or a trusted CA group. For establishing a secure connection, the IKE gateway uses the IKE policy to limit itself to the configured group of CAs (ca-profiles) while validating the certificate. A certificate issued by any source other than the trusted CA or trusted CA group is not validated. If there is a certificate validation request coming from an IKE policy then the associated CA profile of the IKE policy will validate the certificate. If an IKE policy is not associated with any CA then by default the certificate is validated by any one of the configured CA profiles.

In this example, a CA profile named `root-ca` is created and a `root-ca-identity` is associated to the profile.

You can configure a maximum of 20 CA profiles that you want to add to a trusted CA group. You cannot commit your configuration if you configure more than 20 CA profiles in a trusted CA group.

1. Create a CA profile and associate a CA identifier to the profile.

```
[edit]
user@host# set security pki ca-profile root-ca ca-identity root-ca
```

2. Define an IKE proposal and the IKE proposal authentication method.

```
[edit]
user@host# set security ike proposal ike_prop authentication-method rsa-signatures
```

3. Define the Diffie-Hellman group, authentication algorithm, an encryption algorithm for the IKE proposal.

```
[edit]
user@host# set security ike proposal ike_prop dh-group group2
user@host# set security ike proposal ike_prop authentication-algorithm sha-256
user@host# set security ike proposal ike_prop encryption-algorithm aes-256-cbc
```

4. Configure an IKE policy and associate the policy with the IKE proposal.

```
[edit]
user@host# set security ike policy ike_policy proposals ike_prop
```

5. Configure a local certificate identifier for the IKE policy.

```
[edit]
user@host# set security ike policy ike_policy certificate local-certificate SPOKE
```

6. Define the CA to be used for the IKE policy.

```
[edit]
user@host# set security ike policy ike_policy certificate trusted-ca ca-profile root-ca
```

To view the CA profiles and the trusted CA groups configured on your device, run `show security pki` command.

```
user@host# show security ike
    proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy ike_policy {
```

```
    proposals ike_prop;
    certificate {
        local-certificate SPOKE;
        trusted-ca ca-profile root-ca;
    }
}
```

The `show security ike` command displays the CA profile group under the IKE policy named `ike_policy` and the certificate associated with the IKE policy.

SEE ALSO

Basic Elements of PKI in Junos OS

IKE Policy with Trusted CA Group

SUMMARY

Read this topic to learn how to configure an IKE policy to use a trusted CA group for certificate-based authentication.

IN THIS SECTION

- Overview of Trusted CA Groups in IKE | 74
- Configure IKE Policy with Trusted CA Group | 75

Overview of Trusted CA Groups in IKE

A trusted Certificate Authority (CA) group bundles one or more CA profiles and acts as a single trust anchor for validating peer certificate during IKE negotiations. The device can contain multiple CA certificates. You can configure IKE to use a trusted CA group, which is a subset of the configured CA certificates. During certificate-based authentication for IKEv2 or IKEv1, the device verifies that the peer's certificate chain till the root. When an IKE policy uses a trusted CA group, the device validates the peer's certificate against all CA profiles included in the group.

Trusted CA groups simplify certificate management in environments that use multiple CAs or when you migrate from one CA to another. You can update the CA group at any time to add or remove CA profiles.

When a VPN peer initiates a connection, the device validates the peer's certificate using the trusted CA group configured in the IKE policy:

- If the certificate is signed by a CA in the trusted group, authentication succeeds and the connection is allowed.
- If the certificate is not signed by a CA in the trusted group, authentication fails and the IPsec tunnel is not established.

A trusted CA group therefore ensures that the device accepts certificates issued by authorized CAs only, simplifying certificate management while maintaining secure VPN authentication.

Configure IKE Policy with Trusted CA Group

Before you configure an IKE policy with a trusted CA group, ensure that:

- You have a list of all CAs that you want to trust.
- The IKE policy uses certificate-based authentication.
- You have access to configure PKI and IKE settings on the device.

To configure and verify the IKE policy with the trusted CA group:

1. Create CA profiles and load certificates. Each CA profile represents one CA. See Configure CA Profiles.

```
[edit]
user@host# set security pki ca-profile CORP-CA ca-identity CORP-CA
user@host# set security pki ca-profile BRANCH1-CA ca-identity BRANCH1-CA
user@host# set security pki ca-profile BRANCH2-CA ca-identity BRANCH2-CA
```

Each CA profile represents a trusted CA that can validate the peer certificates.

```
user@host> request security pki ca-certificate load ca-profile CORP-CA filename corp-ca.crt
user@host> request security pki ca-certificate load ca-profile BRANCH1-CA filename branch1-
ca.crt
user@host> request security pki ca-certificate load ca-profile BRANCH2-CA filename branch2-
ca.crt
```

Load the CA certificates into their corresponding CA profiles. Each loaded CA certificate enables the device to validate peer certificates issued by that CA.

2. Create a trusted CA group. Group the CA profiles into a trusted CA group. The order of CA profiles in the trusted CA group does not affect certificate validation. See Configure a Trusted CA Group.

```
[edit]
user@host# set security pki trusted-ca-group VPN-TRUSTED-CA-GROUP ca-profiles [CORP-CA
BRANCH1-CA BRANCH2-CA]
```

   The device uses this group to validate peer certificates during IKE authentication.

3. Configure the IKE policy to use certificate-based authentication and associate the trusted CA group with the policy.

```
[edit]
user@host# set security ike policy IKE-POL mode main
user@host# set security ike policy IKE-POL certificate trusted-ca trusted-ca-group VPN-
TRUSTED-CA-GROUP
```

4. Verify that the trusted CA group is configured correctly.

```
[edit]
user@host# show security pki
ca-profile-group VPN-TRUSTED-CA-GROUP {
    ca-profile CORP-CA;
    ca-profile BRANCH1-CA;
    ca-profile BRANCH2-CA;
}
```

```
[edit]
user@host# show security ike
policy IKE-POL {
    certificate {
        trusted-ca-profile-group VPN-TRUSTED-CA-GROUP;
    }
}
```

   Verify that the trusted CA group is associated with the IKE policy.

After the configuration is complete:

- The device authenticates VPN peers using certificates.

- The device trusts certificates issued by any CA in the trusted CA group.
- Incoming VPN connections succeed if the peer certificate is validated by the CA group.

To establish an end-to-end IPsec VPN tunnel, complete the remaining configuration—configure IKE policy, IKE proposal, and IKE gateway followed by IPsec policy and IPsec proposal.

Configuring Establish-Tunnel Responder-only in IKE

This topic shows how to configure establish-tunnels responder-only in Internet Key Exchange (IKE). Initiate the tunnels from the remote peer and send the traffic through all the tunnels. Specifies when IKE is activated.

On firewalls, the `establish-tunnels` option supports the `responder-only` and `responder-only-no-rekey` values under the `[edit security ipsec vpn vpn-name]` hierarchy-level.

These options are supported only on a site-to-site VPN. These option are not supported on Auto VPN.

The `responder-only` and `responder-only-no-rekey` options does not establish any VPN tunnel from the device, so the VPN tunnel is initiated from the remote peer. When you configure `responder-only`, an established tunnel rekeys both IKE and IPsec based on the configured IKE and IPsec lifetime values. When you configure `responder-only-no-rekey`, an established tunnel does not rekey from the device and relies on the remote peer to initiate rekey. If the remote peer does not initiate rekey, then the tunnel teardown occurs after hard-lifetime expires.

Before you begin:

- Understand how to establish an AutoKey IKE IPsec tunnel. Read "IPsec Overview" on page 12.

To configure establish-tunnel responder-only in IKE:

1. Configure establish-tunnel responder-only

```
user@host# set security ipsec vpn S2S_VPN establish-tunnel responder-only
```

2. Confirm your configuration by entering the `show security ipsec vpn IPSEC_VPN` command.

```
user@host# show security ipsec vpn IPSEC_VPN
bind-interface st0.1;
ike {
        gateway IKE_GW;
        ipsec-policy IPSEC_POL;
```

```
    }
establish-tunnels responder-only;
```

3. Configure establish-tunnel responder-only-no-rekey

```
user@host# set security ipsec vpn S2S_VPN establish-tunnel responder-only-no-rekey
```

4. Confirm your configuration by entering the `show security ipsec vpn IPSEC_VPN` command.

```
user@host# show security ipsec vpn IPSEC_VPN
bind-interface st0.1;
ike {
        gateway IKE_GW;
        ipsec-policy IPSEC_POL;
    }
establish-tunnels responder-only-no-rekey;
```

In case of multiple VPN objects, the Responder-only mode will take precedence. If any of the VPN in a gateway is configured with responder-only mode, all VPN's in the gateway must be configured with the responder-only mode.

Platform-Specific IKEv2 Responder Only Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platforms.

Table 6: Platform-Specific Behavior

Platform	Difference

SRX Series	• On SRX5000 line that support IKEv2 `responder-only` and `responder-only-no-rekey` values in `establish-tunnels` option:
	  ◦ The firewall supports the options with an SPC3 card only.
	  ◦ The firewall supports the options if the `junos-ike` package is installed for IPsec VPN service with the iked process.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.4R1	Support for signature authentication in IKEv2 protocol is introduced in Junos OS Release 24.4R1.
23.4R1	Support for IKE protection from DDoS attacks is introduced in Junos OS Release 23.4R1.
20.3R1	Support for IKEv2 configuration payload improvements on SRX5000 line and vSRX Virtual Firewall running the iked process is added in Junos OS Release 20.3R1.
20.1R1	Support for IKEv2 configuration payload feature with point-to-point interfaces on SRX5000 line and vSRX Virtual Firewall running the iked process is introduced in Junos OS Release 20.1R1.
20.1R1	Support for a common password configuration in IKEv2 configuration payload requests for an IKE gateway is introduced in Junos OS Release 20.1R1.
19.1R1	Support for `responder-only` and `responder-only-no-rekey` values in the `establish-tunnels` option at the `[edit security ipsec vpn vpn-name]` hierarchy-level is introduced on SRX5000 line in Junos OS Release 19.1R1.
18.1R1	Starting with Junos OS Release 18.1R1, validation of a configured IKE peer can be done with a specified CA server or group of CA servers.

RELATED DOCUMENTATION

- Certificate Authority

IPsec VPN Overview

SUMMARY

Read this topic to know about the IKE and IPsec packet processing, and IPsec VPN topologies on SRX Series Firewalls. Learn about the services processing

IN THIS SECTION

- IPsec VPN Topologies on SRX Series Firewalls | 81

cards, cryptographic acceleration, routing protocols support, and the iked process support.

- Understanding IKE and IPsec Packet Processing | **81**
- Distribution of IKE and IPsec Sessions Across SPUs | **84**
- VPN Support for Inserting Services Processing Cards | **86**
- IPsec VPN with iked Process | **87**
- Cryptographic Acceleration Support | **89**
- Routing Protocols Support on IPsec VPN Tunnels | **90**
- Anti-Replay Window | **90**
- Understanding Hub-and-Spoke VPNs | **91**
- Platform-Specific IPsec VPN Behavior | **92**
- Additional Platform Information | **94**

A VPN is a private network that uses a public network to connect two or more remote sites. Instead of using dedicated connections between networks, VPNs use virtual connections routed (tunneled) through public networks. IPsec VPN is a protocol, consists of set of standards used to establish a VPN connection.

A VPN provides a means by which remote computers communicate securely across a public WAN such as the Internet.

A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication while passing through the WAN, the two participants create an IPsec tunnel.

The term *tunnel* does not denote tunnel mode (see "Packet Processing in Tunnel Mode" on page 81). Instead, it refers to the IPsec connection.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific IPsec VPN Behavior" on page 92 section for notes related to your platform.

See the "Additional Platform Information" on page 94 section for more information.

IPsec VPN Topologies on SRX Series Firewalls

The following are some of the IPsec VPN topologies that Junos operating system (OS) supports:

- Site-to-site VPNs—Connects two sites in an organization together and allows secure communications between the sites.
- Hub-and-spoke VPNs—Connects branch offices to the corporate office in an enterprise network. You can also use this topology to connect spokes together by sending traffic through the hub.
- Remote access VPNs—Allows users working at home or traveling to connect to the corporate office and its resources. This topology is sometimes referred to as an *end-to-site tunnel*.

Understanding IKE and IPsec Packet Processing

IN THIS SECTION

- Packet Processing in Tunnel Mode | **81**

An IPsec VPN tunnel consists of tunnel setup and applied security. During tunnel setup, the peers establish security associations (SAs), which define the parameters for securing traffic between themselves. See IPsec Overview. After the tunnel is established, IPsec protects the traffic sent between the two tunnel endpoints by applying the security parameters defined by the SAs during tunnel setup. Within the Junos OS implementation, IPsec is applied in tunnel mode, which supports the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols.

This topic includes the following sections:

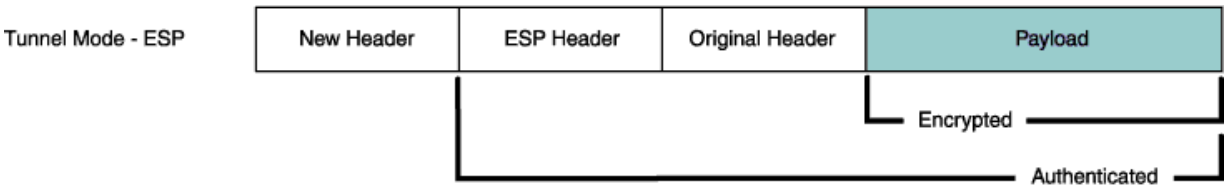
Packet Processing in Tunnel Mode

IPsec operates in one of two modes—transport or tunnel. When both ends of the tunnel are hosts, you can use either mode. When at least one of the endpoints of a tunnel is a security gateway, such as a Junos OS router or firewall, you must use tunnel mode. Juniper Networks devices always operate in tunnel mode for IPsec tunnels.

In tunnel mode, the entire original IP packet—payload and header—is encapsulated within another IP payload, and a new header is appended to it, as shown in Figure 10 on page 82. The entire original packet can be encrypted, authenticated, or both. With the Authentication Header (AH) protocol, the AH

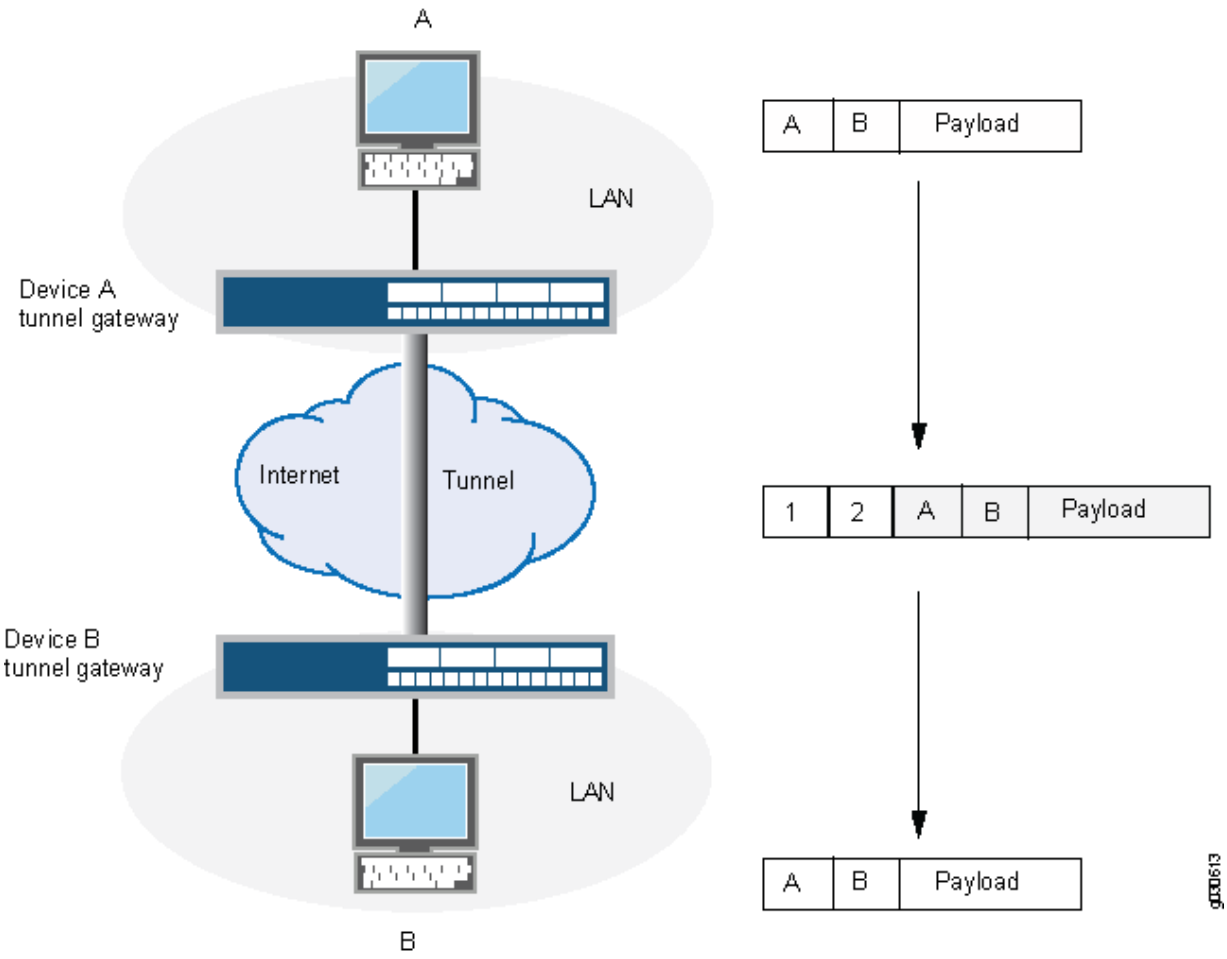
and new headers are also authenticated. With the Encapsulating Security Payload (ESP) protocol, the ESP header can also be authenticated.

Figure 10: Tunnel Mode



In a site-to-site VPN, the source and destination addresses used in the new header are the IP addresses of the outgoing interface. See Figure 11 on page 83.

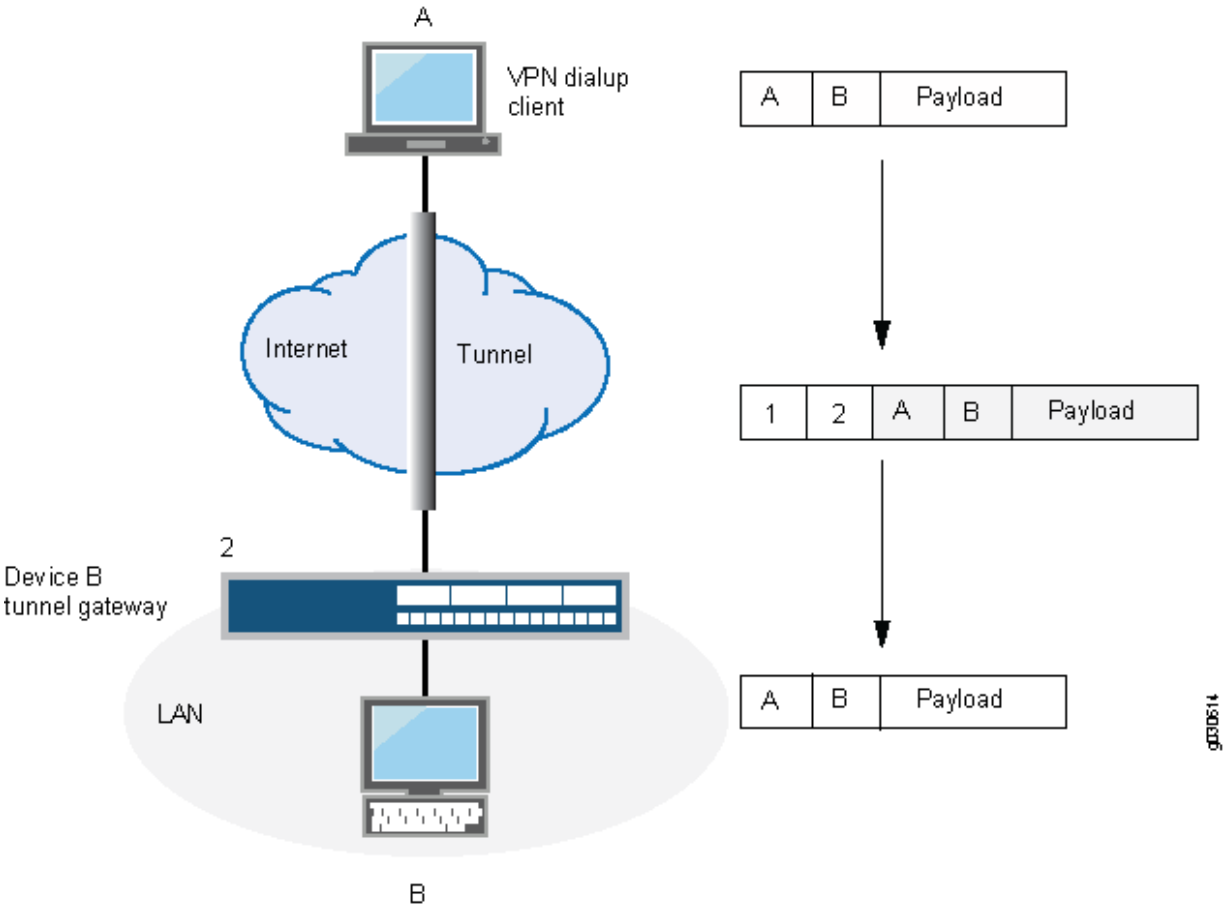
Figure 11: Site-to-Site VPN in Tunnel Mode



In a dial-up VPN, there is no tunnel gateway on the VPN dial-up client end of the tunnel; the tunnel extends directly to the client itself (see Figure 12 on page 84). In this case, on packets sent from the dial-up client, both the new header and the encapsulated original header have the same IP address: that of the client's computer.

Some VPN clients, such as the Netscreen-Remote, use a virtual inner IP address (also called a "sticky address"). Netscreen-Remote enables you to define the virtual IP address. In such cases, the virtual inner IP address is the source IP address in the original packet header of traffic originating from the client, and the IP address that the ISP dynamically assigns the dial-up client is the source IP address in the outer header.

Figure 12: Dial-Up VPN in Tunnel Mode



SEE ALSO

Route-Based IPsec VPNs | 320

Distribution of IKE and IPsec Sessions Across SPUs

Review the "Platform-Specific SPUs VPN Processing Behavior" on page 93 section for notes related to your platform.

In SRX Series Firewalls, IKE provides tunnel management for IPsec and authenticates end entities. IKE performs a Diffie-Hellman (DH) key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the IP layer.

The VPN is created by distributing the IKE and IPsec workload among the multiple Services Processing Units (SPUs) of the platform. For site-to-site tunnels, the least-loaded SPU is chosen as the anchor SPU. If multiple SPUs have the same smallest load, any of them can be chosen as an anchor SPU. Here, load corresponds to the number of site-to-site gateways or manual VPN tunnels anchored on an SPU. For dynamic tunnels, the newly established dynamic tunnels employ a round-robin algorithm to select the SPU.

In IPsec, the workload is distributed by the same algorithm that distributes the IKE. The Phase 2 SA for a given VPN tunnel termination points pair is exclusively owned by a particular SPU, and all IPsec packets belonging to this Phase 2 SA are forwarded to the anchoring SPU of that SA for IPsec processing.

Multiple IPsec sessions (Phase 2 SA) can operate over one or more IKE sessions. The SPU that is selected for anchoring the IPsec session is based on the SPU that is anchoring the underlying IKE session. Therefore, all IPsec sessions that run over a single IKE gateway are serviced by the same SPU and are not load-balanced across several SPUs.

Table 7 on page 85 shows an example of the firewall with three SPUs running seven IPsec tunnels over three IKE gateways.

Table 7: Distribution of IKE and IPsec Sessions Across SPUs

SPU	IKE Gateway	IPsec Tunnel
SPU0	IKE-1	IPsec-1
		IPsec-2
		IPsec-3
SPU1	IKE-2	IPsec-4
		IPsec-5
		IPsec-6
SPU2	IKE-3	IPsec-7

The three SPUs have an equal load of one IKE gateway each. If a new IKE gateway is created, SPU0, SPU1, or SPU2 could be selected to anchor the IKE gateway and its IPsec sessions.

Setting up and tearing down existing IPsec tunnels does not affect the underlying IKE session or existing IPsec tunnels.

Use the following `show` command to view the current tunnel count per SPU: `show security ike tunnel-map`.

Use the `summary` option of the command to view the anchor points of each gateway: `show security ike tunnel-map summary`.

VPN Support for Inserting Services Processing Cards

High-end SRX Series Firewalls have a chassis-based distributed processor architecture. The flow processing power is shared and is based on the number of Services Processing Cards (SPCs). You can scale the processing power of the device by installing new SPCs.

Review the "Platform-Specific SRX5000 Line SPC Behavior" on page 93 section for notes related to your platform.

See the "Additional Platform Information for kmd and iked Process in SRX5000 Line" on page 94 section for more information.

In high-end SRX Series chassis cluster, you can insert SPCs on the devices without affecting or disrupting the traffic on the existing IKE or IPsec VPN tunnels.

You can insert SPC3 or SPC2 card to an existing chassis containing SPC3 card. You can only insert the cards in a higher slot than the existing SPC3 card on the chassis.

However, existing tunnels cannot use the processing power of the Service Processing Units (SPUs) in the new SPCs. A new SPU can anchor newly established site-to-site and dynamic tunnels. Newly configured tunnels are not, however, guaranteed to be anchored on a new SPU.

Site-to-site tunnels are anchored on different SPUs based on a load-balancing algorithm. The load-balancing algorithm is dependent on number flow threads each SPU is using. Tunnels belonging to the same local and remote gateway IP addresses are anchored on the same SPU on different flow RT threads used by the SPU. The SPU with the smallest load is chosen as the anchor SPU. Each SPU maintains number of flow RT threads that are hosted in that particular SPU. The number of flow RT threads hosted on each SPU vary based on the type of SPU.

Tunnel load factor = Number of tunnels anchored on the SPU / Total number of flow RT threads used by the SPU.

Dynamic tunnels are anchored on different SPUs based on a round-robin algorithm. Newly configured dynamic tunnels are not guaranteed to be anchored on the new SPC.

When both SPC2 and SPC3 cards are installed, you can verify the tunnel mapping on different SPUs using the `show security ipsec tunnel-distribution` command.

Use the command `show security ike tunnel-map` to view the tunnel mapping on different SPUs with only SPC2 card inserted. The command `show security ike tunnel-map` is not valid in an environment where SPC2 and SPC3 cards are installed.

Inserting SPC3 Card: Guidelines and Limitations:

- In a chassis cluster, if one of the nodes has 1 SPC3 card and the other node has 2 SPC3 cards, the failover to the node that has 1 SPC3 card is not supported.
- You must insert the SPC3 or SPC2 in an existing chassis in a higher slot than a current SPC3 present in a lower slot.
- For SPC3 ISHU to work, you must insert the new SPC3 card into the higher slot number.
- We do not support SPC3 hot removal.

SEE ALSO

> *show security ike tunnel-map*

IPsec VPN with iked Process

IN THIS SECTION

- IPsec VPN Features Not Supported with iked Process | **88**

The two processes, iked and ikemd support IPsec VPN features on SRX Series Firewalls. A single instance of iked and ikemd run on the Routing Engine at a time.

With `junos-ike` package, the firewall runs IPsec VPN service using the iked process. SRX Series Firewalls support `junos-ike` package is multiple releases.

See the "Additional Platform Information for `junos-ike` Package Support" on page 94 section for more information.

Both the iked and ikemd processes running on the Routing Engine are available with the `junos-ike` package.

To install the `junos-ike` package on SRX Series Firewall, use the following command:

```
user@host> request system software add optional://junos-ike.tgz
Verified junos-ike signed by PackageProductionECP256_2022 method ECDSA256+SHA256
Rebuilding schema and Activating configuration...
mgd: commit complete
Restarting MGD ...

WARNING: cli has been replaced by an updated version:
CLI release 20220208.163814_builder.r1239105 built by builder on 2022-02-08 17:07:55 UTC
Restart cli using the new version ? [yes,no] (yes)
```

To restart the ikemd process in the Routine Engine use the `restart ike-config-management` command.

To restart the iked process in the Routing Engine use the `restart ike-key-management` command.

NOTE: Disregarding the specified Junos OS release versions when installing the `junos-ike` package may result in unsupported features not functioning as expected.

To operate IPsec VPN features using the legacy kmd process on SRX Series Firewalls, run the `request system software delete junos-ike` command and reboot the device.

To check the installed `junos-ike` package, use the following command:

```
user@host> show version | grep ike

JUNOS ike [20190617.180318_builder_junos_182_x41]
JUNOS ike [20190617.180318_builder_junos_182_x41]

{primary:node0}
```

IPsec VPN Features Not Supported with iked Process

This section provides a summary of IPsec VPN features that are not supported on the SRX Series Firewalls.

Table 8 on page 89 summarizes the non-supported IPsec VPN features on SRX Series Firewalls and vSRX Virtual Firewall running iked process.

Table 8: IPsec VPN Features Not Supported on SRX Series Firewalls

Features	Support Availability
AutoVPN Protocol Independent Multicast (PIM) point-to-multipoint mode.	No. But support is available on vSRX 3.0
Configuring forwarding class on IPsec VPNs.	No
Group VPN.	No
Packet size configuration for IPsec datapath verification.	No
Policy-based IPsec VPN.	No

Cryptographic Acceleration Support

Review the "Platform-Specific Cryptographic Acceleration Behavior" on page 92 section for notes related to your platform.

See the "Additional Platform Information for Cryptographic Acceleration Support" on page 94 section for more information.

Junos OS supports acceleration of cryptographic operations to the hardware cryptographic engine. SRX Series Firewall can offload DH, RSA, and ECDSA cryptographic operations to the hardware cryptographic engine.

With junos-ike package, the firewall runs IPsec VPN service using the iked process. The firewall requires the iked process as the control plane software to install and enable advanced IPsec VPN features. As a result of junos-ike package the firewall runs the iked and ikemd process on the routing engine by default instead of IPsec key management daemon (kmd).

The firewalls support hardware acceleration for various ciphers.

SEE ALSO

show security ipsec security-associations

show security ipsec tunnel-distribution

Routing Protocols Support on IPsec VPN Tunnels

See the Table 16 on page 97, Table 17 on page 97, Table 18 on page 98, Table 19 on page 99, Table 20 on page 100, and Table 21 on page 101 in "Additional Platform Information" on page 94 section for more information.

Junos OS supports routing protocols on IPsec VPN tunnels with SRX Series Firewalls and MX Series routers with SPC3. Supported protocols include OSPF, BGP, PIM, RIP, and BFD when running the kmd or iked process. The protocol support varies based on the following:

- IP addressing scheme: IPv4 or IPv6 addresses
- Type of st0 interface: point-to-point (P2P) or point-to-multipoint (P2MP)

Anti-Replay Window

Review the "Platform-Specific Antireplay Window Behavior" on page 94 section for notes related to your platform.

On SRX Series Firewalls, `anti-replay-window` is enabled by default with a window size value of 64.

To configure the window size, use the new `anti-replay-window-size` option. An incoming packet is validated for replay attack based on the `anti-replay-window-size` that is configured.

You can configure `replay-window-size` at two different levels:

- **Global level**—Configured at the [`edit security ipsec`] hierarchy level.

  For example:

  ```
  [edit security ipsec]
   user@host# set anti-replay-window-size <64..8192>;
  ```

- **VPN object**—Configured at the [`edit security ipsec vpn vpn-name ike`] hierarchy level.

For example:

```
[edit security ipsec vpn vpn-name ike]
user@host# set anti-replay-window-size <64..8192>;
```

If anti-replay is configured at both levels, the window size configured for a VPN object level takes precedence over the window size configured at the global level. If anti-replay is not configured, the window size is 64 by default.

To disable the anti-replay window option on a VPN object, use the `set no-anti-replay` command at the [`edit security ipsec vpn vpn-name ike`] hierarchy level. You cannot disable anti-replay at the global level.

You cannot configure both `anti-replay-window-size` and `no-anti-replay` on a VPN object.

SEE ALSO

> *anti-replay-window-size*

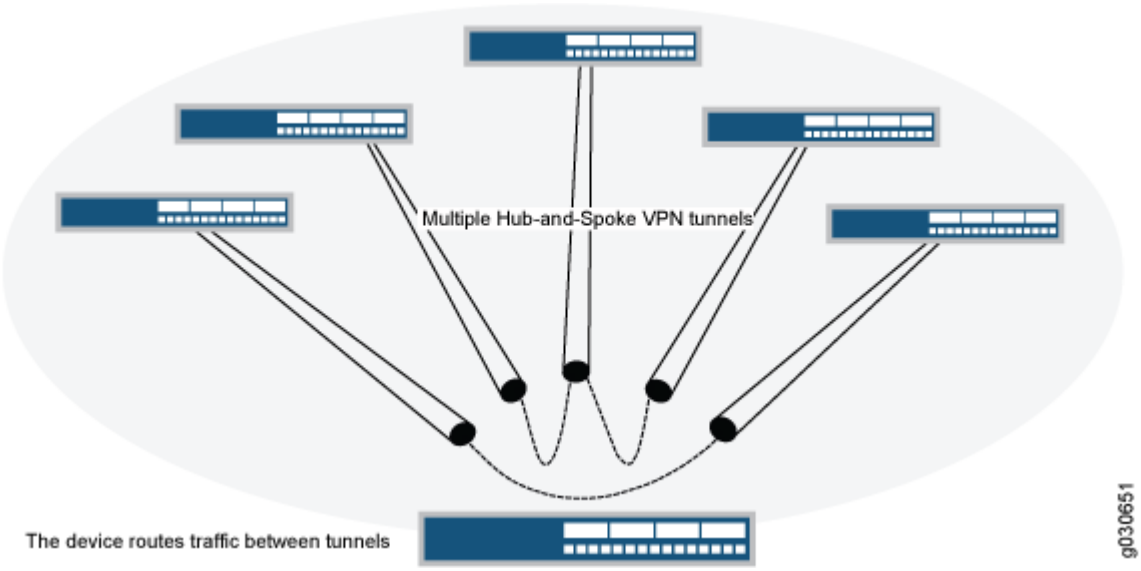
Understanding Hub-and-Spoke VPNs

If you create two VPN tunnels that terminate at a device, you can set up a pair of routes so that the device directs traffic exiting one tunnel to the other tunnel. You also need to create a policy to permit the traffic to pass from one tunnel to the other. Such an arrangement is known as *hub-and-spoke VPN*. (See Figure 13 on page 92.)

You can also configure multiple VPNs and route traffic between any two tunnels.

SRX Series Firewalls support only the route-based hub-and-spoke feature.

Figure 13: Multiple Tunnels in a Hub-and-Spoke VPN Configuration



Platform-Specific IPsec VPN Behavior

IN THIS SECTION

- Platform-Specific SPUs VPN Processing Behavior | 93
- Platform-Specific SRX5000 Line SPC Behavior | 93
- Platform-Specific Cryptographic Acceleration Behavior | 94
- Platform-Specific Antireplay Window Behavior | 94

Use Feature Explorer to confirm platform and release support for specific features.

See the "Additional Platform Information" on page 94 section for more information.

Use the following tables to review platform-specific behaviors for your platforms.

Platform-Specific SPUs VPN Processing Behavior

Table 9: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX5000 line of devices that support SPUs, the firewall distributes the IKE and IPsec sessions across the multiple SPUs of the platform. See "Distribution of IKE and IPsec Sessions Across SPUs" on page 84.

Platform-Specific SRX5000 Line SPC Behavior

Table 10: Platform-Specific Behavior

Platform	Difference

SRX Series	• On SRX5000 line of devices that support SPCs:
	  • When you insert a new SPC in each chassis of the cluster, the firewall ensures uninterrupted traffic flow without affecting the existing tunnels.
	  • Reboot the node after the inserting SPC3 to activate the card. Once the node finishes rebooting, the firewall distributes IPsec tunnels to the cards.
	  • The existing IPsec VPN features supported on SRX5K-SPC3 are also supported when SRX5K-SPC-4-15-320 (SPC2) and SPC3 cards are used in chassis cluster mode or standalone mode. See "VPN Support for Inserting Services Processing Cards" on page 86 for more details.
	  • On SRX5800 chassis cluster, do not insert the SPC3 card in the highest slot (slot 11) due to the power and heat distribution limit.

Platform-Specific Cryptographic Acceleration Behavior

Table 11: Platform-Specific Behavior

Platform	Difference

SRX Series	• On SRX Series Firewalls that support cryptographic acceleration:
	  • The firewall supports most of the ciphers when running IPsec VPN service using the iked process, as listed in Table 15 on page 96.
	  • SRX Series midrange platforms covering SRX1500, SRX4100, SRX4200 and SRX4600 Series Firewalls, offloads the DH, RSA, and ECDSA cryptographic operations to the hardware cryptographic engine on devices that run `junos-ike` package.
	  • On SRX5000 line with SPC3, you must install the `junos-ike` package for the iked process.
	  • On vSRX Virtual Firewall, the data plane CPU thread offloads DH, RSA, and ECDSA operations. vSRX Virtual Firewalls do not provide hardware acceleration. See "Cryptographic Acceleration Support" on page 89.

Platform-Specific Antireplay Window Behavior

Table 12: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX Series Firewalls that support antireplay window configuration, you can set the `anti-replay-window` size between 64 and 8192 (power of 2) for IPsec VPN services with the iked process.

Additional Platform Information

Use Feature Explorer to confirm platform and release support for specific features. Additional Platforms may be supported.

Table 13: Additional Platform Information for kmd or iked Process in SRX5000 Line

Supported Process	SRX5000 Line
iked process	Supports two options: • With only SPC3 card installed • With both SPC2 and SPC3 cards installed
kmd process	• With only SPC2 card installed

Table 14: Additional Platform Information for `junos-ike` Package Support

`junos-ike` Package	SRX1500	SRX1600 SRX2300	SRX400 SRX440 SRX440-2 AC	SRX4100 SRX4200 SRX4600	SRX4300	SRX4700	SRX5000 Line with SPC3	vSRX 3.0
Default	25.2R1 and later	23.4R1 and later	Not applicable, but supports iked process from Junos OS Evolved Release 25.4X300-D10EVO and later	25.2R1 and later	24.2R1 and later	24.4R1 and later	19.4R1 and later (for RE3)	25.2R1 and later
Optional	22.3R1 and later	NA	NA	22.3R1 and later	NA	NA	18.2R1 and later (for RE2)	20.3R1 and later

Table 15: Additional Platform Information for Cryptographic Acceleration Support

Ciphers	SRX1500 (kmd)	SRX1500 (iked)	SRX400 SRX4400 SRX4400-2AC (iked)	SRX4100 SRX4200 (kmd)	SRX4100 SRX4200 (iked)	SRX4600 (kmd)	SRX4600 (iked)	SRX5000 Line with SPC3 (iked)	vSRX 3.0 (kmd)	vSRX 3.0 (iked)
DH Group 14	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DH (Groups 19, 20)	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes
DH (Groups 15, 16)	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes
DH Group 21	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes
DH Group 24	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
RSA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ECDSA (256, 384, 521)	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes

Table 16: Additional Platform Information for OSPF Support on IPsec VPN

OSPF on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+SPC2)	vSRX 3.0	MX-SPC3
st0 in P2P	With IPv4 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	With IPv6 address	No	No	No	No	No	No	No	No
st0 in P2MP	With IPv4 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	With IPv6 address	No	No	No	No	No	No	No	No

Table 17: Additional Platform Information for OSPv3 Support on IPsec VPN

OSPFv3 on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+SPC2)	vSRX 3.0	MX-SPC3
st0 in P2P	With IPv4 address	No	No	No	No	No	No	No	No

Table 17: Additional Platform Information for OSPv3 Support on IPsec VPN *(Continued)*

OSPFv3 on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
	With IPv6 address	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
st0 in P2MP	With IPv4 address	No	No	No	No	No	No	No	No
	With IPv6 address	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes

Table 18: Additional Platform Information for BGP Support on IPsec VPN

BGP on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
st0 in P2P	With IPv4 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	With IPv6 address	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes

Table 18: Additional Platform Information for BGP Support on IPsec VPN *(Continued)*

BGP on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
st0 in P2MP	With IPv4 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	With IPv6 address	Yes	Yes	No	Yes	Yes	Yes	Yes	No

Table 19: Additional Platform Information for PIM Support on IPsec VPN

PIM on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
st0 in P2P	With IPv4 address	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
	With IPv6 address	No	No	No	No	No	No	No	No

Table 19: Additional Platform Information for PIM Support on IPsec VPN *(Continued)*

PIM on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
st0 in P2MP	With IPv4 address	Yes	Yes	No	No	No	No	Yes. Multithread is not supported.	No
	With IPv6 address	No	No	No	No	No	No	No	No

Table 20: Additional Platform Information for RIP Protocol Support on IPsec VPN

RIP Protocol on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
st0 in P2P	With IPv4 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	With IPv6 address	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes

Table 20: Additional Platform Information for RIP Protocol Support on IPsec VPN *(Continued)*

RIP Protocol on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
st0 in P2MP	With IPv4 address	No	No	No	No	No	No	No	No
	With IPv6 address	No	No	No	No	No	No	No	No

Table 21: Additional Platform Information for BFD Support on IPsec VPN

BFD on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
st0 in P2P	With IPv4 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	With IPv6 address	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
st0 in P2MP	With IPv4 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Table 21: Additional Platform Information for BFD Support on IPsec VPN *(Continued)*

BFD on IPsec VPN		SRX300 SRX320 SRX340 SRX345 SRX380	SRX1500	SRX400 SRX440 SRX440-2AC	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3 SRX5000 Line with SPC2	SRX5000 Line with Mixed-Mode (SPC3+ SPC2)	vSRX 3.0	MX-SPC3
	With IPv6 address	Yes	Yes	No	Yes	Yes	Yes	Yes	No

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.2R1	Support for SRX4300 firewalls is added in Junos OS Release 24.2R1. The SRX4300 firewalls offer all the IPsec VPN features with the iked process that SRX4200 offers. Support for Policy-based VPN and Group VPN is not available with these platforms.
23.4R1	Support for Dead Peer Detection (DPD) and Auto Discovery VPN (ADVPN) with **iked** process is added in Junos OS Release 23.4R1.
23.4R1	Support for SRX1600 and SRX2300 firewalls is added in Junos OS Release 23.4R1. The SRX1600 and SRX2300 firewalls offer all the IPsec VPN features with the iked process that SRX1500 and SRX4100 respectively offer. Support for Policy-based VPN and Group VPN is not available with these platforms.
23.2R1	Cryptographic acceleration support for SRX mid-range platforms (SRX1500, SRX4100, SRX4200, SRX4600 Series Firewalls) and vSRX Virtual Firewall is added.
20.1R2	By default, `junos-ike` package is installed in Junos OS Releases 20.1R2, 20.2R2, 20.3R2, 20.4R1, and later for SRX5000 line with RE3. As a result **iked** and **ikemd** process runs on the routing engine by default instead of IPsec key management daemon (kmd).

RELATED DOCUMENTATION

Route-Based IPsec VPNs | 320

Inline IPsec

SUMMARY

Read this topic to learn about inline IPsec in firewalls.

IN THIS SECTION

- Overview | 103
- Benefits | 104
- How Inline IPsec Works | 104
- Feature Support for Inline IPsec | 106

Overview

Inline IPsec is a Junos OS feature that offloads IPsec traffic processing from the CPU to the Packet Forwarding Engine of a Juniper Networks device. The feature securely encrypts and decrypts IPsec traffic inside the Packet Forwarding Engine ASIC. With this feature, the CPU manages only PowerMode IPsec (PMI) or IPsec VPN that uses Quick Assist Technology (QAT).

Before the introduction of inline IPsec, firewalls performed IPsec operations in the CPU, benefiting from hardware acceleration such as Intel QAT or software optimization like PMI. With inline IPsec, the firewalls can use the Packet Forwarding Engine ASIC to improve IPsec performance. Offloading IPsec encryption and decryption to the ASIC contributes to higher throughput. Inline IPsec frees up the firewall CPU for other tasks such as:

- Internet Key Exchange version 2 (IKEv2) negotiations for preshared keys (PSKs)
- Post-quantum preshared key (PPK)
- Public Key Infrastructure (PKI) certificate key negotiations
- Various encryption and hash functions used in the inline IPsec

You need a valid license to use the inline IPsec feature on your firewall. When you apply the license, the firewall activates the feature by default.

Benefits

- VPN performance: Improves tunnel performance by offloading encryption and decryption to the ASIC, achieving higher throughputs.
- CPU optimization: Frees up CPU resources by managing tunnels within the ASIC, allowing the CPU to manage other critical tasks and improving overall firewall performance.
- VPN security: Enhances security by using Advanced Encryption Standard–Galois/Counter Mode (AES-GCM) encryption algorithms for hardware offloaded tunnels, ensuring strong encryption standards without compromising performance.
- Deployment: Supports a large number of IPsec tunnels per chassis, making it suitable for large-scale deployments.
- Agility: Supports the ability to manage few tunnels outside the Packet Forwarding Engine.

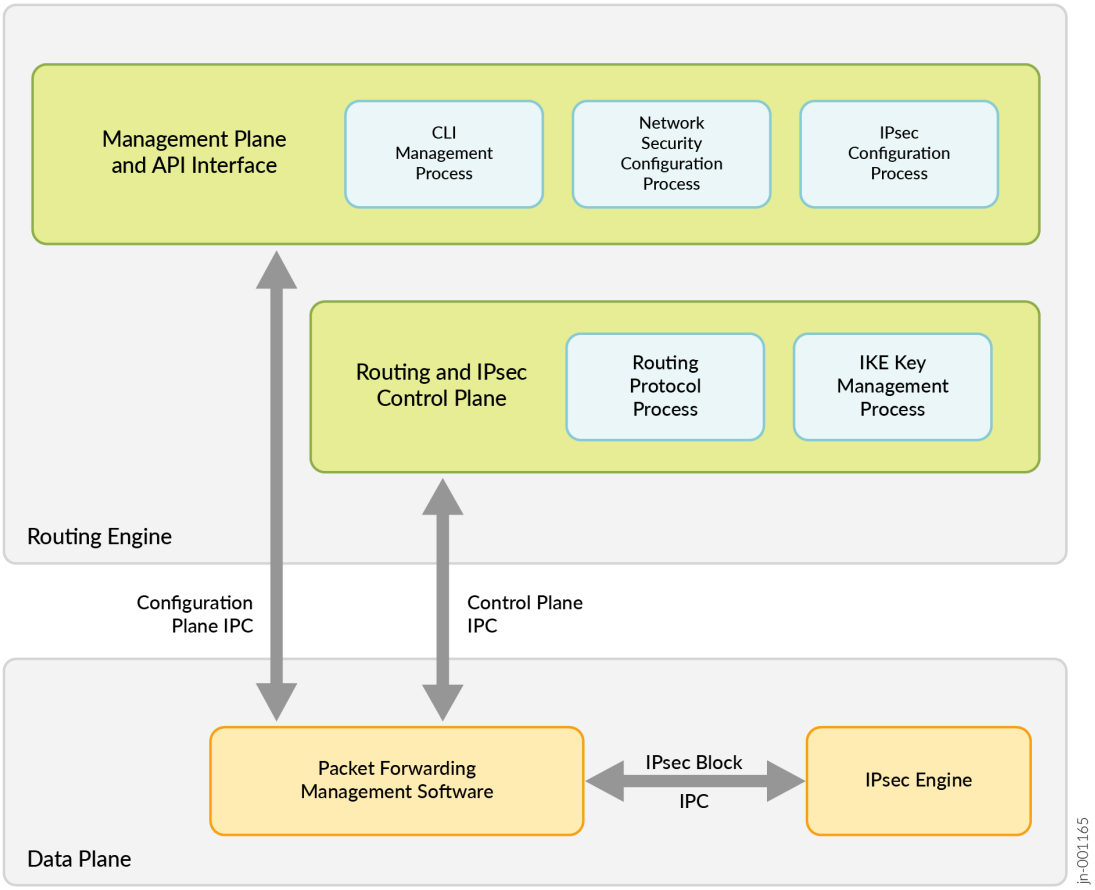
How Inline IPsec Works

Figure 14 on page 105 illustrates the architecture of inline IPsec data plane and its interface with the control and management planes. The inline IPsec architecture includes an IPsec engine that handles IPsec operations. As the Packet Forwarding Engine encrypts or decrypts IPsec traffic without relying on CPU cycles, inline IPsec significantly enhances throughput and optimizes firewall performance.

Understand the following behavior of inline IPsec in firewalls:

- Either the IPsec initiator, the responder, or both can use the feature.
- The firewall performs IPsec Encapsulating Security Payload (ESP) packet encryption and decryption on the built-in ASIC.
- The IKE protocol used for authentication and key exchanges runs in the CPU.
- After the IKE negotiations, the iked process manages the tunnel distribution to the Packet Forwarding Engine.
- The firewall supports:
  - Inline IPsec protocols based on both IPv4 and IPv6 addresses.
  - Inline IPsec for both the IKEv1 and IKEv2 protocols.

Figure 14: Inline IPsec Architecture



You'll notice the following actions in your firewall with inline IPsec:

- The iked process marks the inline IPsec tunnel for hardware offloading in the session flow. This action prevents the tunnel from using the CPU for IPsec encryption and decryption tasks.
- The output of the command `show security ipsec security association` displays:
  - `Hardware Offloaded: Yes` for an inline IPsec VPN tunnel.
  - `Hardware Offloaded: No` when the CPU processes the tunnel.
- The flowd process shows the incoming and outgoing packets for IPsec VPN traffic flows.

To globally disable the inline IPsec hardware offloading of IPsec tunnel processing in the Packet Forwarding Engine ASIC:

- Use the command `set security ipsec hw-offload-disable`.

When you configure this statement, the firewall handles all the IPsec tunnels in CPU instead of the Packet Forwarding Engine ASIC. See ipsec (Security).

Feature Support for Inline IPsec

Firewalls must meet certain criteria to establish IPsec tunnels using inline IPsec feature. Table 22 on page 106 provides the feature support information for inline IPsec, outlining the criteria that determine the tunnel eligibility for the feature. With inline IPsec the CPU manages the features not supported by ASIC, optimizing the firewall performance. The CPU also supports every feature that inline IPsec supports.

Table 22: Feature Support for Inline IPsec

Features Supported by ASIC	Features Supported by CPU
Packet Forwarding Engine on YT ASIC processes IPsec VPN	CPU processes all the IPsec VPN features that uses PMI and QAT. The features include both the Authentication Header (AH) and ESP-based tunnels with AES-GCM and other supported encryption protocols in PMI. See PowerMode IPsec. In addition to managing the PMI and QAT features, the CPU also handles all other features that the Packet Forwarding Engine ASIC does not support. See the following entries in this column for more details.
Tunnel mode using ESP	Tunnel mode using AH
Encryption algorithms AES-GCM with 128-bit key or 256-bit key	All encryption algorithms other than AES-GCM with 128-bit key or 256-bit key
Deployment mode supporting site-to-site IPsec VPN	Deployment mode supporting point-to-multipoint (P2MP) VPNs, group VPN, and Auto Discovery VPN (ADVPN). Interchassis link (ICL) encryption in Multinode High Availability (MNHA).
Antireplay window size up to 4096	Antireplay window size greater than 4096

Table 22: Feature Support for Inline IPsec *(Continued)*

Features Supported by ASIC	Features Supported by CPU
Up to 4000 security associations (SA) or 2000 tunnels	More than 4000 or multiple SAs, also known as child SAs
Network Address Translation-Traversal (NAT-T) Tunnel	Tunnel with `lifesize` option configuration

Note the following limitations with inline IPsec:

- When you set the Don't Fragment (DF) bit for active offloaded tunnels, the configuration takes effect after a rekey.
- When you configure the Differentiated Services Code Point (DSCP) copy for active offloaded tunnels, the configuration takes effect after a rekey.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.2R1	Support for inline IPsec added in Junos OS Release 25.2R1.

RELATED DOCUMENTATION

show security ipsec security-associations

3

CHAPTER

VPN Configuration Overview

IN THIS CHAPTER

- IPsec VPN Configuration Overview | **109**
- Chassis Cluster HA Control Link Encryption | **128**
- Configure IPsec VPN with OCSP for Certificate Revocation Status | **131**
- IPv6 IPsec VPNs | **153**
- Quantum Safe IPsec VPN | **172**

IPsec VPN Configuration Overview

SUMMARY

Read this topic to learn about VPN configuration in Junos OS.

IN THIS SECTION

- IPsec VPN with Autokey IKE Configuration Overview | **110**
- Recommended Configuration Options for Site-to-Site VPN with Static IP Addresses | **111**
- Recommended Configuration Options for Site-to-Site or Dialup VPNs with Dynamic IP Addresses | **112**
- Understanding IPsec VPNs with Dynamic Endpoints | **114**
- Understanding IKE Identity Configuration | **116**
- Configuring Remote IKE IDs for Site-to-Site VPNs | **118**
- Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | **118**
- Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Firewall | **120**
- Configuring IPsec VPN Using the VPN Wizard | **127**

A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. An IPsec tunnel is created between two participant devices to secure VPN communication.

IPsec VPN with Autokey IKE Configuration Overview

IPsec VPN negotiation occurs in two phases. In Phase 1, participants establish a secure channel in which to negotiate the IPsec security association (SA). In Phase 2, participants negotiate the IPsec SA for authenticating traffic that will flow through the tunnel.

This overview describes the basic steps to configure a route-based VPN using autokey IKE (preshared keys or certificates).

To configure a route-based IPsec VPN using autokey IKE:

1. Configure interfaces, security zones, and address book information.

   (For route-based VPNs) Configure a secure tunnel st0.x interface. Configure routing on the device.
2. Configure Phase 1 of the IPsec VPN tunnel.
   a. (Optional) Configure a custom IKE Phase 1 proposal. This step is optional, as you can use a predefined IKE Phase 1 proposal set (Standard, Compatible, or Basic).
   b. Configure an IKE policy that references either your custom IKE Phase 1 proposal or a predefined IKE Phase 1 proposal set. Specify autokey IKE preshared key or certificate information. Specify the mode (main or aggressive) for the Phase 1 exchanges.
   c. Configure an IKE gateway that references the IKE policy. Specify the IKE IDs for the local and remote devices. If the IP address of the remote gateway is not known, specify how the remote gateway is to be identified.
3. Configure Phase 2 of the IPsec VPN tunnel.
   a. (Optional) Configure a custom IPsec Phase 2 proposal. This step is optional, as you can use a predefined IPsec Phase 2 proposal set (Standard, Compatible, or Basic).
   b. Configure an IPsec policy that references either your custom IPsec Phase 2 proposal or a predefined IPsec Phase 2 proposal set. Specify perfect forward secrecy (PFS) keys.
   c. Configure an IPsec VPN tunnel that references both the IKE gateway and the IPsec policy. Specify the proxy IDs to be used in Phase 2 negotiations.

      (For route-based VPNs) Bind the secure tunnel interface st0.x to the IPsec VPN tunnel.
4. Update your global VPN settings.

SEE ALSO

Understanding Route-Based IPsec VPNs | 320

Recommended Configuration Options for Site-to-Site VPN with Static IP Addresses

Table 23 on page 111 lists the configuration options for a generic site-to-site VPN between two security devices with static IP addresses.

Table 23: Recommended Configuration for Site-to-Site VPN with Static IP Addresses

Configuration Option	Comment
IKE configuration options:	
Main mode	Used when peers have static IP addresses.
RSA or DSA certificates	RSA or DSA certificates can be used on the local device. Specify the type of certificate (PKCS7 or X.509) on the peer.
Diffie-Hellman (DH) group 14	DH group 14 provides more security than DH groups 1, 2, or 5.
Advanced Encryption Standard (AES) encryption	AES is cryptographically stronger than Data Encryption Standard (DES) and Triple DES (3DES) when key lengths are equal. Approved encryption algorithm for Federal Information Processing Standards (FIPS) and Common Criteria EAL4 standards.
Secure Hash Algorithm 256 (SHA-256) authentication	SHA-256 provides more cryptographic security than SHA-1 or Message Digest 5 (MD5) .
IPsec configuration options:	
Perfect Forward Secrecy (PFS) DH group 14	PFS DH group 14 provides increased security because the peers perform a second DH exchange to produce the key used for IPsec encryption and decryption.
Encapsulating Security Payload (ESP) protocol	ESP provides both confidentiality through encryption and encapsulation of the original IP packet and integrity through authentication.

Table 23: Recommended Configuration for Site-to-Site VPN with Static IP Addresses *(Continued)*

Configuration Option	Comment
AES encryption	AES is cryptographically stronger than DES and 3DES when key lengths are equal. Approved encryption algorithm for FIPS and Common Criteria EAL4 standards.
SHA-256 authentication	SHA-256 provides more cryptographic security than SHA-1 or MD5.
Anti-replay protection	Enabled by default. Disabling this feature might resolve compatibility issues with third-party peers.

SEE ALSO

IPsec Overview | 12

Recommended Configuration Options for Site-to-Site or Dialup VPNs with Dynamic IP Addresses

Table 24 on page 112 lists the configuration options for a generic site-to-site or dialup VPN, where the peer devices have dynamic IP addresses.

Table 24: Recommended Configuration for Site-to-Site or Dialup VPNs with Dynamic IP Addresses

Configuration Option	Comment
IKE configuration options:	
Main mode	Used with certificates.
2048-bit certificates	RSA or DSA certificates can be used. Specify the certificate to be used on the local device. Specify the type of certificate (PKCS7 or X.509) on the peer.

Table 24: Recommended Configuration for Site-to-Site or Dialup VPNs with Dynamic IP Addresses *(Continued)*

Configuration Option	Comment
Diffie-Hellman (DH) group 14	DH group 14 provides more security than DH groups 1, 2, or 5.
Advanced Encryption Standard (AES) encryption	AES is cryptographically stronger than Data Encryption Standard (DES) and Triple DES (3DES) when key lengths are equal. Approved encryption algorithm for Federal Information Processing Standards (FIPS) and Common Criteria EAL4 standards.
Secure Hash Algorithm 256 (SHA-256) authentication	SHA-256 provides more cryptographic security than SHA-1 or Message Digest 5 (MD5).
IPsec configuration options:	
Perfect Forward Secrecy (PFS) DH group 14	PFS DH group 14 provides increased security because the peers perform a second DH exchange to produce the key used for IPsec encryption and decryption.
Encapsulating Security Payload (ESP) protocol	ESP provides both confidentiality through encryption and encapsulation of the original IP packet and integrity through authentication.
AES encryption	AES is cryptographically stronger than DES and 3DES when key lengths are equal. Approved encryption algorithm for FIPS and Common Criteria EAL4 standards.
SHA-256 authentication	SHA-256 provides more cryptographic security than SHA-1 or MD5.
Anti-replay protection	Enabled by default. Disabling this might resolve compatibility issues with third-party peers.

SEE ALSO

IPsec Overview | 12

Understanding IPsec VPNs with Dynamic Endpoints

IN THIS SECTION

- Overview | 114
- IKE Identity | 114
- Aggressive Mode for IKEv1 Policy | 115
- IKE Policies and External Interfaces | 115
- NAT | 115
- Group and Shared IKE IDs | 115

Overview

An IPsec VPN peer can have an IP address that is not known to the peer with which it is establishing the VPN connection. For example, a peer can have an IP address dynamically assigned by means of Dynamic Host Configuration Protocol (DHCP). This could be the case with a remote access client in a branch or home office or a mobile device that moves between different physical locations. Or, the peer can be located behind a NAT device that translates the peer's original source IP address into a different address. A VPN peer with an unknown IP address is referred to as a *dynamic endpoint* and a VPN established with a dynamic endpoint is referred to as a *dynamic endpoint VPN*.

On SRX Series Firewalls, IKEv1 or IKEv2 is supported with dynamic endpoint VPNs. Dynamic endpoint VPNs on SRX Series Firewalls support IPv4 traffic on secure tunnels. Dynamic endpoint VPNs on SRX Series Firewalls support IPv6 traffic on secure tunnels.

IPv6 traffic is not supported for AutoVPN networks.

The following sections describe items to note when configuring a VPN with a dynamic endpoint.

IKE Identity

On the dynamic endpoint, an IKE identity must be configured for the device to identify itself to its peer. The local identity of the dynamic endpoint is verified on the peer. By default, the SRX Series Firewall expects the IKE identity to be one of the following:

- When certificates are used, a distinguished name (DN) can be used to identify users or an organization.
- A hostname or fully qualified domain name (FQDN) that identifies the endpoint.

- A user fully qualified domain name (UFQDN), also known as *user-at-hostname*. This is a string that follows the e-mail address format.

Aggressive Mode for IKEv1 Policy

When IKEv1 is used with dynamic endpoint VPNs, the IKE policy must be configured for aggressive mode.

IKE Policies and External Interfaces

All dynamic endpoint gateways configured on SRX Series Firewalls that use the same external interface can use different IKE policies, but the IKE policies must use the same IKE proposal. This applies to IKEv1 and IKEv2.

NAT

If the dynamic endpoint is behind a NAT device, NAT-T must be configured on the SRX Series Firewall. NAT keepalives might be required to maintain the NAT translation during the connection between the VPN peers. By default, NAT-T is enabled on SRX Series Firewalls and NAT keepalives are sent at 20-second intervals.

Group and Shared IKE IDs

You can configure an individual VPN tunnel for each dynamic endpoint. For IPv4 dynamic endpoint VPNs, you can use the group IKE ID or shared IKE ID features to allow a number of dynamic endpoints to share an IKE gateway configuration.

The group IKE ID allows you to define a common part of a full IKE ID for all dynamic endpoints, such as “example.net.” A user-specific part, such as the username “Bob,” concatenated with the common part forms a full IKE ID (Bob.example.net) that uniquely identifies each user connection.

The shared IKE ID allows dynamic endpoints to share a single IKE ID and preshared key.

SEE ALSO

Example: Configuring NAT-T with Dynamic Endpoint VPN | 568

Understanding IKE Identity Configuration

IN THIS SECTION

- IKE ID Types | 116
- Remote IKE IDs and Site-to-Site VPNs | 117
- Remote IKE IDs and Dynamic Endpoint VPNs | 117
- Local IKE ID of the SRX Series Firewall | 117

The IKE identification (IKE ID) is used for validation of VPN peer devices during IKE negotiation. The IKE ID received by the SRX Series Firewall from a remote peer can be an IPv4 or IPv6 address, a hostname, a fully qualified domain name (FQDN), a user FQDN (UFQDN), or a distinguished name (DN). The IKE ID sent by the remote peer needs to match what is expected by the SRX Series Firewall. Otherwise, IKE ID validation fails and the VPN is not established.

IKE ID Types

The SRX Series Firewalls support the following types of IKE identities for remote peers:

- An IPv4 or IPv6 address is commonly used with site-to-site VPNs, where the remote peer has a static IP address.
- A hostname is a string that identifies the remote peer system. This can be an FQDN that resolves to an IP address. It can also be a partial FQDN that is used in conjunction with an IKE user type to identify a specific remote user.

  When a hostname is configured instead of an IP address, the committed configuration and subsequent tunnel establishment is based on the currently-resolved IP address. If the remote peer's IP address changes, the configuration is no longer valid.
- A UFQDN is a string that follows the same format as an e-mail address, such as `user@example.com`.
- A DN is a name used with digital certificates to uniquely identify a user. For example, a DN can be "CN=user, DC=example, DC=com." Optionally, you can use the `container` keyword to specify that the order of the fields in a DN and their values exactly match the configured DN, or use the `wildcard` keyword to specify that the values of fields in a DN must match but the order of the fields does not matter.

  You can now configure only one dynamic DN attribute among `container-string` and `wildcard-string` at `[edit security ike gateway` *`gateway_name`* `dynamic distinguished-name]` hierarchy. If you try configuring the

second attribute after you configure the first attribute, the first attribute is replaced with the second attribute. Before your upgrade your device, you must remove one of the attributes if you have configured both the attributes.

- An IKE user type can be used with AutoVPN and remote access VPNs when there are multiple remote peers connecting to the same VPN gateway on the SRX Series Firewall. Configure `ike-user-type group-ike-id` to specify a group IKE ID or `ike-user-type shared-ike-id` to specify a shared IKE ID.

Remote IKE IDs and Site-to-Site VPNs

For site-to-site VPNs, the remote peer's IKE ID can be the IP address of the egress network interface card, a loopback address, a hostname, or a manually configured IKE ID, depending on the configuration of the peer device.

By default, SRX Series Firewalls expect the remote peer's IKE ID to be the IP address configured with the `set security ike gateway` *`gateway-name`* `address` configuration. If the remote peer's IKE ID is a different value, you need to configure the `remote-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level.

For example, an IKE gateway on the SRX Series Firewalls is configured with the `set security ike gateway remote-gateway address 203.0.113.1` command. However, the IKE ID sent by the remote peer is `host.example.net`. There is a mismatch between what the SRX Series Firewall expects for the remote peer's IKE ID (203.0.113.1) and the actual IKE ID (`host.example.net`) sent by the peer. In this case, IKE ID validation fails. Use the `set security ike gateway remote-gateway remote-identity hostname host.example.net` to match the IKE ID received from the remote peer.

Remote IKE IDs and Dynamic Endpoint VPNs

For dynamic endpoint VPNs, the remote peer's expected IKE ID is configured with the options at the [`edit security ike gateway` *`gateway-name`* `dynamic`] hierarchy level. For AutoVPN, `hostname` combined with `ike-user-type group-ike-id` can be used where there are multiple peers that have a common domain name. If certificates are used for verifying the peer, a DN can be configured.

Local IKE ID of the SRX Series Firewall

By default, the SRX Series Firewall uses the IP address of its external interface to the remote peer as its IKE ID. This IKE ID can be overridden by configuring the `local-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level. If you need to configure the `local-identity` statement on an SRX Series Firewall, make sure that the configured IKE ID matches the IKE ID expected by the remote peer.

SEE ALSO

Understanding Spoke Authentication in AutoVPN Deployments | 925

Configuring Remote IKE IDs for Site-to-Site VPNs

By default, SRX Series Firewalls validate the IKE ID received from the peer with the IP address configured for the IKE gateway. In certain network setups, the IKE ID received from the peer (which can be an IPv4 or IPv6 address, fully qualified domain name [FQDN], distinguished name, or e-mail address) does not match the IKE gateway configured on the SRX Series Firewall. This can lead to a Phase 1 validation failure.

To modify the configuration of the SRX Series Firewall or the peer device for the IKE ID that is used:

- On the SRX Series Firewall, configure the `remote-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level to match the IKE ID that is received from the peer. Values can be an IPv4 or IPv6 address, FQDN, distinguished name, or e-mail address.

  If you do not configure `remote-identity`, the device uses the IPv4 or IPv6 address that corresponds to the remote peer by default.

- On the peer device, ensure that the IKE ID is the same as the `remote-identity` configured on the SRX Series Firewall. If the peer device is an SRX Series Firewall, configure the `local-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level. Values can be an IPv4 or IPv6 address, FQDN, distinguished name, or e-mail address.

SEE ALSO

Understanding NAT-T | 530

Example: Configuring a Route-Based VPN with the Responder behind a NAT Device | 531

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls

OSPFv3 does not have a built-in authentication method and relies on the IP Security (IPsec) suite to provide this functionality. IPsec provides authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. You can use IPsec to secure specific OSPFv3 interfaces and virtual links and to provide encryption for OSPF packets.

OSPFv3 uses the IP authentication header (AH) and the IP Encapsulating Security Payload (ESP) portions of the IPsec protocol to authenticate routing information between peers. AH can provide connectionless integrity and data origin authentication. It also provides protection against replays. AH authenticates as much of the IP header as possible, as well as the upper-level protocol data. However, some IP header fields might change in transit. Because the value of these fields might not be predictable by the sender, they cannot be protected by AH. ESP can provide encryption and limited traffic flow confidentiality or connectionless integrity, data origin authentication, and an anti-replay service.

IPsec is based on security associations (SAs). An SA is a set of IPsec specifications that are negotiated between devices that are establishing an IPsec relationship. This simplex connection provides security services to the packets carried by the SA. These specifications include preferences for the type of authentication, encryption, and IPsec protocol to be used when establishing the IPsec connection. An SA is used to encrypt and authenticate a particular flow in one direction. Therefore, in normal bidirectional traffic, the flows are secured by a pair of SAs. An SA to be used with OSPFv3 must be configured manually and use transport mode. Static values must be configured on both ends of the SA.

To configure IPsec for OSPF or OSPFv3, first define a manual SA with the `security-association` *`sa-name`* option at the [`edit security ipsec`] hierarchy level. This feature only supports bidirectional manual key SAs in transport mode. Manual SAs require no negotiation between the peers. All values, including the keys, are static and specified in the configuration. Manual SAs statically define the security parameter index (SPI) values, algorithms, and keys to be used and require matching configurations on both endpoints (OSPF or OSPFv3 peers). As a result, each peer must have the same configured options for communication to take place.

The actual choice of encryption and authentication algorithms is left to your IPsec administrator; however, we have the following recommendations:

- Use ESP with null encryption to provide authentication to protocol headers but not to the IPv6 header, extension headers, and options. With null encryption, you are choosing not to provide encryption on protocol headers. This can be useful for troubleshooting and debugging purposes. For more information about null encryption, see RFC 2410, *The NULL Encryption Algorithm and Its Use with IPsec*.
- Use ESP with DES or 3DES for full confidentiality.
- Use AH to provide authentication to protocol headers, immutable fields in IPv6 headers, and extension headers and options.

The configured SA is applied to the OSPF or OSPFv3 configurations as follows:

- For an OSPF or OSPFv3 interface, include the `ipsec-sa` *`name`* statement at the [`edit protocols ospf area` *`area-id`* `interface` *`interface-name`*] or [`edit protocols ospf3 area` *`area-id`* `interface` *`interface-name`*] hierarchy level. Only one IPsec SA name can be specified for an OSPF or OSPFv3 interface; however, different OSPF/OSPFv3 interfaces can specify the same IPsec SA.

- For an OSPF or OSPFv3 virtual link, include the `ipsec-sa` *`name`* statement at the [`edit protocols ospf area` *`area-id`* `virtual-link neighbor-id` *`router-id`* `transit-area` *`area-id`*] or [`edit protocols ospf3 area` *`area-id`* `virtual-link neighbor-id` *`router-id`* `transit-area` *`area-id`*] hierarchy level. You must configure the same IPsec SA for all virtual links with the same remote endpoint address.

The following restrictions apply to IPsec authentication for OSPF or OSPFv3 on SRX Series Firewalls:

- Manual VPN configurations that are configured at the [`edit security ipsec vpn` *`vpn-name`* `manual`] hierarchy level cannot be applied to OSPF or OSPFv3 interfaces or virtual links to provide IPsec authentication and confidentiality.
- You cannot configure IPsec for OSPF or OSPFv3 authentication if there is an existing IPsec VPN configured on the device with the same local and remote addresses.
- IPsec for OSPF or OSPFv3 authentication is not supported over secure tunnel st0 interfaces.
- Rekeying of manual keys is not supported.
- Dynamic Internet Key Exchange (IKE) SAs are not supported.
- Only IPsec transport mode is supported. In transport mode, only the payload (the data you transfer) of the IP packet is encrypted, authenticated, or both. Tunnel mode is not supported.
- Because only bidirectional manual SAs are supported, all OSPFv3 peers must be configured with the same IPsec SA. You configure a manual bidirectional SA at the [`edit security ipsec`] hierarchy level.
- You must configure the same IPsec SA for all virtual links with the same remote endpoint address.

SEE ALSO

IPsec Overview | 12

Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Firewall

IN THIS SECTION

- Requirements | 121
- Overview | 121

Configuration | 122
Verification | 126

This example shows how to configure and apply a manual security association (SA) to an OSPF interface.

Requirements

Before you begin:

- Configure the device interfaces.
- Configure the router identifiers for the devices in your OSPF network.
- Control OSPF designated router election.
- Configure a single-area OSPF network.
- Configure a multiarea OSPF network.

Overview

You can use IPsec authentication for both OSPF and OSPFv3. You configure the manual SA separately and apply it to the applicable OSPF configuration. Table 25 on page 121 lists the parameters and values configured for the manual SA in this example.

Table 25: Manual SA for IPsec OSPF Interface Authentication

Parameter	Value
SA name	sa1
Mode	transport
Direction	bidirectional
Protocol	AH
SPI	256

Table 25: Manual SA for IPsec OSPF Interface Authentication *(Continued)*

Parameter	Value
Authentication algorithm	hmac-md5-96
Key	(ASCII) 123456789012abc
Encryption algorithm	des
Key	(ASCII) cba210987654321

Configuration

IN THIS SECTION

- Configuring a Manual SA | **122**
- Enabling IPsec Authentication for an OSPF Interface | **125**

Configuring a Manual SA

CLI Quick Configuration

To quickly configure a manual SA to be used for IPsec authentication on an OSPF interface, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set security ipsec security-association sa1
set security ipsec security-association sa1 mode transport
set security ipsec security-association sa1 manual direction bidirectional
set security ipsec security-association sa1 manual direction bidirectional protocol ah
set security ipsec security-association sa1 manual direction bidirectional spi 256
set security ipsec security-association sa1 manual direction bidirectional authentication
algorithm hmac-md5-96 key ascii-text 123456789012abc
```

```
set security ipsec security-association sa1 manual direction bidirectional encryption algorithm
des key ascii-text cba210987654321
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure a manual SA:

1. Specify a name for the SA.

```
[edit]
user@host# edit security ipsec security-association sa1
```

2. Specify the mode of the manual SA.

```
[edit security ipsec security-association sa1]
user@host# set mode transport
```

3. Configure the direction of the manual SA.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional
```

4. Configure the IPsec protocol to use.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional protocol ah
```

5. Configure the value of the SPI.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional spi 256
```

6. Configure the authentication algorithm and key.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional authentication algorithm hmac-md5-96 key ascii-
text 123456789012abc
```

7. Configure the encryption algorithm and key.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional encryption algorithm des key ascii-text
cba210987654321
```

Results

Confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

After you configure the password, you do not see the password itself. The output displays the encrypted form of the password you configured.

```
[edit]
user@host# show security ipsec
security-association sa1 {
    mode transport;
    manual {
        direction bidirectional {
            protocol ah;
            spi 256;
            authentication {
                algorithm hmac-md5-96;
                key ascii-text "$9$AP5Hp1RcylMLxSygoZUHk1REhKMVwY2oJx7jHq.zF69A0OR"; ## SECRET-
DATA
            }
            encryption {
                algorithm des;
                key ascii-text "$9$AP5Hp1RcylMLxSygoZUHk1REhKMVwY2oJx7jHq.zF69A0OR"; ## SECRET-
DATA
            }
        }
```

```
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Enabling IPsec Authentication for an OSPF Interface

CLI Quick Configuration

To quickly apply a manual SA used for IPsec authentication to an OSPF interface, copy the following command, paste it into a text file, change any details necessary to match your network configuration, copy and paste the command into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set protocols ospf area 0.0.0.0 interface so-0/2/0 ipsec-sa sa1
```

Step-by-Step Procedure

To enable IPsec authentication for an OSPF interface:

1. Create an OSPF area.

   To specify OSPFv3, include the ospf3 statement at the [edit protocols] hierarchy level.

   ```
   [edit]
   user@host# edit protocols ospf area 0.0.0.0
   ```

2. Specify the interface.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@host# edit interface so-0/2/0
   ```

3. Apply the IPsec manual SA.

   ```
   [edit protocols ospf area 0.0.0.0 interface so-0/2/0.0]
   user@host# set ipsec-sa sa1
   ```

Results

Confirm your configuration by entering the **show ospf interface detail** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

```
[edit]
user@host# show protocols ospf
area 0.0.0.0 {
    interface so-0/2/0.0 {
         ipsec-sa sa1;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying the IPsec Security Association Settings | **126**
- Verifying the IPsec Security Association on the OSPF Interface | **127**

Confirm that the configuration is working properly.

Verifying the IPsec Security Association Settings

Purpose

Verify the configured IPsec security association settings. Verify the following information:

- The Security association field displays the name of the configured security association.
- The SPI field displays the value you configured.
- The Mode field displays transport mode.
- The Type field displays manual as the type of security association.

Action

From operational mode, enter the **show ospf interface detail** command.

Verifying the IPsec Security Association on the OSPF Interface

Purpose

Verify that the IPsec security association that you configured has been applied to the OSPF interface. Confirm that the IPsec SA name field displays the name of the configured IPsec security association.

Action

From operational mode, enter the **show ospf interface detail** command for OSPF, and enter the **show ospf3 interface detail** command for OSPFv3.

SEE ALSO

Understanding IPsec SA Configuration for Group VPNv1 | **604**

Configuring IPsec VPN Using the VPN Wizard

The VPN Wizard enables you to perform basic IPsec VPN configuration, including both Phase 1 and Phase 2. For more advanced configuration, use the J-Web interface or the CLI. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

To configure IPsec VPN using the VPN Wizard:

1. Select `Configure>Device Setup>VPN` in the J-Web interface.
2. Click the Launch VPN Wizard button.
3. Follow the wizard prompts.

The upper left area of the wizard page shows where you are in the configuration process. The lower left area of the page shows field-sensitive help. When you click a link under the Resources heading, the document opens in your browser. If the document opens in a new tab, be sure to close only the tab (not the browser window) when you close the document.

SEE ALSO

IPsec Overview | 12

Internet Key Exchange | 2

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.4R1	Starting in Junos OS Release 19.4R1, you can now configure only one dynamic DN attribute among `container-string` and `wildcard-string` at `[edit security ike gateway gateway_name dynamic distinguished-name]` hierarchy. If you try configuring the second attribute after you configure the first attribute, the first attribute is replaced with the second attribute. Before your upgrade your device, you must remove one of the attributes if you have configured both the attributes.
15.1X49-D80	Starting with Junos OS Release 15.1X49-D80, dynamic endpoint VPNs on SRX Series Firewalls support IPv6 traffic on secure tunnels.
12.3X48-D40	Starting with Junos OS Release 12.3X48-D40, Junos OS Release 15.1X49-D70, and Junos OS Release 17.3R1, all dynamic endpoint gateways configured on SRX Series Firewalls that use the same external interface can use different IKE policies, but the IKE policies must use the same IKE proposal.

RELATED DOCUMENTATION

Route-Based VPN with IKEv2 | 347

Chassis Cluster HA Control Link Encryption

SUMMARY

Learn how to configure chassis cluster high availability control link encryption.

Connect the dedicated control ports on node 0 and node 1. Connect the user defined fabricated ports on node 0 and node 1. To configure two chassis in cluster mode, follow the below steps:

Enable chassis cluster mode on both the nodes, see Chassis Cluster Configuration Overview.

1. After you enable the chassis cluster, on the device 1 configure high availability (HA) link encryption as shown, then commit and reboot. Device 1 needs to be configured with both node0 and node1 HA link encryption before commit and reboot.

```
[edit]
user@host# set groups node0 security ike proposal HA authentication-method pre-shared-keys
user@host# set groups node0 security ike proposal HA dh-group group20
user@host# set groups node0 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node0 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ike policy HA proposals HA
user@host# prompt groups node0 security ike policy HA pre-shared-key ascii-text
This Should Be A Strong And Secure Key
Retype This Should Be A Strong And Secure Key
user@host# set groups node0 security ike gateway HA ike-policy HA
user@host# set groups node0 security ike gateway HA version v2-only
user@host# set groups node0 security ipsec proposal HA protocol esp
user@host# set groups node0 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node0 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node0 security ipsec policy HA proposal HA
user@host# set groups node0 security ipsec vpn HA ha-link-encryption
user@host# set groups node0 security ipsec vpn HA ike gateway HA
user@host# set groups node0 security ipsec vpn HA ike ipsec-policy HA
user@host# set groups node1 security ike proposal HA authentication-method pre-shared-keys
user@host# set groups node1 security ike proposal HA dh-group group20
user@host# set groups node1 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node1 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ike policy HA proposals HA
user@host# prompt groups node1 security ike policy HA pre-shared-key ascii-text
New ascii-text(secret): juniper
Retype This Should Be A Strong And Secure Key
user@host# set groups node1 security ike gateway HA ike-policy HA
user@host# set groups node1 security ike gateway HA version v2-only
user@host# set groups node1 security ipsec proposal HA protocol esp
user@host# set groups node1 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node1 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node1 security ipsec policy HA proposals HA
```

```
user@host# set groups node1 security ipsec vpn HA ha-link-encryption
user@host# set groups node1 security ipsec vpn HA ike gateway HA
user@host# set groups node1 security ipsec vpn HA ike ipsec-policy HA
user@host# commit
user@host> request system reboot
```

2. To proceed further with device 2 configuration and commit, you need to ensure device 1 and device 2 are not reachable to each other. One way to achieve this is to power off device 1 at this point.
3. After the device 2 is up, configure HA link encryption as shown in sample configuration below on device 2. Device 2 needs to be configured with both node0 and node1 HA link encryption configuration. Commit on node1 (device 2), and finally reboot node1 (device 2).

```
[edit]
user@host# set groups node0 security ike proposal HA authentication-method pre-shared-keys
user@host# set groups node0 security ike proposal HA dh-group group20
user@host# set groups node0 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node0 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ike policy HA proposals HA
user@host# prompt groups node0 security ike policy HA pre-shared-key ascii-text
This Should Be A Strong And Secure Key
Retype This Should Be A Strong And Secure Key
user@host# set groups node0 security ike gateway HA ike-policy HA
user@host# set groups node0 security ike gateway HA version v2-only
user@host# set groups node0 security ipsec proposal HA protocol esp
user@host# set groups node0 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node0 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node0 security ipsec policy HA proposal HA
user@host# set groups node0 security ipsec vpn HA ha-link-encryption
user@host# set groups node0 security ipsec vpn HA ike gateway HA
user@host# set groups node0 security ipsec vpn HA ike ipsec-policy HA
user@host# set groups node1 security ike proposal HA authentication-method pre-shared-keys
user@host# set groups node1 security ike proposal HA dh-group group20
user@host# set groups node1 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node1 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ike policy HA proposals HA
user@host# prompt groups node1 security ike policy HA pre-shared-key ascii-text
New ascii-text(secret): juniper
Retype This Should Be A Strong And Secure Key
user@host# set groups node1 security ike gateway HA ike-policy HA
user@host# set groups node1 security ike gateway HA version v2-only
user@host# set groups node1 security ipsec proposal HA protocol esp
```

```
user@host# set groups node1 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node1 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node1 security ipsec policy HA proposals HA
user@host# set groups node1 security ipsec vpn HA ha-link-encryption
user@host# set groups node1 security ipsec vpn HA ike gateway HA
user@host# set groups node1 security ipsec vpn HA ike ipsec-policy HA
user@host# commit
user@host> request system reboot
```

NOTE: To enable HA link encryption on node1 in step 3, the other node needs to be in lost state for the commit to go through. So this timing needs to be taken care by you, else step 3 needs to be redone until enabling HA link encryption on node1 commit goes through.

Configure IPsec VPN with OCSP for Certificate Revocation Status

IN THIS SECTION

- Requirements | 132
- Overview | 132
- Configuration | 135
- Verification | 146

This example shows how to improve security by configuring two peers using the Online Certificate Status Protocol (OCSP) to check the revocation status of the certificates used in Phase 1 negotiations for the IPsec VPN tunnel.

Requirements

On each device:

- Obtain and enroll a local certificate. This can be done either manually or by using the Simple Certificate Enrollment Protocol (SCEP).
- Optionally, enable automatic renewal of the local certificate.
- Configure security policies to permit traffic to and from the peer device.

Overview

IN THIS SECTION

- Topology | **135**

On both peers, a certificate authority (CA) profile OCSP-ROOT is configured with the following options:

- CA name is OCSP-ROOT.
- Enrollment URL is http://10.1.1.1:8080/scep/OCSP-ROOT/. This is the URL where SCEP requests to the CA are sent.
- The URL for the OCSP server is http://10.157.88.56:8210/OCSP-ROOT/.
- OCSP is used first to check the certificate revocation status. If there is no response from the OCSP server, then the certificate revocation list (CRL) is used to check the status. The CRL URL is http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45.
- The CA certificate received in an OCSP response is not checked for certificate revocation. Certificates received in an OCSP response generally have shorter lifetimes and a revocation check is not required.

Table 26 on page 133 shows the Phase 1 options used in this example.

Table 26: Phase 1 Options for OCSP Configuration Example

Option	Peer A	Peer B
IKE proposal	ike_prop	ike_prop
Authentication method	RSA signatures	RSA signatures
DH group	group2	group2
Authentication algorithm	SHA 1	SHA 1
Encryption algorithm	3DES CBC	3DES CBC
IKE policy	ike_policy	ike_policy
Mode	aggressive	aggressive
Proposal	ike_prop	ike_prop
Certificate	local-certificate localcert1	local-certificate localcert1
IKE gateway	jsr_gateway	jsr_gateway
Policy	ike_policy	ike_policy
Gateway address	198.51.100.50	192.0.2.50
Remote identity	localcert11.example.net	-
Local identity	-	localcert11.example.net
External interface	reth1	ge-0/0/2.0

Table 26: Phase 1 Options for OCSP Configuration Example *(Continued)*

Option	Peer A	Peer B
Version	v2	v2

Table 27 on page 134 shows the Phase 2 options used in this example.

Table 27: Phase 2 Options for OCSP Configuration Example

Option	Peer A	Peer B
IPsec proposal	ipsec_prop	ipsec_prop
Protocol	ESP	ESP
Authentication algorithm	HMAC SHA1-96	HMAC SHA1-96
Encryption algorithm	3DES CBC	3DES CBC
Lifetime seconds	1200	1200
Lifetime kilobytes	150,000	150,000
IPsec policy	ipsec_policy	ipsec_policy
PFC keys	group2	group2
Proposal	ipsec_prop	ipsec_prop
VPN	test_vpn	test_vpn
Bind interface	st0.1	st0.1
IKE gateway	jsr_gateway	jsr_gateway

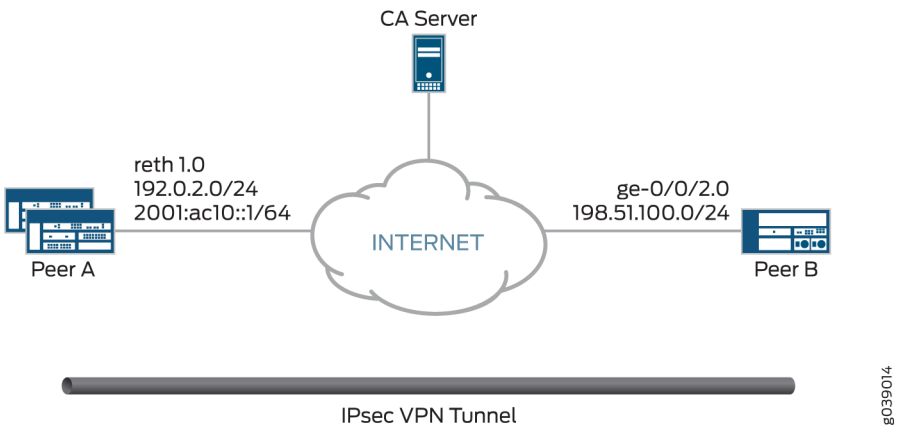
Table 27: Phase 2 Options for OCSP Configuration Example *(Continued)*

Option	Peer A	Peer B
Policy	ipsec_policy	ipsec_policy
Establish tunnels	-	immediately

Topology

Figure 15 on page 135 shows the peer devices that are configured in this example.

Figure 15: OCSP Configuration Example



Configuration

IN THIS SECTION

- Configuring Peer A | 136
- Configuring Peer B | 141

Configuring Peer A

CLI Quick Configuration

To quickly configure VPN peer A to use OCSP, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/3 gigether-options redundant-parent reth1
set interfaces ge-9/0/3 gigether-options redundant-parent reth1
set interfaces lo0 unit 0 family inet address 172.16.1.100/24
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 192.0.2.50/24
set interfaces st0 unit 1 family inet address 172.18.1.100/24
set security pki ca-profile OCSP-ROOT ca-identity OCSP-ROOT
set security pki ca-profile OCSP-ROOT enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set security pki ca-profile OCSP-ROOT revocation-check ocsp url http://10.157.88.56:8210/OCSP-
ROOT/
set security pki ca-profile OCSP-ROOT revocation-check use-ocsp
set security pki ca-profile OCSP-ROOT revocation-check ocsp disable-responder-revocation-check
set security pki ca-profile OCSP-ROOT revocation-check ocsp connection-failure fallback-crl
set security pki ca-profile OCSP-ROOT revocation-check crl url http://10.1.1.1:8080/crl-as-der/
currentcrl-45.crlid=45
set security ike proposal ike_prop authentication-method rsa-signatures
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_policy mode aggressive
set security ike policy ike_policy proposals ike_prop
set security ike policy ike_policy certificate local-certificate localcert1
set security ike gateway jsr_gateway ike-policy ike_policy
set security ike gateway jsr_gateway address 198.51.100.50
set security ike gateway jsr_gateway remote-identity hostname localcert11.example.net
set security ike gateway jsr_gateway external-interface reth1
set security ike gateway jsr_gateway version v2-only
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec proposal ipsec_prop lifetime-seconds 1200
set security ipsec proposal ipsec_prop lifetime-kilobytes 150000
```

```
set security ipsec policy ipsec_policy perfect-forward-secrecy keys group2
set security ipsec policy ipsec_policy proposals ipsec_prop
set security ipsec vpn test_vpn bind-interface st0.1
set security ipsec vpn test_vpn ike gateway jsr_gateway
set security ipsec vpn test_vpn ike ipsec-policy ipsec_policy
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure VPN peer A to use OCSP:

1. Configure interfaces.

```
[edit interfaces]
set ge-0/0/3 gigether-options redundant-parent reth1
set ge-9/0/3 gigether-options redundant-parent reth1
set lo0 unit 0 family inet address 172.16.1.100/24
set  lo0 redundant-pseudo-interface-options redundancy-group 1
set reth1 redundant-ether-options redundancy-group 1
set reth1 unit 0 family inet address 192.0.2.0/24
set st0 unit 1 family inet address 172.18.1.100/24
```

2. Configure the CA profile.

```
[edit security pki ca-profile OCSP-ROOT]
set ca-identity OCSP-ROOT
set enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set revocation-check ocsp url http://10.157.88.56:8210/OCSP-ROOT/
set  revocation-check use-ocsp
set revocation-check ocsp disable-responder-revocation-check
set revocation-check ocsp connection-failure fallback-crl
set revocation-check crl url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike_prop]
set authentication-method rsa-signatures
```

```
set dh-group group2
set authentication-algorithm sha1
set encryption-algorithm 3des-cbc

[edit security ike policy ike_policy]
set mode aggressive
set proposals ike_prop
set certificate local-certificate localcert1

[edit security ike gateway jsr_gateway]
set ike-policy ike_policy
set address 198.51.100.50
set remote-identity hostname localcert11.example.net
set external-interface reth1
set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_prop]
set protocol esp
set authentication-algorithm hmac-sha1-96
set encryption-algorithm 3des-cbc
set lifetime-seconds 1200
set lifetime-kilobytes 150000

[edit security ipsec policy ipsec_policy]
set perfect-forward-secrecy keys group2
set proposals ipsec_prop

[edit security ipsec vpn test_vpn]
set bind-interface st0.1
set ike gateway jsr_gateway
set ike ipsec-policy ipsec_policy
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security pki ca-profile OCSP-ROOT`, `show security ike`, and `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/3 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-9/0/3 {
    gigether-options {
        redundant-parent reth1;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 172.16.1.100/24;
        }
    }
    redundant-pseudo-interface-options {
        redundancy-group 1;
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.0.2.0/24;
        }
    }
}
st0 {
    unit 1 {
        family inet {
            address 172.18.1.100/24;
```

```
        }
    }
}
[edit]
user@host# show security pki ca-profile OCSP-ROOT
ca-identity OCSP-ROOT;
enrollment {
    url http://10.1.1.1:8080/scep/OCSP-ROOT/;
}
revocation-check {
    crl {
        url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45;
    }
    ocsp {
        disable-responder-revocation-check;
        url http://10.157.88.56:8210/OCSP-ROOT/;
    }
    use-ocsp;
}
[edit]
user@host# show security ike
proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_policy {
    mode aggressive;
    proposals ike_prop;
    certificate {
        local-certificate localcert1;
    }
}
gateway jsr_gateway {
    ike-policy ike_policy;
    address 10.10.2.50;
    remote-identity hostname localcert11.example.net;
    external-interface reth1;
    version v2-only;
}
[edit]
user@host# show security ipsec
```

```
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 1200;
    lifetime-kilobytes 150000;
}
policy ipsec_policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn test_vpn {
    bind-interface st0.1;
    ike {
        gateway jsr_gateway;
        ipsec-policy ipsec_policy;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Peer B

CLI Quick Configuration

To quickly configure VPN peer B to use OCSP, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [`edit`] hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/2 unit 0 family inet address 198.51.100.0/24
set interfaces lo0 unit 0 family inet address 172.17.1.100/24
set interfaces st0 unit 1 family inet address 172.18.1.1/24
set security pki ca-profile OCSP-ROOT ca-identity OCSP-ROOT
set security pki ca-profile OCSP-ROOT enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set security pki ca-profile OCSP-ROOT revocation-check ocsp url http://10.157.88.56:8210/OCSP-
ROOT/
set security pki ca-profile OCSP-ROOT revocation-check use-ocsp
set security pki ca-profile OCSP-ROOT revocation-check ocsp disable-responder-revocation-check
set security pki ca-profile OCSP-ROOT revocation-check ocsp connection-failure fallback-crl
```

```
set security pki ca-profile OCSP-ROOT revocation-check crl url http://10.1.1.1:8080/crl-as-der/
currentcrl-45.crlid=45
set security ike proposal ike_prop authentication-method rsa-signatures
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_policy mode aggressive
set security ike policy ike_policy proposals ike_prop
set security ike policy ike_policy certificate local-certificate localcert11
set security ike gateway jsr_gateway ike-policy ike_policy
set security ike gateway jsr_gateway address 192.0.2.50
set security ike gateway jsr_gateway local-identity hostname localcert11.example.net
set security ike gateway jsr_gateway external-interface ge-0/0/2.0
set security ike gateway jsr_gateway version v2-only
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec proposal ipsec_prop lifetime-seconds 1200
set security ipsec proposal ipsec_prop lifetime-kilobytes 150000
set security ipsec policy ipsec_policy perfect-forward-secrecy keys group2
set security ipsec policy ipsec_policy proposals ipsec_prop
set security ipsec vpn test_vpn bind-interface st0.1
set security ipsec vpn test_vpn ike gateway jsr_gateway
set security ipsec vpn test_vpn ike ipsec-policy ipsec_policy
set security ipsec vpn test_vpn establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure VPN peer B to use OCSP:

1. Configure interfaces.

```
[edit interfaces]
set ge-0/0/2 unit 0 family inet address 198.51.100.0/24
set lo0 unit 0 family inet address 172.17.1.100/24
set st0 unit 1 family inet address 172.18.1.1/24
```

2. Configure the CA profile.

```
[edit security pki ca-profile OCSP-ROOT]
set ca-identity OCSP-ROOT
set enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set revocation-check ocsp url http://10.157.88.56:8210/OCSP-ROOT/
set  revocation-check use-ocsp
set revocation-check ocsp disable-responder-revocation-check
set revocation-check ocsp connection-failure fallback-crl
set revocation-check crl url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike_prop]
set authentication-method rsa-signatures
set dh-group group2
set authentication-algorithm sha1
set encryption-algorithm 3des-cbc

[edit security ike policy ike_policy]
set mode aggressive
set proposals ike_prop
set certificate local-certificate localcert1

[edit security ike gateway jsr_gateway]
set ike-policy ike_policy
set address 192.0.2.50
set local-identity hostname localcert11.example.net
set external-interface ge-0/0/2.0
set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_prop]
set protocol esp
set authentication-algorithm hmac-sha1-96
set encryption-algorithm 3des-cbc
set lifetime-seconds 1200
set lifetime-kilobytes 150000
```

```
[edit security ipsec policy ipsec_policy]
set perfect-forward-secrecy keys group2
set proposals ipsec_prop

[edit security ipsec vpn test_vpn]
set bind-interface st0.1
set ike gateway jsr_gateway
set ike ipsec-policy ipsec_policy
set establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security pki ca-profile OCSP-ROOT`, `show security ike`, and `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/2 {
    unit 0 {
        family inet {
            address 198.51.100.0/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 172.17.1.100/24;
        }
    }
}
st0 {
    unit 1 {
        family inet {
            address 172.18.1.1/24;
        }
    }
}
[edit]
user@host# show security pki ca-profile OCSP-ROOT
```

```
ca-identity OCSP-ROOT;
enrollment {
    url http://10.1.1.1:8080/scep/OCSP-ROOT/;
}
revocation-check {
    crl {
        url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45;
    }
    ocsp {
        disable-responder-revocation-check;
        url http://10.157.88.56:8210/OCSP-ROOT/;
    }
    use-ocsp;
}
[edit]
user@host# show security ike
proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_policy {
    mode aggressive;
    proposals ike_prop;
    certificate {
        local-certificate localcert11;
    }
}
gateway jsr_gateway {
    ike-policy ike_policy;
    address 192.0.2.50;
    local-identity hostname localcert11.example.net;
    external-interface ge-0/0/2.0;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 1200;
```

```
    lifetime-kilobytes 150000;
}
policy ipsec_policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn test_vpn {
    bind-interface st0.1;
    ike {
        gateway jsr_gateway;
        ipsec-policy ipsec_policy;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying CA Certificates | **146**
- Verifying Local Certificates | **148**
- Verifying IKE Phase 1 Status | **150**
- Verifying IPsec Phase 2 Status | **151**

Confirm that the configuration is working properly.

Verifying CA Certificates

Purpose

Verify the validity of a CA certificate on each peer device.

Action

From operational mode, enter the `show security pki ca-certificate ca-profile OCSP-ROOT` or `show security pki ca-certificate ca-profile OCSP-ROOT detail` command.

```
user@host> show security pki ca-certificate ca-profile OCSP-ROOT
Certificate identifier: OCSP-ROOT
  Issued to: OCSP-ROOT, Issued by: C = US, O = example, CN = OCSP-ROOT
  Validity:
    Not before: 11-15-2013 22:26 UTC
    Not after: 11-14-2016 22:26 UTC
  Public key algorithm: rsaEncryption(2048 bits)

user@host> show security pki ca-certificate ca-profile OCSP-ROOT detail
Certificate identifier: OCSP-ROOT
  Certificate version: 3
  Serial number: 0000a17f
  Issuer:
    Organization: example, Country: US, Common name: OCSP-ROOT
  Subject:
    Organization: example, Country: US, Common name: OCSP-ROOT
  Subject string:
    C=US, O=example, CN=OCSP-ROOT
  Validity:
    Not before: 11-15-2013 22:26 UTC
    Not after: 11-14-2016 22:26 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:c6:38:e9:03:69:5e:45:d8:a3:ea:3d
    2e:e3:b8:3f:f0:5b:39:f0:b7:35:64:ed:60:a0:ba:89:28:63:29:e7
    27:82:47:c4:f6:41:53:c8:97:d7:1e:3c:ca:f0:a0:b9:09:0e:3d:f8
    76:5b:10:6f:b5:f8:ef:c5:e8:48:b9:fe:46:a3:c6:ba:b5:05:de:2d
    91:ce:20:12:8f:55:3c:a6:a4:99:bb:91:cf:05:5c:89:d3:a7:dc:a4
    d1:46:f2:dc:36:f3:f0:b5:fd:1d:18:f2:e6:33:d3:38:bb:44:8a:19
    ad:e0:b1:1a:15:c3:56:07:f9:2d:f6:19:f7:cd:80:cf:61:de:58:b8
    a3:f5:e0:d1:a3:3a:19:99:80:b0:63:03:1f:25:05:cc:b2:0c:cd:18
    ef:37:37:46:91:20:04:bc:a3:4a:44:a9:85:3b:50:33:76:45:d9:ba
    26:3a:3b:0d:ff:82:40:36:64:4e:ea:6a:d8:9b:06:ff:3f:e2:c4:a6
    76:ee:8b:58:56:a6:09:d3:4e:08:b0:64:60:75:f3:e2:06:91:64:73
    d2:78:e9:7a:cb:8c:57:0e:d1:9a:6d:3a:4a:9e:5b:d9:e4:a2:ef:31
    5d:2b:2b:53:ab:a1:ad:45:49:fd:a5:e0:8b:4e:0b:71:52:ca:6b:fa
    8b:0e:2c:7c:7b:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
```

```
  Distribution CRL:
    http://10.1.1.1:8080/crl-as-der/currentcrl-45.crl?id=45
  Authority Information Access OCSP:
    http://10.1.1.1:8090/OCSP-ROOT/
  Use for key: CRL signing, Certificate signing, Key encipherment, Digital signature
  Fingerprint:
    ed:ce:ec:13:1a:d2:ab:0a:76:e5:26:6d:2c:29:5d:49:90:57:f9:41 (sha1)
    af:87:07:69:f0:3e:f7:c6:b8:2c:f8:df:0b:ae:b0:28 (md5)
```

In this example, IP addresses are used in the URLs in the CA profile configuration. If IP addresses are not used with CA-issued certificates or CA certificates, DNS must be configured in the device's configuration. DNS must be able to resolve the host in the distribution CRL and in the CA URL in the CA profile configuration. Additionally, you must have network reachability to the same host to receive revocation checks.

Meaning

The output shows the details and validity of CA certificate on each peer as follows:

- `C`—Country.
- `O`—Organization.
- `CN`—Common name.
- `Not before`—Begin date of validity.
- `Not after`—End date of validity.

Verifying Local Certificates

Purpose

Verify the validity of a local certificate on each peer device.

Action

From operational mode, enter the `show security pki local-certificate certificate-id localcert1 detail` command.

```
user@host> show security pki local-certificate certificate-id localcert1 detail
Certificate identifier: localcert1
  Certificate version: 3
```

```
  Serial number: 013e3f1d
  Issuer:
    Organization: example, Country: US, Common name: OCSP-ROOT
  Subject:
    Organization: example, Organizational unit: example, State: california1, Locality:
sunnyvale1, Common name: localcert1, Domain component: domain_component1
  Subject string:
    DC=domain_component1, CN=localcert1, OU=example, O=example, L=sunnyvale1, ST=california1,
C=us1
  Alternate subject: "localcert1@example.net", localcert1.example.net, 10.10.1.50
  Validity:
    Not before: 01-28-2014 22:23 UTC
    Not after: 03-29-2014 22:53 UTC
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:a6:df:c1:57:59:f8:4d:0f:c4:a8:96:25:97
    03:c4:a0:fb:df:d5:f3:d5:56:b6:5a:26:65:b8:1a:ec:be:f6:c6:5f
    b3:d7:d3:59:39:48:52:4a:e3:1b:e4:e0:6d:24:c3:c1:50:8c:55:3b
    c0:c1:29:a0:45:29:8e:ec:3e:52:2f:84:b3:e8:89:9a:0f:8b:7d:e8
    90:4b:c1:28:48:95:b3:aa:11:ab:b4:8c:a8:80:ce:90:07:2a:13:a2
    2f:84:44:92:3b:be:7d:39:5b:2f:9a:4c:7a:2f:2d:31:8b:12:6d:52
    34:7d:6b:e4:69:7e:f3:86:55:e2:89:31:98:c9:15:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://10.1.1.1:8080/crl-as-der/currentcrl-45.crl?id=45
  Authority Information Access OCSP:
    http://10.1.1.1/:8090/OCSP-ROOT/
  Fingerprint:
    00:c6:56:64:ad:e3:ce:8e:26:6b:df:17:1e:de:fc:14:a4:bb:8c:e4 (sha1)
    7f:43:c6:ed:e4:b3:7a:4f:9a:8c:0b:61:95:01:c9:52 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Meaning

The output shows the details and validity of a local certificate on each peer as follows:

- `DC`—Domain component.
- `CN`—Common name.
- `OU`—Organizational unit.

- O—Organization.
- L—Locality
- ST—State.
- C—Country.
- Not before—Begin date of validity.
- Not after—End date of validity.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status on each peer device.

Action

From operational mode, enter the show security ike security-associations command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
6534660 UP     3e62e05abd6a703f  c552b238e8a26668  IKEv2          198.51.100.50
```

From operational mode, enter the show security ike security-associations detail command.

```
user@host> show security ike security-associations detail
IKE peer 198.51.100.50, Index 6534660, Gateway Name: jsr_gateway
  Role: Responder, State: UP
  Initiator cookie: 3e62e05abd6a703f, Responder cookie: c552b238e8a26668
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.0.2.50:500, Remote: 198.51.100.50:500
  Lifetime: Expires in 26906 seconds
  Peer ike-id: localcert11.example.net
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
```

```
  Traffic statistics:
   Input  bytes  :                2152
   Output bytes  :                2097
   Input  packets:                   4
   Output packets:                   4
  Flags: IKE SA is created
  IPSec security associations: 4 created, 0 deleted
  Phase 2 negotiations in progress: 0

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.0.2.50:500, Remote: 198.51.100.50:500
    Local identity: 192.0.2.50
    Remote identity: localcert11.example.net
    Flags: IKE SA is created
```

Meaning

The `flags` field in the output shows that, IKE security association is created.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status on each peer device.

Action

From operational mode, enter the `show security ipsec security-associations` command.

```
user@host> show security ipsec security-associations
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:3des/sha1 9d1066e2 252/   150000 -   root 500   198.51.100.50
  >131073 ESP:3des/sha1 82079c2c 252/   150000 -   root 500   198.51.100.50
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@host> show security ipsec security-associations detail
  ID: 131073 Virtual-system: root, VPN Name: test_vpn
  Local Gateway: 192.0.2.50, Remote Gateway: 198.51.100.50
```

```
Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Version: IKEv2
  DF-bit: clear
  Bind-interface: st0.1

Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
Last Tunnel Down Reason: Delete payload received
  Direction: inbound, SPI: 9d1066e2, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 249 seconds
  Lifesize Remaining:  150000 kilobytes
  Soft lifetime: Expires in 10 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
  Anti-replay service: counter-based enabled, Replay window size: 64

  Direction: outbound, SPI: 82079c2c, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 249 seconds
  Lifesize Remaining:  150000 kilobytes
  Soft lifetime: Expires in 10 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
  Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output shows the ipsec security associations details.

RELATED DOCUMENTATION

- Basic Elements of PKI in Junos OS

IPv6 IPsec VPNs

SUMMARY

Read this topic to learn about IPv6 IPsec VPNs.

IN THIS SECTION

- VPN Feature Support for IPv6 Addresses | **153**
- Understanding IPv6 IKE and IPsec Packet Processing | **159**
- IPv6 IPsec Configuration Overview | **166**
- Example: Configuring an IPv6 IPsec Manual VPN | **166**
- Platform-Specific IPv6 Tunnels Behavior | **170**
- Additional Platform Information | **171**

Juniper Networks supports manual and autokey IKE with preshared keys configurations for IPv6 IPsec VPN.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific IPv6 Tunnels Behavior" on page 170 section for notes related to your platform.

See the "Additional Platform Information" on page 171 section for more information.

VPN Feature Support for IPv6 Addresses

A route-based site-to-site VPN tunnel with a point-to-point secure tunnel interface can operate in IPv4-in-IPv4, IPv6-in-IPv6, IPv6-in-IPv4, or IPv4-in-IPv6 tunnel modes. IPv6 addresses can be in the outer IP header, which represents the tunnel endpoint, or in the inner IP header, which represents the final source and destination addresses for a packet.

Table 28 on page 154 defines the support for IPv6 addresses in VPN features.

Table 28: IPv6 Address Support in VPN Features

Feature	Supported	Exceptions
IKE and IPsec Support:		
IKEv1 and IKEv2	Yes	Unless specified, all supported features are applicable for IKEv1 and IKEv2.
Route-based VPN	Yes	–
Site-to-site VPN	Yes	Only one-to-one, site-to-site VPN is supported. Many-to-one, site-to-site VPN (NHTB) is not supported. NHTB configuration cannot be committed for tunnel modes other than IPv4-in-IPv4 tunnels.
Dynamic endpoint VPN	Yes	–
Dialup VPN	Yes	–
AutoVPN	Yes	AutoVPN networks that use secure tunnel interfaces in point-to-point mode support IPv6 addresses for traffic selectors and for IKE peers. AutoVPN in point-to-multipoint mode supports IPv6 traffic.
Group VPN	No	–
Point-to-point tunnel interfaces	Yes	–
Point-to-multipoint tunnel interfaces	Yes	–
Hub-and-spoke scenario for site-to-site VPNs	Yes	–

Table 28: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
Numbered and unnumbered tunnel interfaces	Yes	–
Unicast static and dynamic (RIP, OSPF, BGP) routing	Yes	–
Multicast dynamic routing (PIM)	No	–
Virtual router	Yes	–
Logical system	No	–
Automatic and manual SA and key management	Yes	–
Multiple SPUs	Yes	–
Chassis cluster	Yes	-
Statistics, logs, per-tunnel debugging	Yes	–
SNMP MIB	Yes	–
Local address selection	Yes	When multiple addresses in the same address family are configured on a physical external interface to a VPN peer, we recommend that you also configure `local-address` at the [`edit security ike gateway` *`gateway-name`*] hierarchy level.
Loopback address termination	Yes	–
Xauth or modecfg over IPv6	No	–

Table 28: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
SPC insert	Yes	–
ISSU	Yes	–
DNS name as IKE gateway address	Yes	As with IPv4 tunnels, peer gateway address changes in the DNS name are not supported with IPv6 tunnels.
Preshared key or certificate authentication	Yes	–
NAT-Traversal (NAT-T) for IPv4 IKE peers	Yes	NAT-T is supported only for IPv6-in-IPv4 and IPv4-in-IPv4 tunnel modes with IKEv1. IPv6-in-IPv6 and IPv4-in-IPv6 tunnel modes are not supported. IKEv2 is not supported for NAT-T. NAT-T from IPv6 to IPv4 or from IPv4 to IPv6 is not supported.
Dead peer detection (DPD) and DPD gateway failover	Yes	DPD gateway failover is only supported for different gateway addresses within the same family. Failover from an IPv6 gateway address to an IPv4 gateway address, or vice versa, is not supported.
Encryption sets, authentication algorithms, and DH groups	Yes	–
Generic proposals and policies for IPv6 and IPv4	Yes	–
General IKE ID	Yes	–
ESP and AH transport modes	No	These modes are not supported for IPv4.
ESP and AH tunnel modes	Yes	AH tunnel mode with mutable extension headers and options is not supported.

Table 28: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
Extended sequence number	No	–
Single proxy ID pairs	Yes	–
Multiple traffic selector pairs	Yes	Supported with IKEv1 only.
Lifetime of IKE or IPsec SA, in seconds	Yes	–
Lifetime of IKE SA, in kilobytes	Yes	–
VPN monitoring	No	Configuration with IPv6 tunnels cannot be committed.
DF bit	Yes	For IPv6-in-IPv6 tunnels, the DF bit is set only if configured at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. `df-bit clear` is the default.
Dual-stack (parallel IPv4 and IPv6 tunnels) over a single physical interface	Yes	For route-based site-to-site VPNs. A single IPv4 tunnel can operate in both IPv4-in-IPv4 and IPv6-in-IPv4 tunnel modes and a single IPv6 tunnel can operate in both IPv4-in-IPv6 and IPv6-in-IPv6 tunnel modes.
IPv6 extension headers	Yes	IPv6 extension headers and IPv4 options for IKE and IPsec packets are accepted but are not processed. AH with mutable EHs and options is not supported.
Fragmentation and reassembly	Yes	–
VPN session affinity	Yes	–
Multicast traffic	No	–

Table 28: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
Tunnel IP services (Screen, NAT, ALG, IPS, AppSecure)	Yes	–
Packet reordering for IPv6 fragments over tunnel	No	–
Bidirectional Forwarding Detection (BFD) over OSPFv3 routes on st0 interface	No	–
Neighbor Discovery Protocol (NDP) over st0 interfaces	No	–
PKI Support:		
PKI in virtual router	Yes	–
RSA signature authentication (512-, 1024-, 2048-, or 4096-bit key size)	Yes	–
DSA signature authentication (1024-, 2048-, or 4096-bit key size)	Yes	–
ECDSA signatures	Yes	–
Certificate chain authentication	No	–
Automatic or manual enrollment over IPv4	Yes	–
Automatic or manual revocation over IPv4	Yes	–
Automatic or manual enrollment over IPv6	No	–

Table 28: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
Automatic or manual revocation over IPv6	No	–
IPv6 addresses within PKI certificate fields	No	–

SEE ALSO

Understanding VPN Tunnel Modes | 422

IPsec Overview | 12

Understanding IPv6 IKE and IPsec Packet Processing

IN THIS SECTION

IPv6 IKE Packet Processing | 159

IPv6 IPsec Packet Processing | 161

This topic includes the following sections:

IPv6 IKE Packet Processing

Internet Key Exchange (IKE) is part of the IPsec suite of protocols. It automatically enables two tunnel endpoints to set up security associations (SAs) and negotiate secret keys with each other. There is no need to manually configure the security parameters. IKE also provides authentication for communicating peers.

IKE packet processing in IPv6 networks involves the following elements:

- Internet Security Association and Key Management Protocol (ISAKMP) Identification Payload

ISAKMP identification payload is used to identify and authenticate the communicating IPv6 peers. Two ID types (ID_IPV6_ADDR and ID_IPV6_ADDR_SUBNET) are enabled for IPv6. The ID type indicates the type of identification to be used. The ID_IPV6_ADDR type specifies a single 16-octet IPv6 address. This ID type represents an IPv6 address. The ID_IPV6_ADDR_SUBNET type specifies a range of IPv6 addresses represented by two 16-octet values. This ID type represents an IPv6 network mask. Table 29 on page 160 lists the ID types and their assigned values in the identification payload.

Table 29: ISAKMP ID Types and Their Values

ID Type	Value
RESERVED	0
ID_IPV4_ADDR	1
ID_FQDN	2
ID_USER_FQDN	3
ID_IPV4_ADDR_SUBNET	4
ID_IPV6_ADDR	5
ID_IPV6_ADDR_SUBNET	6
ID_IPV4_ADDR_RANGE	7
ID_IPV6_ADDR_RANGE	8
ID_DER_ASN1_DN	9
ID_DER_ASN1_GN	10
ID_KEY_ID	11

Table 29: ISAKMP ID Types and Their Values *(Continued)*

ID Type	Value
ID_LIST	12

The ID_IPV6_ADDR_RANGE type specifies a range of IPv6 addresses represented by two 16-octet values. The first octet value represents the starting IPv6 address and the second octet value represents the ending IPv6 address in the range. All IPv6 addresses falling between the first and last IPv6 addresses are considered to be part of the list.

Two ID types in ISAKMP identification payload (ID_IPV6_ADDR_RANGE and ID_IPV4_ADDR_RANGE) are not supported in this release.

- Proxy ID

  A proxy ID is used during Phase 2 of IKE negotiation. It is generated before an IPsec tunnel is established. A proxy ID identifies the SA to be used for the VPN. Two proxy IDs are generated—local and remote. The local proxy ID refers to the local IPv4 or IPv6 address/network and subnet mask. The remote proxy ID refers to the remote IPv4 or IPv6 address/network and subnet mask.

- Security Association

  An SA is an agreement between VPN participants to support secure communication. SAs are differentiated based on three parameters—security parameter index (SPI), destination IPv6 address, and security protocol (either AH or ESP). The SPI is a unique value assigned to an SA to help identify an SA among multiple SAs. In an IPv6 packet, the SA is identified from the destination address in the outer IPv6 header and the security protocol is identified from either the AH or the ESP header.

IPv6 IPsec Packet Processing

After IKE negotiations are completed and the two IKE gateways have established Phase 1 and Phase 2 SAs, IPv6 IPsec employs authentication and encryption technologies to secure the IPv6 packets. Because IPv6 addresses are 128 bits long compared to IPv4 addresses, which are 32-bits long, IPv6 IPsec packet processing requires more resources.

Packet reordering for IPv6 fragments over a tunnel is not supported.

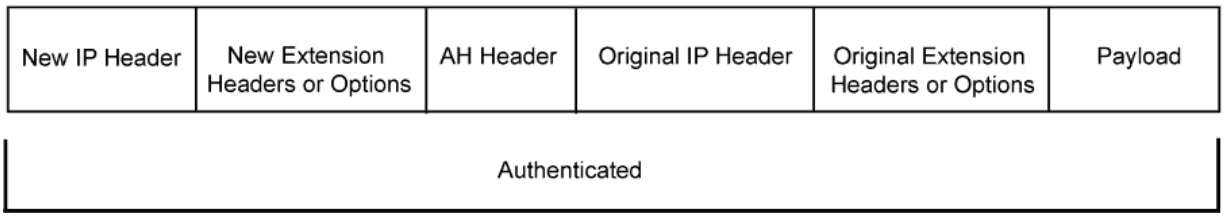
Devices with IPv6 addressing do not perform fragmentation. IPv6 hosts should either perform path MTU discovery or send packets smaller than the IPv6 minimum MTU size of 1280 bytes.

This topic includes the following sections:

AH Protocol in IPv6

The AH protocol provides data integrity and data authentication for IPv6 packets. IPv6 IPsec uses extension headers (for example, hop-by-hop and routing options) that must be arranged in a particular way in the IPv6 datagram. In AH tunnel mode, the AH header immediately follows the new outer IPv6 header similar to that in IPv4 AH tunnel mode. The extension headers are placed after the original inner header. Therefore, in AH tunnel mode, the entire packet is encapsulated by adding a new outer IPv6 header, followed by an authentication header, an inner header, extension headers, and the rest of the original datagram as shown in Figure 16 on page 162.

Figure 16: IPv6 AH Tunnel Mode



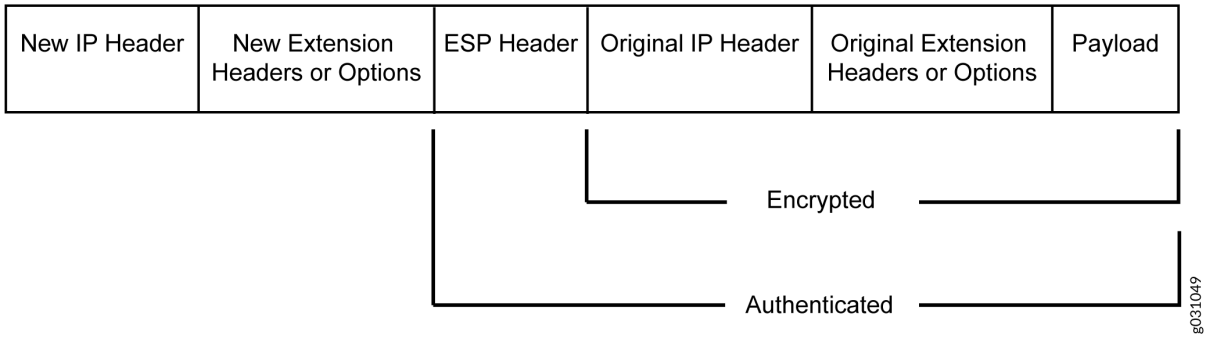
Unlike ESP, the AH authentication algorithm covers the outer header as well as any new extension headers and options.

AH tunnel mode on firewalls does not support IPv4 mutable options or IPv6 mutable extension headers.

ESP Protocol in IPv6

ESP protocol provides both encryption and authentication for IPv6 packets. Because IPv6 IPsec uses extension headers (for example, hop-by-hop and routing options) in the IPv6 datagram, the most important difference between IPv6 ESP tunnel mode and IPv4 ESP tunnel mode is the placement of extension headers in the packet layout. In ESP tunnel mode, the ESP header immediately follows the new outer IPv6 header similar to that in IPv4 ESP tunnel mode. Therefore, in ESP tunnel mode, the entire packet is encapsulated by adding a new outer IPv6 header, followed by an ESP header, an inner header, extension headers, and the rest of the original datagram as shown in Figure 17 on page 163.

Figure 17: IPv6 ESP Tunnel Mode



IPv4 Options and IPv6 Extension Headers with AH and ESP

IPsec packets with IPv4 options or IPv6 extension headers can be received for decapsulation on firewalls.

See the "Additional Platform Information" on page 171 section for more information.

The table shows the IPv4 options or IPv6 extension headers that are supported with the ESP or AH protocol on firewalls. If an unsupported IPsec packet is received, ICV calculation fails and the packet is dropped.

Integrity Check Value Calculation in IPv6

The AH protocol verifies the integrity of the IPv6 packet by computing an Integrity Check Value (ICV) on the packet contents. ICV is usually built over an authentication algorithm such as MD5 or SHA-1. The IPv6 ICV calculations differ from that in IPv4 in terms of two header fields—mutable header and optional extension header.

You can calculate the AH ICV over the IPv6 header fields that are either immutable in transit or predictable in value upon arrival at the tunnel endpoints. You can also calculate the AH ICV over the AH header and the upper level protocol data (considered to be immutable in transit). You can calculate the ESP ICV over the entire IPv6 packet, excluding the new outer IPv6 header and the optional extension headers.

Unlike IPv4, IPv6 has a method for tagging options as mutable in transit. IPv6 optional extension headers contain a flag that indicates mutability. This flag determines the appropriate processing.

IPv4 mutable options and IPv6 extension headers are not supported with the AH protocol.

Header Construction in Tunnel Modes

In tunnel mode, the source and destination addresses of the outer IPv4 or IPv6 header represent the tunnel endpoints, while the source and destination addresses of the inner IPv4 or IPv6 header represent the final source and destination addresses. Table 30 on page 164 summarizes how the outer IPv6 header relates to the inner IPv6 or IPv4 header for IPv6-in-IPv6 or IPv4-in-IPv6 tunnel modes. In outer header fields, “Constructed” means that the value of the outer header field is constructed independently of the value in the inner header field.

Table 30: IPv6 Header Construction for IPv6-in-IPv6 and IPv4-in-IPv6 Tunnel Modes

Header Fields	Outer Header at Encapsulator	Inner Header at Decapsulator
version	6.	No change.
DS field	Copied from the inner header.	No change.
ECN field	Copied from the inner header.	Constructed.
flow label	0.	No change.
payload length	Constructed.	No change.
next header	AH, ESP, and routing header.	No change.
hop limit	64.	Decrement.
src address	Constructed.	No change.
dest address	Constructed.	No change.
Extension headers	Never copied.	No change.

Table 31 on page 165 summarizes how the outer IPv4 header relates to the inner IPv6 or IPv4 header for IPv6-in-IPv4 or IPv4-in-IPv4 tunnel modes. In outer header fields, “Constructed” means that the value of the outer header field is constructed independently of the value in the inner header field.

Table 31: IPv4 Header Construction for IPv6-in-IPv4 and IPv4-in-IPv4 Tunnel Modes

Header Fields	Outer Header	Inner Header
version	4.	No change.
header length	Constructed.	No change.
DS field	Copied from the inner header.	No change.
ECN field	Copied from the inner header.	Constructed.
total length	Constructed.	No change.
ID	Constructed.	No change.
flags (DF, MF)	Constructed.	No change.
fragment offset	Constructed.	No change.
TTL	64.	Decrement.
protocol	AH, ESP	No change.
checksum	Constructed.	Constructed.
src address	Constructed.	No change.
dest address	Constructed.	No change.
options	Never copied.	No change.

For IPv6-in-IPv4 tunnel mode, the Don't Fragment (DF) bit is cleared by default. If the `df-bit set` or `df-bit copy` options are configured at the [`edit security ipsec vpn` *vpn-name*] hierarchy level for the corresponding IPv4 VPN, the DF bit is set in the outer IPv4 header.

For IPv4-in-IPv4 tunnel mode, the DF bit in the outer IPv4 header is based on the `df-bit` option configured for the inner IPv4 header. If `df-bit` is not configured for the inner IPv4 header, the DF bit is cleared in the outer IPv4 header.

SEE ALSO

IPsec Overview | 12

IPv6 IPsec Configuration Overview | 166

IPv6 IPsec Configuration Overview

Juniper Networks supports manual and autokey IKE with preshared keys configurations for IPv6 IPsec VPN.

- AutoKey IKE VPN—In an autoKey IKE VPN configuration, the secret keys and SAs are automatically created using the autoKey IKE mechanism. To set up an IPv6 autoKey IKE VPN, two phases of negotiations are required—Phase 1 and Phase 2.
  - Phase 1—In this phase, the participants establish a secure channel for negotiating the IPsec SAs.
  - Phase 2—In this phase, the participants negotiate the IPsec SAs for authenticating and encrypting the IPv6 data packets.

  For more information on Phase 1 and Phase 2 negotiations, see "Internet Key Exchange" on page 2

SEE ALSO

IPsec VPN with Autokey IKE Configuration Overview | 110

Example: Configure an IPv6 address as the Source Address for a CA Profile

Example: Configuring an IPv6 IPsec Manual VPN

IN THIS SECTION

Requirements | 167

- Overview | **167**
- Configuration | **167**
- Verification | **170**

This example shows how to configure an IPv6 IPsec manual VPN.

Requirements

Before you begin:

- Understand how VPNs work. See "IPsec Overview" on page 12.
- Understand IPv6 IPsec packet processing. See "Understanding IPv6 IKE and IPsec Packet Processing" on page 159.

Overview

In a Manual VPN configuration, the secret keys are manually configured on the two IPsec endpoints.

In this example, you:

- Configure the authentication parameters for a VPN named vpn-sunnyvale.
- Configure the encryption parameters for vpn-sunnyvale.
- Specify the outgoing interface for the SA.
- Specify the IPv6 address of the peer.
- Define the IPsec protocol. Select the ESP protocol because the configuration includes both authentication and encryption.
- Configure a security parameter index (SPI).

Configuration

IN THIS SECTION

- Procedure | **168**

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec vpn vpn-sunnyvale manual authentication algorithm hmac-md5-96 key ascii-text
“$ABC123”
set security ipsec vpn vpn-sunnyvale manual encryption algorithm 3des-cbc key ascii-text
“$ABC123”
set security ipsec vpn vpn-sunnyvale manual external-interface ge-0/0/14.0
set security ipsec vpn vpn-sunnyvale manual gateway 2001:db8:1212::1112
set security ipsec vpn vpn-sunnyvale manual protocol esp
set security ipsec vpn vpn-sunnyvale manual spi 12435
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security algorithms:

1. Configure the authentication parameters.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set authentication algorithm hmac-md5-96 key ascii-text “$ABC123”
```

2. Configure the encryption parameters.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set encryption algorithm 3des-cbc key ascii-text “$ABC123”
```

3. Specify the outgoing interface for the SA.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set external-interface ge-0/0/14.0
```

4. Specify the IPv6 address of the peer.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set gateway 2001:db8:1212::1112
```

5. Define the IPsec protocol.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set protocol esp
```

6. Configure an SPI.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set spi 12435
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec vpn vpn-sunnyvale` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
[user@host]show security ipsec vpn vpn-sunnyvale
manual {
gateway 2001:db8:1212::1112 ;
external-interface ge-0/0/14.0 ;
protocol esp ;
spi 12435 ;
authentication {
    algorithm hmac-md5-96 ;
    key ascii-text $ABC123” ;## SECRET DATA
}
    encryption {
        algorithm 3des-cbc ;
        key ascii-text $ABC123”; ## SECRET DATA
        }
    }
```

Verification

IN THIS SECTION

- Verifying Security Algorithms | 170

To confirm that the configuration is working properly, perform this task:

Verifying Security Algorithms

Purpose

Determine if security algorithms are applied or not.

Action

From operational mode, enter the `show security ipsec security-associations` command.

SEE ALSO

IPv6 IPsec Configuration Overview | 166

Platform-Specific IPv6 Tunnels Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behavior for your platform.

Table 32: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX300, SRX320, SRX340, and SRX345 devices that support IPv6 IPsec VPNs, you can configure: • Route-based IPv6 IPsec VPN tunnels in active/active mode in Chassis Cluster configuration. • On SRX5400, SRX5600, and SRX5800 devices that support IPv6 IPsec VPNs, you cannot configure route-based IPv6 IPsec VPN tunnels in active/active mode in Chassis Cluster configuration.

Additional Platform Information

Use Feature Explorer to confirm platform and release support for specific features. Additional Platforms may be supported.

Table 33: Additional Platform Information for IPv4 Options or IPv6 Extension Headers

Options or Extension Headers	**SRX300 SRX320 SRX340 SRX345**	SRX400 SRX440 SRX440-2AC	**SRX5400 SRX5600 SRX5800**
ESP with IPv4 options	Supported	Supported	Supported
ESP with IPv6 extension headers	Supported	Not supported	Supported
AH with IPv4 immutable options	Supported	Supported	Supported
AH with IPv6 immutable extension headers	Supported	Not supported	Supported
AH with IPv4 mutable options	Not supported	Not supported	Not supported
AH with IPv6 mutable extension headers	Not supported	Not supported	Not supported

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
21.4R1	Support for IPv6 address in point-to-multipoint mode for IPsec VPN service with the iked process is available from Junos OS Release 21.4R1.
18.1R1	Support for IPv6 address in point-to-multipoint mode for IPsec VPN service with the kmd process is available from Junos OS Release 18.1R1.

Quantum Safe IPsec VPN

SUMMARY

Learn how to use and configure the out-of-band key retrieval mechanisms in the iked process to negotiate with quantum secured IKE and IPsec SAs.

IN THIS SECTION

- Quantum Security Overview | **173**
- Junos Key Manager Overview | **173**
- Use Key Profile for Quantum Safe IPsec VPN | **174**
- Quantum Key Distribution | **174**
- Configure Static Key Profile for Junos Key Manager | **176**
- Example: Configure Static Keys Profile for Site-To-Site VPN | **178**
- Example: Configure Static Keys Profile for AutoVPN | **203**
- Configure Quantum Key Manager Key Profile for Junos Key Manager | **243**
- Example: Configure Quantum Key Manager Key Profile for Site-To-Site IPsec VPN | **247**
- Example: Configure Quantum-Secured IPsec AutoVPN Topology Using Quantum Key Manager Key Profile | **274**

Quantum Security Overview

The IPsec communication channel relies on the Internet Key Exchange (IKE) protocol. The IKE maintains security parameters to protect the data traffic. The security parameters include encryption and authentication algorithms, and associated keys.

The security protocols rely on asymmetric cryptographic algorithms such as Diffie Hellman (DH) or Elliptic Curve Diffie Hellman (ECDH) to establish keys are vulnerable to attacks.

To avoid security attacks, the RFC8784 introduces a method out-of-band method. The out-of-band method adds a secret key at the initiator and the responder. The secret key is Post-quantum Pre-shared Key (PPK).

- You can use the PPK in addition to the authentication method in IKEv2.
- PPK provides quantum resistance to any child SAs in initial negotiated IPsec SAs and any subsequent re-iked IPsec SAs.
- With PPK and peer authentication key, initiator and responder can detect key mismatch.

Junos Key Manager Overview

You can use Junos Key Manager (JKM) to configure the static keys or dynamics keys to protect the data plane and control plane.

The JKM process acts as a key store and a proxy between the client or crypto application. The client or crypto application requires a key to establish an encrypted and authenticated quantum safe session with peer or application. The quantum safe uses the out-of-band key retrieval mechanism that lets two peers have the key. Different out-of-band mechanisms will have different protocols or methods to communicate. The JKM provides a common uniform interface for client or crypto applications to communicate.

Key Retrieval Mechanism

Two out-of-band key retrieval mechanisms in the iked process to negotiate with quantum secured IKE and IPsec SAs.

- **Static Key**—With static key profiles, you can configure a static key ID and a corresponding key. The same static key ID and key gets generated every time a request to JKM over a static key profile.
- **Quantum Key Manager**—With quantum key manager key profiles, you can access the Quantum Key Distribution (QKD) devices and Quantum Network. The Quantum Network generates and exchange quantum keys between peers. Generates a different key ID and key every time on request to JKM over a quantum key manager key profile.

Use Key Profile for Quantum Safe IPsec VPN

With static key profiles, you can configure a static key ID and a corresponding key. To establish the quantum safe IPsec SAs, use the static key profile as Post-Quantum Pre-Shared Key (PPK) profile in the IPsec-VPN configuration. Uses the same key and key ID to re-authenticate existing IKE SA.

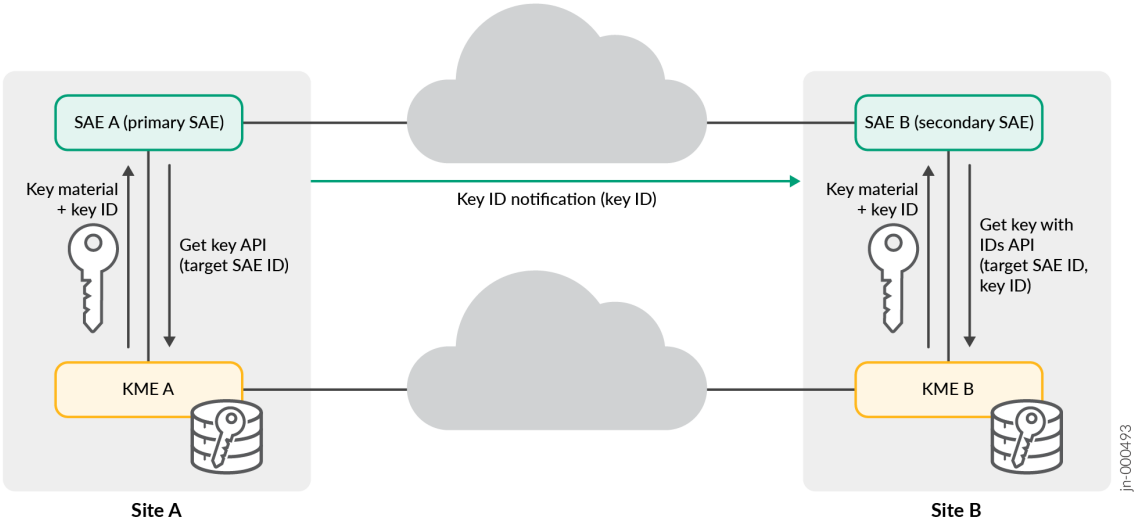
With quantum key manager key profile profiles, to access the Quantum Networks you need access to the QKD devices. The Quantum Network generates and exchanges quantum keys between peers. You can configure all the necessary parameters such as local SAE ID, URL to the QKD device, and so on. To establish IPsec SAs, use the quantum key manager key profile as Post-Quantum Pre-Shared Key (PPK) profile in the IPsec VPN configuration. Uses a different key and key ID to re-authenticate existing IKE SA.

Quantum Key Distribution

Quantum key distribution (QKD) is a secure key distribution method that uses quantum. Networks use quantum channels for generating the same key at both ends and monitor the quantum channel between the peers. These keys are dynamic, protects the data plane, and control plane.

Key Management Entity (KME) is the term we use to refer to the QKD devices on the management or control layer. QKD devices connect to each other through their quantum or QKD network. The KMEs connects over the public network through the secure channels for exchanging any control messages. The applications, Secure Application Entity (SAEs), and devices interact with KMEs through the secure channels as per ETSI specification. HTTPS combines with mutual TLS authentication and enables secure operations over the QKD network.

Figure 18: Two Devices Interacting with Their Corresponding QKD Devices to Establish a Quantum Secured Session



In the Figure 18 on page 175 describes how the two devices interacting with their corresponding QKD devices to establish a quantum secured session

- SAE A role is primary. SAE A acts as the initiator to establish a quantum secured session with SAE B.
- The SAE B role is secondary. SAE B acts as the responder.
- The SAE A request the KME A through the Get key API to generate and share a new quantum key with SAE B with target SAE ID.
- The KME A performs the operation and responds to SAE A with the generated key ID and key material.
- KME B receives the key material and the generated ID key over the QKD network.
- The SAE A initiates secured session with SAE B directly using the same key and key ID.
- An exchange of messages establishes a secure session with SAE B.
- SAE A sends the key ID in plaintext or encrypted for the corresponding quantum key that is used to secure the session with SAE B.
- Once SAE B receives the key ID, the SAE B contacts KME B through the Get key with IDs API to get the corresponding quantum-key for the given key ID and target SAE ID or SAE A.
- After SAE B gets the key, a fully quantum secured session establishes between SAE A and SAE B.

Configure Static Key Profile for Junos Key Manager

IN THIS SECTION

- Requirements | 176
- Overview | 176
- Configuration | 176
- Verification | 177

This example shows how to configure static key profile for Junos key manager. Configure the static keys on concerned gateways and do not need share static keys over the Internet to establish the IPsec tunnel.

Requirements

Use Feature Explorer to confirm platform and release support for specific features.

1. Hardware requirements —Device that supports the feature.
2. Software requirements—Operating system with **Junos IKE** and **Junos Key Manager** packages that supports the feature.

Overview

With static key based profiles you need to configure a static key ID and a corresponding key. If you use the static key profile in the IPsec VPN object, when the re-authentication for existing IKE SA the same key and key ID are used.

Configuration

Configure the static key profile for Junos key manager.

```
user@host# set security key-manager profiles km_profile_1 static key-id ascii-text test-ppk-id
user@host# set security key-manager profiles km_profile_1 static key ascii-text
qjwbdip139u5mcy89m28pcgowerefnkjsdg
```

Verification

Purpose

Verify the static key profile and keys.

Action

From operational mode, enter the `request security key-manager profiles get profile-keys name km_profile_1` to view the static key profile and keys.

```
user@host> request security key-manager profiles get profile-keys name km_profile_1

  - Response:
    - Status: SUCCESS
    - Name: km_profile_1
    - Type: Static
    - Key-size: 280 bits
    - Key-count: 1
    - Key-ids:
      - test-ppk-id
    - Keys:
      - 716a77626469703133397535 6d637938396d32387063676f77657265666e6b6a736467
```

From operational mode, enter the `show security key-manager profiles name km_profile_1 detail` to view the static key profile details.

```
user@host> show security key-manager profiles name km_profile_1 detail

Name: km_profile_1, Index: 1, Type: Static
  Configured-at: 10.09.23 (20:16:34)
  Time-elapsed: 0 hrs 2 mins 21 secs
  Request stats:
    Received: 1
    In-progress: 0
    Success: 1
    Failed: 0
```

Meaning

The `request security key-manager profiles get profile-keys name km_profile_1` displays the status, static key profile name, type, key size, key ID, and keys.

The `show security key-manager profiles name km_profile_1 detail` displays the static key profile name, type, and request status.

Example: Configure Static Keys Profile for Site-To-Site VPN

SUMMARY

Use this configuration example to configure the static key profile. You can use the static key profile to secure an IPsec Site-to-Site VPN infrastructure.

IN THIS SECTION

- Example Prerequisites | **179**
- Before You Begin | **179**
- Functional Overview | **180**
- Topology Overview | **183**
- Topology Illustration | **184**
- Step-By-Step Configuration on Firewall Devices | **184**
- Verification | **187**
- Appendix 1: Set Commands on all Devices | **192**
- Appendix 2: Show Configuration Output on DUT | **195**

You can secure an IPsec Site-to-Site VPN infrastructure by configuring the static key profile.

In this configuration example, the SRX1 and SRX2 devices use the static key profile to fetch the QKD keys on IPsec VPN. The QKD keys help to send traffic securely over the Internet.

TIP:

Table 34: Estimated Timers

Reading Time	Less than an hour

Configuration Time	Less than an hour

Example Prerequisites

Use Feature Explorer to confirm platform and release support for specific features.

Table 35: Requirements

Hardware requirements	Devices that support the feature.
Software requirements	OS that supports the feature.

Before You Begin

Table 36: Benefits, Resources, and Additional Information

Benefits	• **Threat identification** By configuring quantum keys, you can establish a secure quantum channel between the QKD devices. The feature improves threat identification and secures the network. • **Extend security** You can merge the existing keys with quantum keys and encrypt and decrypt the keys over existing VPN tunnels. The feature improves the security of the IPsec VPN infrastructure. • **Enhanced cryptographic strength** RFC 8784 compliance provides you with an easy way to prevent attackers from eavesdropping on the connection and intercepting the keys. The feature also ensures interoperability with other devices that adhere to the standard. • **Interoperability support** Use any QKD device supporting ETSI QKD Rest API.
Useful Resources	

Know more	• IPsec VPN • Route based IPsec VPN
Hands-on experience	vLABs Sandbox
Learn more	RFC 8784 - Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security

Functional Overview

Table 37: Static Key Manager Functional Overview

IPsec VPN	Deploy an IPsec VPN topology where the devices are connected by VPN tunnels that send traffic through the IPsec VPN tunnel. The VPN tunnels are later configured to use quantum keys making the tunnels quantum-safe VPN tunnels.
IKE gateway	Establish a secure connection, the IKE gateway uses the IKE policy to limit itself to the configured group of CAs (ca-profiles) while validating the certificate.
Proposals	
IKE proposal	Define the algorithms and keys used to establish the secure IKE connection with the peer security gateway. IKE creates the dynamic SAs and negotiates them for IPsec.
IPsec proposal	List protocols, algorithms, and security services to be negotiated with the remote IPsec peer.
Policies	
IKE policy	Define a combination of security parameters (IKE proposals) to be used during IKE negotiation.

IPsec policy	Contain rules and security policies to allow group VPN traffic between the zones specified.

Security policy	Allows you to select the type of data traffic to secure through the IPsec SAs.
	• VPN-OUT – Permits traffic from the trust zone to the vpn zone, where the match criteria is:
	  • source-address: HOST-1-Net
	  • destination-address: HOST-2-Net
	  • application: any
	• VPN-IN – Permits traffic from the vpn zone to the trust zone, where the match criteria is:
	  • source-address: HOST-2-Net
	  • destination-address: HOST-1-Net
	  • application: any

Profiles	

Key profile	Define how the devices use the static key profile to fetch the QKD keys on IPsec VPN to send traffic securely over the Internet. • Key profile—A static key-profile *km_profile_1* is configured for applications and services to retrieve the configured key-id and corresponding key. • IKE proposal—An IKE proposal *IKE_PROP* with the required algorithms to establish an IKE SA. • IKE policy—An IKE policy *IKE_POL* is configured to set the runtime negotiation and authentication attributes. • IKE gateway—An IKE gateway *IKE_GW* is configured to manage the IPsec tunnels between endpoints. A *ppk-profile* indicates which key-profile to use to establish Quantum safe IKE or IPsec SA. • IPsec proposal—An IPsec proposal *IPSEC_PROP* is configured with the required algorithms to establish an IPsec SA. • IPsec policy—An IPsec policy *IPSEC_POL* is configured to set the runtime IPsec negotiation attributes. • IPsec VPN—An IPsec VPN policy *IPSEC_VPN* is configured to set the range of subnets that needs to be secured. • Security zone—Three different security zones *trust*, *untrust* and *vpn* are configured for better segregation of expected traffic within each of these zones. • Security policy—Security policies *trust to vpn* and *vpn to trust* are configured between the security zones to filter out which type of data traffic gets secured through the IPsec SAs.
PPK Profile	Indicate which key profile to use to establish quantum-safe IKE or IPsec SAs by referencing the key profile under the IKE gateway.
Certificates	
CA certificate	Verify identity of devices and authenticate communication link between them.

Local certificate	Generate PKI and enroll it with the CA certificate for verification.
KME certificate	Third-party certificate generated by vendor
Security Zones	
trust	Network segment at the host zone
untrust	Network segment at the destination server zone
vpn	Network segment through which the SRX1 and SRX2 devices interact.
Primary verification tasks	Verify the established IKE and IPsec SAs are Quantum safe.

Topology Overview

In this example, SRX1 initiates the negotiation of quantum safe IPsec tunnels with SRX2 using CLI configured static key. SRX2 responds to this request by verifying SRX1's identity along with the key and establishes a quantum safe IPsec VPN. Once the tunnel is established, data traffic between Host1 and Host2 are secured using the established IPsec tunnel.

Table 38: Devices, Role, and Functionality used in this Configuration

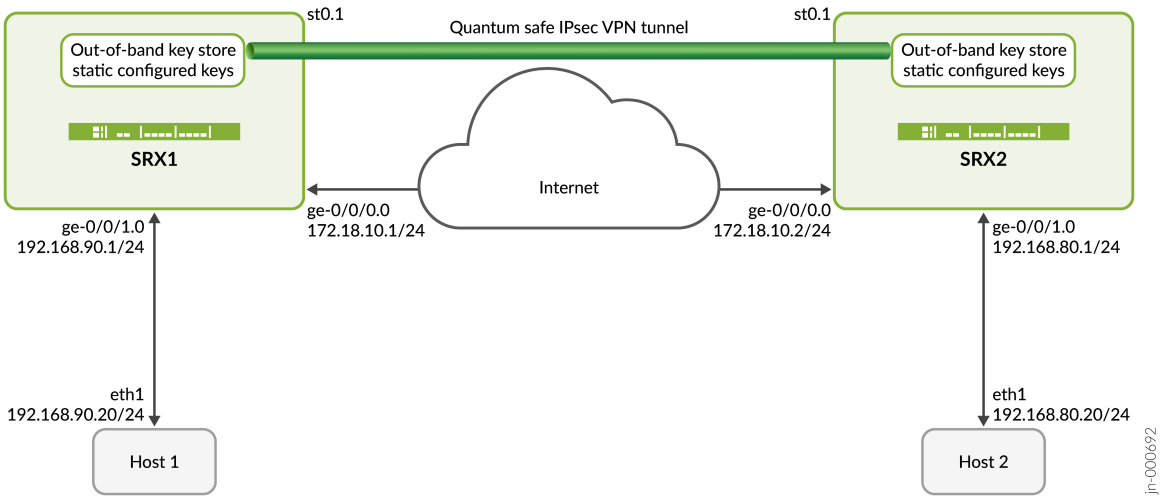
Hostname	Role	Function
SRX1	A device capable of establishing IPsec tunnels	Initiates IKE or IPsec SA negotiation and establishes Quantum-safe IPsec tunnels with SRX2 using static key configured on the SRX1.
SRX2	A device capable of establishing IPsec tunnels	Responds to the IKE or IPsec SA negotiation initiated by SRX1 and establishes Quantum-safe IPsec tunnels using static key configured on the SRX2.

Table 38: Devices, Role, and Functionality used in this Configuration *(Continued)*

Hostname	Role	Function
Host1	A Host inside the trusted zone or LAN side of SRX1	Initiates client-side traffic toward Host2
Host2	A Host inside the trusted zone or LAN side of SRX2	Responds to client-side traffic from Host1

Topology Illustration

Figure 19: Site-to-Site VPN



Step-By-Step Configuration on Firewall Devices

NOTE: For complete sample configurations on the DUT, see:

- "Set Commands on SRX1" on page 192
- "Set Commands on SRX2" on page 193

This configuration is applicable for only SRX1 and SRX2 devices. You must make the appropriate device-specific configuration changes.

1. Configure the interfaces.

```
[edit interfaces]
user@srx# set ge-0/0/0 unit 0 family inet address 172.18.10.1/24
user@srx# set st0 unit 1 family inet
user@srx# set ge-0/0/1 unit 0 family inet address 192.168.90.1/24
```

2. Configure a key profile of type static with a key-id and a corresponding key.

```
[edit security key-manager profiles]
user@srx# set km_profile_1 static key-id ascii-text test-key-id
user@srx# set km_profile_1 static key ascii-text qjwbdip139u5mcy89m28pcgowerefnkjsdg
```

3. Configure the security zones.

```
[edit security zones]
user@srx# set security-zone untrust host-inbound-traffic system-services ike
user@srx# set security-zone untrust interfaces ge-0/0/0.0
user@srx# set security-zone vpn interfaces st0.1
user@srx# set security-zone trust host-inbound-traffic system-services ping
user@srx# set security-zone trust interfaces ge-0/0/1.0
```

```
[edit security policies]
user@srx# set from-zone trust to-zone vpn policy vpn_out match source-address any
user@srx# set from-zone trust to-zone vpn policy vpn_out match destination-address any
user@srx# set from-zone trust to-zone vpn policy vpn_out match application any
user@srx# set from-zone trust to-zone vpn policy vpn_out then permit
user@srx# set from-zone vpn to-zone trust policy vpn_in match source-address any
user@srx# set from-zone vpn to-zone trust policy vpn_in match destination-address any
user@srx# set from-zone vpn to-zone trust policy vpn_in match application any
user@srx# set from-zone vpn to-zone trust policy vpn_in then permit
```

```
[edit security ike proposal]
user@srx# set IKE_PROP authentication-method pre-shared-keys
user@srx# set IKE_PROP dh-group group14
user@srx# set IKE_PROP authentication-algorithm sha-256
```

```
user@srx# set IKE_PROP encryption-algorithm aes-256-cbc
user@srx# set IKE_PROP lifetime-seconds 3600
```

```
[edit security ike policy]
user@srx# set IKE_POL proposals IKE_PROP
user@srx# set IKE_POL pre-shared-key ascii-text ipsec-test
```

```
[edit security ike gateway]
user@srx# set IKE_GW ike-policy IKE_POL
user@srx# set IKE_GW address 172.18.10.2
user@srx# set IKE_GW external-interface ge-0/0/0.0
user@srx# set IKE_GW local-address 172.18.10.1
user@srx# set IKE_GW version v2-only
user@srx# set IKE_GW ppk-profile km_profile_1
```

```
[edit security ipsec proposal]
user@srx# set IPSEC_PROP protocol esp
user@srx# set IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@srx# set IPSEC_PROP encryption-algorithm aes-256-cbc
user@srx# set IPSEC_PROP lifetime-seconds 2400
```

```
[edit security ipsec policy]
user@srx# set IPSEC_POL proposals IPSEC_PROP
```

```
[edit security ipsec vpn]
user@srx# set IPSEC_VPN bind-interface st0.1
user@srx# set IPSEC_VPN ike gateway IKE_GW
user@srx# set IPSEC_VPN ike ipsec-policy IPSEC_POL
user@srx# set IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
user@srx# set IPSEC_VPN traffic-selector ts1 remote-ip 192.168.80.0/24
user@srx# set IPSEC_VPN establish-tunnels immediately
```

Verification

IN THIS SECTION

- Verify IKE SAs | **187**
- Verify IPsec SAs | **189**
- Verify IPsec Statistics | **190**
- Verify Key Manager Profile | **191**
- Ping from HOST 1 to HOST 2 | **192**

This section provides a list of show commands that you can use to verify the feature in this example.

Table 39: Show Commands to Verify

Command	Verification Task
show security ike security-associations detail	"Verify that the IKE SAs are established." on page 187
show security ipsec security-associations detail	"Verify that the IPsec SAs are established." on page 189
show security ipsec statistics	"Verify IPsec encryption and decryption statistics." on page 190
show security key-manager profiles detail	"Verify key profile statistics." on page 191
ping 192.168.80.20 source 192.168.90.20 count 4	"Ping from HOST1 to HOST2 or vice versa." on page 192

Verify IKE SAs

Purpose

Verify the IKE SAs

Action

From operational mode, enter the `show security ike security-associations detail` command to view the IKE SAs.

```
user@srx> show security ike security-associations detail IKE peer 172.18.10.2, Index 1, Gateway
Name: IKE_GW
Role: Initiator, State: UP
Initiator cookie: dee592254e808a2b, Responder cookie: 51f6b1d4a8618332 Exchange type: IKEv2,
Authentication method: Pre-shared-keys
Local gateway interface: ge-0/0/2.0 Routing instance: default
Local: 172.18.10.1:500, Remote: 172.18.10.2:500
Lifetime: Expires in 1286 seconds Reauth Lifetime: Disabled
IKE Fragmentation: Enabled, Size: 576 SRG ID: 0
Remote Access Client Info: Unknown Client Peer ike-id: 172.18.10.2
AAA assigned IP: 0.0.0.0
PPK-profile: km_profile_1 Optional: No
State    : Used
Algorithms:
Authentication    : hmac-sha256-128
Encryption    : aes256-cbc Pseudo random function: hmac-sha256 Diffie-Hellman group : DH-group-14
Traffic statistics:
Input bytes :    1058
Output bytes :    1074
Input packets:    4
Output packets:    4
Input fragmented packets:    0
Output fragmented packets:    0
IPSec security associations: 4 created, 1 deleted Phase 2 negotiations in progress: 1
IPSec Tunnel IDs: 500002
Negotiation type: Quick mode, Role: Initiator, Message ID: 0 Local: 172.18.10.1:500, Remote:
172.18.10.2:500
Local identity: 172.18.10.1
Remote identity: 172.18.10.2 Flags: IKE SA is created
IPsec SA Rekey CREATE_CHILD_SA exchange stats:
Initiator stats:    Responder stats:
Request Out    : 0    Request In    : 1
Response In    : 0    Response Out    : 1
No Proposal Chosen In    : 0    No Proposal Chosen Out : 0
```

```
Invalid KE In    : 0    Invalid KE Out    : 0
TS Unacceptable In    : 0    TS Unacceptable Out    : 0
Res DH Compute Key Fail : 0    Res DH Compute Key Fail: 0 Res Verify SA Fail    : 0
Res Verify DH Group Fail: 0 Res Verify TS Fail    : 0
```

Meaning

The `Role: Initiator, State: UP, PPK-profile: km_profile_1 Optional: No, IPSec security associations: 4 created,` and `Flags: IKE SA is created` fields shows the device creates IKE SAs successfully.

Verify IPsec SAs

Purpose

Verify the IPsec SAs

Action

From operational mode, enter the `show security ipsec security-associations detail` command to view the IPsec SAs.

```
user@srx> show security ipsec security-associations detail

ID: 500002 Virtual-system: root, VPN Name: IPSEC_VPN Local Gateway: 172.18.10.1, Remote Gateway:
172.18.10.2 Traffic Selector Name: ts1
Local Identity: ipv4(192.168.90.0-192.168.90.255)
Remote Identity: ipv4(192.168.80.0-192.168.80.255) TS Type: traffic-selector
Version: IKEv2 Quantum Secured: Yes PFS group: N/A
SRG ID: 0
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: IPSEC_POL Port:
500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
Multi-sa, Configured SAs# 0, Negotiated SAs#: 0 Tunnel events:
Thu Mar 30 2023 23:43:42: IPsec SA negotiation succeeds (1 times)
Location: FPC 0, PIC 0, KMD-Instance 0
Anchorship: Thread 1
Distribution-Profile: default-profile Direction: inbound, SPI: 0x983a0221, AUX-SPI: 0
, VPN Monitoring: - Hard lifetime: Expires in 1330 seconds Lifesize Remaining: Unlimited
Soft lifetime: Expires in 662 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits) Anti-replay
```

```
service: counter-based enabled, Replay window size: 64
Extended-Sequence-Number: Disabled
tunnel-establishment: establish-tunnels-immediately IKE SA Index: 1
Direction: outbound, SPI: 0x4112746b, AUX-SPI: 0
, VPN Monitoring: - Hard lifetime: Expires in 1330 seconds Lifesize Remaining: Unlimited
Soft lifetime: Expires in 662 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed



Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits) Anti-replay
service: counter-based enabled, Replay window size: 64
Extended-Sequence-Number: Disabled
tunnel-establishment: establish-tunnels-immediately IKE SA Index: 1
```

Meaning

The `Version: IKEv2 Quantum Secured: Yes` and `tunnel-establishment: establish-tunnels-immediately IKE SA Index: 1` fields shows that device creates the IPsec SAs successfully.

The sample output confirms the IPsec SAs.

Verify IPsec Statistics

Purpose

Verify the IPsec statistics.

Action

From operational mode, enter the `show security ipsec statistics` command to view the IPsec statistics.

```
user@srx> show security ipsec statistics
ESP Statistics:
Encrypted bytes:    624
Decrypted bytes:    624
Encrypted packets:    4
Decrypted packets:    4
AH Statistics:
Input bytes:    0
Output bytes:    0
```

```
Input packets:    0
Output packets:    0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0 Bad headers: 0, Bad trailers: 0
Invalid SPI: 0, TS check fail: 0 Exceeds tunnel MTU: 0
Discarded: 0
```

Meaning

The `ESP Statistics` and `AH Statistics` fields shows the IPsec statistics.

Verify Key Manager Profile

Purpose

Verify the key manager profile.

Action

From operational mode, enter the `show security key-manager profiles` detail to view the key manager profile.

```
user@srx> show security key-manager profiles detail
Name: km_profile_1, Index: 1, Type: Static
Configured-at: 30.03.23 (23:22:43)
Time-elapsed: 1 hrs 16 mins 3 secs Request stats:
Received: 1
In-progress: 0
Success: 1
Failed: 0
```

Meaning

The `Name: km_profile_1` and `Type: Static` fields shows the key manager profile.

Ping from HOST 1 to HOST 2

Purpose

Verify the connectivity from HOST 1 to HOST 2.

Action

From operational mode, enter the ping 192.168.80.20 source 192.168.90.20 count 4 to view the connectivity from HOST 1 to HOST 2.

```
user@HOST1# ping 192.168.80.20 source 192.168.90.20 count 4
PING 192.168.80.20 (192.168.80.20): 56 data bytes
64 bytes from 192.168.80.1: icmp_seq=0 ttl=64 time=2.151 ms
64 bytes from 192.168.80.1: icmp_seq=1 ttl=64 time=1.710 ms
64 bytes from 192.168.80.1: icmp_seq=2 ttl=64 time=1.349 ms
64 bytes from 192.168.80.1: icmp_seq=3 ttl=64 time=1.597 ms
--- 192.168.80.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss round-trip min/avg/max/stddev =
1.349/1.702/2.151/0.290 ms
Data traffic is successfully flowing between the HOSTs
```

Meaning

The `PING 192.168.80.20 (192.168.80.20): 56 data bytes` confirms the connectivity from HOST 1 to HOST 2.

Appendix 1: Set Commands on all Devices

Set command output on all devices.

Set Commands on SRX1

```
set security key-manager profiles km_profile_1 static key-id ascii-text test-key-id
set security key-manager profiles km_profile_1 static key ascii-text
qjwbdip139u5mcy89m28pcgowerefnkjsdg
set interfaces ge-0/0/0 unit 0 family inet address 172.18.10.1/24
set interfaces st0 unit 1 family inet
set interfaces ge-0/0/1 unit 0 family inet address 192.168.90.1/24
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0
```

```
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group14
set security ike proposal IKE_PROP authentication-algorithm sha-256
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text ipsec-test
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 172.18.10.2
set security ike gateway IKE_GW external-interface ge-0/0/0.0
set security ike gateway IKE_GW local-address 172.18.10.1
set security ike gateway IKE_GW version v2-only
set security ike gateway IKE_GW ppk-profile km_profile_1
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal IPSEC_PROP lifetime-seconds 2400
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN bind-interface st0.1
set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
set security ipsec vpn IPSEC_VPN traffic-selector ts1 remote-ip 192.168.80.0/24
set security ipsec vpn IPSEC_VPN establish-tunnels immediately
```

Set Commands on SRX2

```
set security key-manager profiles km_profile_1 static key-id ascii-text test-key-id
set security key-manager profiles km_profile_1 static key ascii-text
qjwbdip139u5mcy89m28pcgowerefnkjsdg
```

```
set interfaces ge-0/0/0 unit 0 family inet address 172.18.10.2/24
set interfaces st0 unit 1 family inet
set interfaces ge-0/0/1 unit 0 family inet address 192.168.80.1/24
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group14
set security ike proposal IKE_PROP authentication-algorithm sha-256
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text “ipsec-test”
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 172.18.10.1
set security ike gateway IKE_GW external-interface ge-0/0/0.0
set security ike gateway IKE_GW local-address 172.18.10.2
set security ike gateway IKE_GW version v2-only
set security ike gateway IKE_GW ppk-profile km_profile_1
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal IPSEC_PROP lifetime-seconds 2400
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN bind-interface st0.1
set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
set security ipsec vpn IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set security ipsec vpn IPSEC_VPN establish-tunnels immediately
```

Appendix 2: Show Configuration Output on DUT

IN THIS SECTION

- SRX1 | 195
- SRX2 | 199

SRX1

From configuration mode, confirm your configuration by entering the `show security key-manager profiles`, `show security key-manager`, `show interfaces`, `show security zones`, `show security policies`, `show security ike proposal IKE_PROP`, `show security ike policy IKE_POL`, `show security ike gateway IKE_GW`, `show security ipsec proposal IPSEC_PROP`, `show security ipsec policy IPSEC_POL`, and `show security ipsec vpn IPSEC_VPN` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@srx1# show security key-manager profiles
km_profile_1 {
    static {
        key-id ascii-text "$9$.mz6pu1hyKBI8X-boajHqmF/hcylK836"; ## SECRET-DATA
        key ascii-text "$9$5Q6AhclXNbtuIcyeXxGDikfT369A0Bn/
vWLNY2aZUjPQAp0BEcFnyleMXxGDi.mT9CuhSeIElMLXwsaZUikPpu1hSen/eW8XbwJGD"; ## SECRET-DATA
    }
}
```

```
user@srx1# show security key-manager
profiles {
    km_profile_1 {
        static {
            key-id ascii-text "$9$.mz6pu1hyKBI8X-boajHqmF/hcylK836"; ## SECRET-DATA
            key ascii-text "$9$5Q6AhclXNbtuIcyeXxGDikfT369A0Bn/
vWLNY2aZUjPQAp0BEcFnyleMXxGDi.mT9CuhSeIElMLXwsaZUikPpu1hSen/eW8XbwJGD"; ## SECRET-DATA
        }
```

```
    }
}
```

```
user@srx1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.2/24;
            address 172.18.10.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.80.1/24;
            address 192.168.90.1/24;
        }
        family mpls;
    }
}
ge-1/0/0 {
    unit 0 {
        family mpls;
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
```

```
user@srx1# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
```

```
        ge-0/0/0.0;
    }
}
security-zone vpn {
    interfaces {
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

```
user@srx1# show security policies
from-zone trust to-zone vpn {
    policy vpn_out {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn_in {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
```

```
        }
    }
}
```

```
user@srx1# show security ike proposal IKE_PROP
authentication-method pre-shared-keys;
dh-group group14;
authentication-algorithm sha-256;
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@srx1# show security ike policy IKE_POL
proposals IKE_PROP;
pre-shared-key ascii-text "$9$z0C63/tp0Icrvz39p0Ihcs24aZjqmTn9p"; ## SECRET-DATA
```

```
user@srx1# show security ike gateway IKE_GW
ike-policy IKE_POL;
address [ 172.18.10.1 172.18.10.2 ];
external-interface ge-0/0/0.0;
local-address 172.18.10.1;
version v2-only;
ppk-profile km_profile_1;
```

```
user@srx1# show security ipsec proposal IPSEC_PROP
protocol esp;
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
lifetime-seconds 2400;
```

```
user@srx1# show security ipsec policy IPSEC_POL
proposals IPSEC_PROP;
```

```
user@srx1# show security ipsec vpn IPSEC_VPN
bind-interface st0.1;
```

```
ike {
    gateway IKE_GW;
    ipsec-policy IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.90.0/24;
    remote-ip 192.168.80.0/24;
}
establish-tunnels immediately;
```

SRX2

From configuration mode, confirm your configuration by entering the `show security key-manager profiles`, `show security key-manager`, `show interfaces`, `show security zones`, `show security policies`, `show security ike proposal IKE_PROP`, `show security ike policy IKE_POL`, `show security ike gateway IKE_GW`, `show security ipsec proposal IPSEC_PROP`, `show security ipsec policy IPSEC_POL`, and `show security ipsec vpn IPSEC_VPN` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@srx2# show security key-manager profiles
km_profile_1 {
    static {
        key-id ascii-text "$9$Hk5FCA0IhruOvWx-2gGDikT3IRhSrvQF"; ## SECRET-DATA
        key ascii-text "$9$zDD33CuyrvNVY0BhreMN-jHqmQF/Ctu1R9A8X7V4oGDikT3uO1RSr69evMLN-
jHqf5FtpBylMhSvL7N2gGDiqmTOBEylM9AMXxNY2UjH"; ## SECRET-DATA
    }
}
```

```
user@srx2# show security key-manager
profiles {
    km_profile_1 {
        static {
            key-id ascii-text "$9$Hk5FCA0IhruOvWx-2gGDikT3IRhSrvQF"; ## SECRET-DATA
            key ascii-text "$9$zDD33CuyrvNVY0BhreMN-jHqmQF/Ctu1R9A8X7V4oGDikT3uO1RSr69evMLN-
jHqf5FtpBylMhSvL7N2gGDiqmTOBEylM9AMXxNY2UjH"; ## SECRET-DATA
        }
```

```
    }
}
```

```
user@srx2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
        }
        family mpls;
    }
}
ge-1/0/0 {
    unit 0 {
        family mpls;
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
```

```
user@srx2# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
```

```
        ge-0/0/0.0;
    }
}
security-zone vpn {
    interfaces {
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

```
user@srx2# show security policies
from-zone trust to-zone vpn {
    policy vpn_out {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn_in {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
```

```
        }
    }
}
```

```
user@srx2# show security ike proposal IKE_PROP
authentication-method pre-shared-keys;
dh-group group14;
authentication-algorithm sha-256;
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@srx2# show security ike policy IKE_POL
proposals IKE_PROP;
pre-shared-key ascii-text "$9$zTi03/tp0Icrvz39p0Ihcs24aZjqmTn9p"; ## SECRET-DATA
```

```
user@srx2# show security ike gateway IKE_GW
ike-policy IKE_POL;
address 172.18.10.1;
external-interface ge-0/0/0.0;
local-address 172.18.10.2;
version v2-only;
ppk-profile km_profile_1;
```

```
user@srx2# show security ipsec proposal IPSEC_PROP
protocol esp;
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
lifetime-seconds 2400;
```

```
user@srx2# show security ipsec policy IPSEC_POL
proposals IPSEC_PROP;

[edit]
user@srx2# show security ipsec vpn IPSEC_VPN
bind-interface st0.1;
ike {
```

```
    gateway IKE_GW;
    ipsec-policy IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.80.0/24;
    remote-ip 192.168.90.0/24;
}
establish-tunnels immediately;
```

Example: Configure Static Keys Profile for AutoVPN

SUMMARY

Use this configuration example to secure an IPsec AutoVPN infrastructure by configuring the static key profile.

IN THIS SECTION

- Example Prerequisites | **204**
- Before You Begin | **204**
- Functional Overview | **205**
- Topology Overview | **209**
- Topology Illustration | **211**
- Step-By-Step Configuration on Hub | **211**
- Step-By-Step Configuration on Spoke Devices | **214**
- Verification | **217**
- Appendix 1: Set Commands on all Devices | **226**
- Appendix 2: Show Configuration Output on DUT | **230**

You can secure an IPsec AutoVPN infrastructure by configuring the static key profile.

In this configuration example, the Hub, Spoke 1, and Spoke 2 use static key profiles to fetch the QKD keys on IPsec VPN. The QKD keys help send traffic securely over the Internet.

TIP:

Table 40: Estimated Timers

Reading Time	Less than an hour
Configuration Time	Less than an hour

Example Prerequisites

Use Feature Explorer to confirm platform and release support for specific features.

Table 41: Requirements

Hardware requirements	• Devices that support the feature. • Third-party Key Management Entity (KME) or Quantum Key Distribution (QKD) devices. The KME parameters are as per ETSI GS QKD 014 specification.
Software requirements	OS that supports the feature.

Before You Begin

Table 42: Benefits, Resources, and Additional Information

Benefits	• **Threat identification** By configuring quantum keys, you can establish a secure quantum channel between the QKD devices. This improves threat identification and secures the network. • **Extend security** You can merge the existing keys with quantum keys and encrypt and decrypt them over existing VPN tunnels. This improves the security of the IPsec VPN infrastructure. • **Enhanced cryptographic strength** RFC 8784 compliance provides you with an easy way to prevent attackers from eavesdropping on the connection and intercepting the keys. This also ensures interoperability with other devices that adhere to the standard. • **Interoperability support** Use any QKD device supporting ETSI QKD Rest API.
Useful Resources	
Know more	• IPsec VPN • AutoVPN on Hub-and-Spoke Devices
Hands-on experience	vLABs Sandbox
Learn more	RFC 8784 - Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) then you submits requests for local certificates. See Understanding Local Certificate Requests. Enroll the digital certificates in each device. See Example: Loading CA and Local Certificates Manually.

Functional Overview

Table 43: Static Key Manager Functional Overview

IPsec VPN	Deploys a hub-and-spoke IPsec VPN topology where spokes are connected by VPN tunnels that send traffic through the hub. These VPN tunnels are later configured to use quantum keys making them quantum-safe VPN tunnels.
IKE gateway	Establishes a secure connection, the IKE gateway uses the IKE policy to limit itself to the configured group of CAs (ca-profiles) while validating the certificate.
Proposals	
IKE proposal	Defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. IKE creates the dynamic SAs and negotiates them for IPsec.
IPsec proposal	Lists protocols, algorithms, and security services to be negotiated with the remote IPsec peer.
Policies	
IKE policy	Defines a combination of security parameters (IKE proposals) to be used during IKE negotiation.
IPsec policy	Contains rules and security policies to allow group VPN traffic between the zones specified.

Security policy	Allows you to select the type of data traffic to secure through the IPsec SAs.
	• VPN-OUT—Permits traffic from the trust zone to the vpn zone, where the match criteria is:
	  • source-address: HOST-1-Net
	  • destination-address: HOST-2-Net
	  • application: any
	• VPN-IN—Permits traffic from the vpn zone to the trust zone, where the match criteria is:
	  • source-address: HOST-2-Net
	  • destination-address: HOST-1-Net
	  • application: any

Profiles	

Key profile	Define how the devices communicate with the KME devices to retrieve QKD keys from the external KME server. Key profiles are configured on the hub (HUB_KM_PROFILE_1) and spokes (SPOKE_1_KM_PROFILE_1 and SPOKE_2_KM_PROFILE_1) separately. • Key profile—Static key-profiles *HUB_KM_PROFILE_1*, *SPOKE_1_KM_PROFILE_1* and *SPOKE_2_KM_PROFILE_1* are configured on the HUB, SPOKE-1 and SPOKE-2 respectively for applications/services to retrieve a CLI configured key-id and corresponding key. • IKE proposal—IKE proposals *HUB_IKE_PROP*, *SPOKE_1_IKE_PROP* and *SPOKE_2_IKE_PROP* are configured on the HUB, SPOKE-1 and SPOKE-2 respectively with the required algorithms for establishing an IKE security association. • IKE policy—IKE policies *HUB_IKE_POL*, *SPOKE_1_IKE_POL* and *SPOKE_3_IKE_POL* are configured on the HUB, SPOKE-1 and SPOKE-2 respectively to set the runtime negotiation/authentication attributes. • IKE gateway—IKE gateways *HUB_IKE_GW*, *SPOKE_1_IKE_GW* and *SPOKE_2_IKE_GW* are configured on the HUB, SPOKE-1 and SPOKE-2 respectively to set the endpoints between whom the IPsec tunnels need to be established, reference the configured IKE policy, the version of IKE that needs to be used and a ppk-profile to signify which key-profile needs to be used to establish Quantum safe IKE/IPsec security associations. • IPsec proposal—IPSEC proposals *HUB_IPSEC_PROP*, *SPOKE_1_IPSEC_PROP* and *SPOKE_2_IPSEC_PROP* are configured on the HUB, SPOKE-1 and SPOKE-2 respectively with the required algorithms for establishing an IPSEC security association. • IPsec policy—IPSEC policies *HUB_IPSEC_POL*, *SPOKE_1_IPSEC_POL* and *SPOKE_2_IPSEC_POL* are configured on the HUB, SPOKE-1 and SPOKE-2 respectively to set the runtime IPsec negotiation attributes. • IPsec VPN—IPSEC VPNs *HUB_IPSEC_VPN*, *SPOKE_1_IPSEC_VPN* and *SPOKE_2_IPSEC_VPN* are

	configured on the HUB, SPOKE-1 and SPOKE-2 respectively to set the range of subnets that needs to be secured, reference the configured ipsec policy and ike gateway. • Security zone—3 different security zones *trust*, *untrust* and *vpn* are configured for better segregation of expected traffic within each of these zones. • Security policy—Security policies *trust to vpn* and *vpn to trust* are configured between the security zones to filter out which type of data traffic gets secured through the IPsec security associations.
PPK Profile	Indicates which key profile to use to establish quantum-safe IKE or IPsec SAs by referencing the key profile under the IKE gateway.
Certificates	
CA certificate	Verifies identity of devices and authenticate communication link between them.
Local certificate	Generates PKI and enroll it with the CA certificate for verification.
KME certificate	Third-party certificate generated by vendor.
Security Zones	
trust	Network segment at the host zone.
untrust	Network segment at the destination server zone.
vpn	Network segment through which the hub-and-spoke interacts.
Primary verification tasks	Verify the established IKE and IPsec SAs are Quantum safe.

Topology Overview

In this example, SPOKE 1 and SPOKE 2 initiate the negotiation of quantum-safe IPsec tunnels with the Hub using CLI-configured static key. The Hub responds to the requests by verifying the identity of

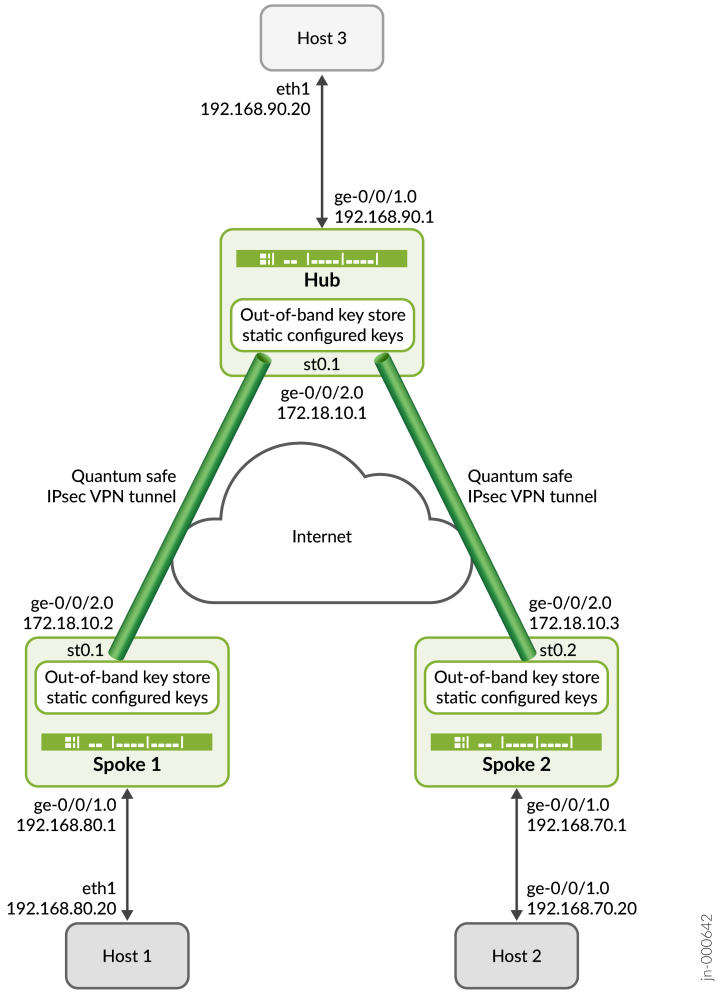
Spoke 1 and Spoke 2 along with their respective keys and establishes a quantum-safe IPsec VPN with both the spokes. Once the tunnels are established, data traffic between Host 1 and Host 3, and between Host 2 and Host 3 are secured using the established IPsec tunnels.

Table 44: Devices, Role, and Functionality used in this Configuration

Hostname	Role	Function
Hub	Device capable of establishing IPsec tunnels	Responds to IKE or IPsec SA negotiation initiated by SPOKE 1 and SPOKE 2 and establishes quantum-safe IPsec tunnels using static key configured on the Hub device.
Spoke 1	Device capable of establishing IPsec tunnels	Initiates IKE/IPsec SA negotiation and establishes quantum-safe IPsec tunnels with the Hub using static key configured on the Spoke 1.
Spoke 2	Device capable of establishing IPsec tunnels	Initiates IKE or IPsec SA negotiation and establishes quantum-safe IPsec tunnels with the Hub using static key configured on the Spoke 2.
Host 1	Host inside the trusted zone or LAN side of Spoke 1	Initiates client-side traffic toward Host 3.
Host 2	Host inside the trusted zone or LAN side of Spoke 2	Initiates client-side traffic toward Host 3.
Host 3	Host inside the trusted zone or LAN side of HUB	Responds to client-side traffic from Host 1 and Host 2.

Topology Illustration

Figure 20: Static Key with Auto VPN



Step-By-Step Configuration on Hub

NOTE: For complete sample configurations on the DUT, see:

- "Set Commands on Hub" on page 226

This configuration is applicable for only the Hub devices. You must make the appropriate device-specific configuration changes.

1. Configure the hub interfaces.

```
[edit interfaces]
user@hub# set ge-0/0/2 unit 0 family inet address 172.18.10.1/24
user@hub# set ge-0/0/1 unit 0 family inet address 192.168.90.1/24
user@hub# set st0 unit 1 family inet
```

2. Configure the CA profile and CA certificate.

```
[edit security pki]
user@hub# set ca-profile Root-CA ca-identity Root-CA
user@hub# set ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
user@hub# set ca-profile Root-CA revocation-check disable
```

3. From the operational mode, bind the CA certificate to CA profile.

```
user@hub> request security pki ca-certificate enroll ca-profile Root-CA
user@hub> request security pki generate-key-pair certificate-id HUB_CRT size 2048 type rsa
user@hub> request security pki local-certificate enroll certificate-id HUB_CRT challenge-
password <different> domain-name hub.juniper.net email hub@juniper.net subject
DC=juniper,CN=hub.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-profile
Root-CA
```

4. Configure the static key manager profile.

```
[edit security key-manager profiles]
user@hub# set HUB_KM_PROFILE_1 static key-id ascii-text test-key-id
user@hub# set HUB_KM_PROFILE_1 static key ascii-text qjwbdip139u5mcy89m28pcgowerefnkjsdg
```

5. Configure the hub-spoke on the IPsec VPN. This includes configuring the security zones, security policies, and relevant certificates for authenticating device identities and their communication links.

```
[edit security ike proposal]
user@hub# set HUB_IKE_PROP authentication-method rsa-signatures
user@hub# set HUB_IKE_PROP dh-group group14
user@hub# set HUB_IKE_PROP authentication-algorithm sha-256
```

```
user@hub# set HUB_IKE_PROP encryption-algorithm aes-256-cbc
user@hub# set HUB_IKE_PROP lifetime-seconds 3600
```

```
[edit security ike policy]
user@hub# set HUB_IKE_POL proposals HUB_IKE_PROP
user@hub# set HUB_IKE_POL certificate local-certificate HUB_CRT
```

```
[edit security ike gateway]
user@hub# set HUB_IKE_GW local-address 172.18.10.1
user@hub# set HUB_IKE_GW ike-policy HUB_IKE_POL
user@hub# set HUB_IKE_GW external-interface ge-0/0/2.0
user@hub# set HUB_IKE_GW local-identity distinguished-name
user@hub# set HUB_IKE_GW dynamic ike-user-type group-ike-id
user@hub# set HUB_IKE_GW dynamic distinguished-name wildcard C=us,DC=juniper
user@hub# set HUB_IKE_GW ppk-profile HUB_KM_PROFILE_1
user@hub# set HUB_IKE_GW version v2-only
```

```
[edit security ipsec proposal]
user@hub# set HUB_IPSEC_PROP protocol esp
user@hub# set HUB_IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@hub# set HUB_IPSEC_PROP encryption-algorithm aes-256-cbc
```

```
[edit security ipsec policy]
user@hub# set HUB_IPSEC_POL proposals HUB_IPSEC_PROP
```

```
[edit security ipsec vpn]
user@hub# set HUB_IPSEC_VPN bind-interface st0.1
user@hub# set HUB_IPSEC_VPN ike gateway HUB_IKE_GW
user@hub# set HUB_IPSEC_VPN ike ipsec-policy HUB_IPSEC_POL
user@hub# set HUB_IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
user@hub# set HUB_IPSEC_VPN traffic-selector ts1 remote-ip 0.0.0.0/0
```

```
[edit security zones]
user@hub# set security-zone untrust host-inbound-traffic system-services ike
user@hub# set security-zone untrust interfaces ge-0/0/2.0
```

```
user@hub# set security-zone vpn interfaces st0.1
user@hub# set security-zone trust host-inbound-traffic system-services ping
user@hub# set security-zone trust interfaces ge-0/0/1.0
```

```
[edit security policies]
user@hub# set from-zone trust to-zone vpn policy vpn_out match source-address any
user@hub# set from-zone trust to-zone vpn policy vpn_out match destination-address any
user@hub# set from-zone trust to-zone vpn policy vpn_out match application any
user@hub# set from-zone trust to-zone vpn policy vpn_out then permit
user@hub# set from-zone vpn to-zone trust policy vpn_in match source-address any
user@hub# set from-zone vpn to-zone trust policy vpn_in match destination-address any
user@hub# set from-zone vpn to-zone trust policy vpn_in match application any
user@hub# set from-zone vpn to-zone trust policy vpn_in then permit
```

Step-By-Step Configuration on Spoke Devices

NOTE: For complete sample configurations on the DUT, see:

- "Set Commands on Spoke 1" on page 227
- "Set Commands on Spoke 2" on page 229

This configuration is applicable for Spoke 1 and Spoke 2 devices. For other devices, you must make appropriate device-specific configuration changes.

1. Configure the spoke interfaces.

```
[edit interfaces]
user@spoke# set ge-0/0/2 unit 0 family inet address 172.18.10.2/24
user@spoke# set ge-0/0/1 unit 0 family inet address 192.168.80.1/24
user@spoke# set st0 unit 1 family inet
```

2. Configure hub-spoke on the IPsec VPN. This includes configuring the security zones, security policies, and relevant certificates for authenticating device identities and their communication links.

```
[edit security ike proposal]
user@spoke# set SPOKE_1_IKE_PROP authentication-method rsa-signatures
user@spoke# set SPOKE_1_IKE_PROP dh-group group14
user@spoke# set SPOKE_1_IKE_PROP authentication-algorithm sha-256
```

```
user@spoke# set SPOKE_1_IKE_PROP encryption-algorithm aes-256-cbc
user@spoke# set SPOKE_1_IKE_PROP lifetime-seconds 3600
```

```
[edit security ike policy]
user@spoke# set SPOKE_1_IKE_POL proposals SPOKE_1_IKE_PROP
user@spoke# set SPOKE_1_IKE_POL certificate local-certificate SPOKE_1_CRT
```

```
[edit security ike gateway]
user@spoke# set SPOKE_1_IKE_GW address 172.18.10.1
user@spoke# set SPOKE_1_IKE_GW local-address 172.18.10.2
user@spoke# set SPOKE_1_IKE_GW ike-policy SPOKE_1_IKE_POL
user@spoke# set SPOKE_1_IKE_GW external-interface ge-0/0/2.0
user@spoke# set SPOKE_1_IKE_GW local-identity distinguished-name
user@spoke# set SPOKE_1_IKE_GW remote-identity distinguished-name
user@spoke# set SPOKE_1_IKE_GW ppk-profile SPOKE_1_KM_PROFILE_1
user@spoke# set SPOKE_1_IKE_GW version v2-only
```

```
[edit security ipsec proposal]
user@spoke# set SPOKE_1_IPSEC_PROP protocol esp
user@spoke# set SPOKE_1_IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@spoke# set SPOKE_1_IPSEC_PROP encryption-algorithm aes-256-cbc
```

```
[edit security ipsec policy]
user@spoke# set SPOKE_1_IPSEC_POL proposals SPOKE_1_IPSEC_PROP
```

```
[edit security ipsec vpn]
user@spoke# set SPOKE_1_IPSEC_VPN bind-interface st0.1
user@spoke# set SPOKE_1_IPSEC_VPN ike gateway SPOKE_1_IKE_GW
user@spoke# set SPOKE_1_IPSEC_VPN ike ipsec-policy SPOKE_1_IPSEC_POL
user@spoke# set SPOKE_1_IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
user@spoke# set SPOKE_1_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
```

```
[edit security zones]
user@spoke# set security-zone untrust host-inbound-traffic system-services ike
user@spoke# set security-zone untrust interfaces ge-0/0/2.0
```

```
user@spoke# set security-zone vpn interfaces st0.1
user@spoke# set security-zone trust host-inbound-traffic system-services ping
user@spoke# set security-zone trust interfaces ge-0/0/1.0
```

```
[edit security policies]
user@spoke# set from-zone trust to-zone vpn policy vpn_out match source-address any
user@spoke# set from-zone trust to-zone vpn policy vpn_out match destination-address any
user@spoke# set from-zone trust to-zone vpn policy vpn_out match application any
user@spoke# set from-zone trust to-zone vpn policy vpn_out then permit
user@spoke# set from-zone vpn to-zone trust policy vpn_in match source-address any
user@spoke# set from-zone vpn to-zone trust policy vpn_in match destination-address any
user@spoke# set from-zone vpn to-zone trust policy vpn_in match application any
user@spoke# set from-zone vpn to-zone trust policy vpn_in then permit
```

```
[edit security pki ]
user@spoke# set ca-profile Root-CA ca-identity Root-CA
user@spoke# set ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
user@spoke# set ca-profile Root-CA revocation-check disable
```

```
user@spoke> request security pki ca-certificate enroll ca-profile Root-CA
user@spoke> request security pki generate-key-pair certificate-id SPOKE_1_CRT size 2048 type
rsa
user@spoke> request security pki local-certificate enroll certificate-id SPOKE_1_CRT
challenge-password <different> domain-name spoke_1.juniper.net email spoke_1@juniper.net
subject
DC=juniper,CN=spoke_1.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
```

3. Configure the static key manager profile.

```
[edit security key-manager profiles]
user@spoke# set SPOKE_1_KM_PROFILE_1 static key-id ascii-text test-key-id
user@spoke# set SPOKE_1_KM_PROFILE_1 static key ascii-text qjwbdip139u5mcy89m28pcgowerefnkjsdg
```

Verification

IN THIS SECTION

- Verify IKE SAs | **218**
- Verify IPsec SAs | **221**
- Verify IPsec Statistics | **223**
- Verify Key Manager Profile | **224**
- Ping from Host 1 to Host 3 or vice versa | **225**
- Ping from Host 2 to Host 3 or vice versa | **225**

This section provides a list of show commands that you can use to verify the feature in this example.

Command	Verification Task
show security ike security-associations detail	"Verify that the IKE SAs are established." on page 221
show security ipsec security-associations detail	"Verify that the IPsec SAs are established." on page 218
show security ipsec statistics	"Verify IPsec encryption and decryption statistics." on page 223
show security key-manager profiles detail	"Verify key profile statistics." on page 224
ping 192.168.90.20 source 192.168.80.20 count 4	"Ping from Host 1 to Host 3 or vice versa." on page 225

Verify IKE SAs

Purpose

Verify the IKE SAs.

Action

From operational mode, enter the `show security ike security-associations detail` command to view the IKE SAs.

```
user@hub> show security ike security-associations detail

IKE peer 172.18.10.2, Index 2123, Gateway Name: HUB_IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 0e40ccdcee1b54bd, Responder cookie: 43964f5cc4d4491c
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/2.0
  Routing instance: default
  Local: 172.18.10.1:500, Remote: 172.18.10.2:500
  Lifetime: Expires in 2840 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke.juniper.net
  AAA assigned IP: 0.0.0.0
  PPK-profile: HUB_KM_PROFILE_1
     Optional: No
     State   : Used
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                 2610
   Output bytes  :                 2571
   Input  packets:                    5
   Output packets:                    5
   Input  fragmented packets:       4
   Output fragmented packets:       4
```

```
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500440

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.18.10.1:500, Remote: 172.18.10.2:500
    Local identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=hub.juniper.net
    Remote identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke.juniper.net
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                Responder stats:
    Request Out             : 0                     Request In             :
0
    Response In             : 0                     Response Out           :
0
    No Proposal Chosen In   : 0                     No Proposal Chosen Out :
0
    Invalid KE In           : 0                     Invalid KE Out         :
0
    TS Unacceptable In      : 0                     TS Unacceptable Out    :
0
    Res DH Compute Key Fail : 0                     Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0

IKE peer 172.18.10.3, Index 2124, Gateway Name: HUB_IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 651bf4a52a9375ec, Responder cookie: d9a9c95c27e3f929
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/2.0
  Routing instance: default
  Local: 172.18.10.1:500, Remote: 172.18.10.3:500
  Lifetime: Expires in 2901 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke_2.juniper.net
```

```
  AAA assigned IP: 0.0.0.0
  PPK-profile: HUB_KM_PROFILE_1
     Optional: No
     State   : Used
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                 2610
   Output bytes  :                 2571
   Input  packets:                    5
   Output packets:                    5
   Input  fragmented packets:       4
   Output fragmented packets:       4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500441

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.18.10.1:500, Remote: 172.18.10.3:500
    Local identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=hub.juniper.net
    Remote identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke_2.juniper.net
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                Responder stats:
    Request Out             : 0                     Request In             :
0
    Response In             : 0                     Response Out           :
0
    No Proposal Chosen In   : 0                     No Proposal Chosen Out :
0
    Invalid KE In           : 0                     Invalid KE Out         :
0
    TS Unacceptable In      : 0                     TS Unacceptable Out    :
0
    Res DH Compute Key Fail : 0                     Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
```

```
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

Meaning

The `Role: Responder, State: UP`, `PPK-profile: HUB_KM_PROFILE_1`, `IPSec security associations: 2 created, 0 deleted`, and `Flags: IKE SA is created` fields shows the IKE SAs are created successfully.

Verify IPsec SAs

Purpose

Verify the IPsec SAs.

Action

From operational mode, enter the `show security ipsec security-associations detail` command to view the IPsec SAs.

```
user@hub> show security ipsec security-associations detail

ID: 500440 Virtual-system: root, VPN Name: HUB_IPSEC_VPN
  Local Gateway: 172.18.10.1, Remote Gateway: 172.18.10.2
  Traffic Selector Name: ts1
  Local Identity: ipv4(192.168.90.0-192.168.90.255)
  Remote Identity: ipv4(192.168.80.0-192.168.80.255)
  TS Type: traffic-selector
  Version: IKEv2
  Quantum Secured: Yes
  PFS group: N/A
  Passive mode tunneling: Disabled
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: HUB_IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Thu Jul 20 2023 10:44:19: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x649d371f, AUX-SPI: 0
                              , VPN Monitoring: -
```

```
    Hard lifetime: Expires in 2840 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2183 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2123
  Direction: outbound, SPI: 0xd5ef611e, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2840 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2183 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2123

ID: 500441 Virtual-system: root, VPN Name: HUB_IPSEC_VPN
  Local Gateway: 172.18.10.1, Remote Gateway: 172.18.10.3
  Traffic Selector Name: ts1
  Local Identity: ipv4(192.168.90.0-192.168.90.255)
  Remote Identity: ipv4(192.168.70.0-192.168.70.255)
  TS Type: traffic-selector
  Version: IKEv2
  Quantum Secured: Yes
  PFS group: N/A
  Passive mode tunneling: Disabled
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: HUB_IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Thu Jul 20 2023 10:45:19: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0xa0d3ba32, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2901 seconds
    Lifesize Remaining:  Unlimited
```

```
    Soft lifetime: Expires in 2258 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2124
  Direction: outbound, SPI: 0xe54414e3, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2901 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2258 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2124
```

Meaning

The `Quantum Secured: Yes`, `Passive mode tunneling: Disabled`, `Policy-name: HUB_IPSEC_POL`, and `IPsec SA negotiation succeeds (1 times)` fields shows the IPsec SAs are created successfully.

Verify IPsec Statistics

Purpose

Verify the IPsec statistics.

Action

From operational mode, enter the `show security ipsec statistics` command to view the IPsec statistics.

```
user@hub> show security ipsec statistics

ESP Statistics:
  Encrypted bytes:             1248
  Decrypted bytes:             1248
  Encrypted packets:              8
  Decrypted packets:              8
```

```
AH Statistics:
  Input bytes:                  0
  Output bytes:                 0
  Input packets:                0
  Output packets:               0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Exceeds tunnel MTU: 0
  Discarded: 0
```

Meaning

The `ESP Statistics` and `AH Statistics` fields shows the IPsec statistics.

Verify Key Manager Profile

Purpose

Verify the key manager profile.

Action

From operational mode, enter the `show security key-manager profiles detail` command to view the key manager profile.

```
user@hub> show security key-manager profiles detail

Name: HUB_KM_PROFILE_1, Index: 4, Type: Static
  Configured-at: 20.07.23 (09:59:06)
  Time-elapsed: 1 hrs 2 mins 7 secs
  Request stats:
    Received: 2
    In-progress: 0
    Success: 2
    Failed: 0
```

Meaning

The `Name: HUB_KM_PROFILE_1` and `Type: Static` fields shows the key manager profile

Ping from Host 1 to Host 3 or vice versa

Purpose

Verify the connectivity from Host 1 to Host 3.

Action

From operational mode, enter the `ping 192.168.90.20 source 192.168.80.20 count 4` command to view the connectivity from Host 1 to Host 3.

```
user@HOST1# ping 192.168.90.20 source 192.168.80.20 count 4
PING 192.168.90.20 (192.168.90.20): 56 data bytes
64 bytes from 192.168.90.20: icmp_seq=0 ttl=64 time=2.151 ms
64 bytes from 192.168.90.20: icmp_seq=1 ttl=64 time=1.710 ms
64 bytes from 192.168.90.20: icmp_seq=2 ttl=64 time=1.349 ms
64 bytes from 192.168.90.20: icmp_seq=3 ttl=64 time=1.597 ms
--- 192.168.90.20 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.349/1.702/2.151/0.290 ms

Data traffic is successfully flowing between the HOSTs
```

Meaning

The `PING 192.168.80.20 (192.168.80.20): 56 data bytes` confirms the connectivity from HOST 1 to HOST 3.

Ping from Host 2 to Host 3 or vice versa

Purpose

Verify the connectivity from Host 2 to Host 3.

Action

From operational mode, enter the `ping 192.168.90.20 source 192.168.80.20 count 4` to view the connectivity from Host 2 to Host 3.

```
user@HOST1# ping 192.168.90.20 source 192.168.70.20 count 4
PING 192.168.90.20 (192.168.90.20): 56 data bytes
64 bytes from 192.168.90.20: icmp_seq=0 ttl=64 time=2.151 ms
64 bytes from 192.168.90.20: icmp_seq=1 ttl=64 time=1.710 ms
64 bytes from 192.168.90.20: icmp_seq=2 ttl=64 time=1.349 ms
64 bytes from 192.168.90.20: icmp_seq=3 ttl=64 time=1.597 ms
--- 192.168.90.20 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.349/1.702/2.151/0.290 ms

Data traffic is successfully flowing between the HOSTs
```

Meaning

The `PING 192.168.80.20 (192.168.80.20): 56 data bytes` confirms the connectivity from HOST 2 to HOST 3.

Appendix 1: Set Commands on all Devices

Set command output on all devices.

Set Commands on Hub

```
set security ike proposal HUB_IKE_PROP dh-group group14
set security ike proposal HUB_IKE_PROP authentication-algorithm sha-256
set security ike proposal HUB_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal HUB_IKE_PROP lifetime-seconds 3600
set security ike policy HUB_IKE_POL proposals HUB_IKE_PROP
set security ike policy HUB_IKE_POL certificate local-certificate HUB_CRT
set security ike gateway HUB_IKE_GW local-address 172.18.10.1
set security ike gateway HUB_IKE_GW ike-policy HUB_IKE_POL
set security ike gateway HUB_IKE_GW external-interface ge-0/0/2.0
set security ike gateway HUB_IKE_GW local-identity distinguished-name
set security ike gateway HUB_IKE_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_IKE_GW dynamic distinguished-name wildcard C=us,DC=juniper
set security ike gateway HUB_IKE_GW ppk-profile HUB_KM_PROFILE_1
set security ike gateway HUB_IKE_GW version v2-only
```

```
set security ipsec proposal HUB_IPSEC_PROP protocol esp
set security ipsec proposal HUB_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal HUB_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy HUB_IPSEC_POL proposals HUB_IPSEC_PROP
set security ipsec vpn HUB_IPSEC_VPN bind-interface st0.1
set security ipsec vpn HUB_IPSEC_VPN ike gateway HUB_IKE_GW
set security ipsec vpn HUB_IPSEC_VPN ike ipsec-policy HUB_IPSEC_POL
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 remote-ip 0.0.0.0/0
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.1/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.90.1/24
set interfaces st0 unit 1 family inet
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Set Commands on Spoke 1

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id SPOKE_1_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_1_CRT challenge-password
<different> domain-name spoke_1.juniper.net email spoke_1@juniper.net subject
DC=juniper,CN=spoke_1.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
set security key-manager profiles SPOKE_1_KM_PROFILE_1 static key-id ascii-text test-key-id
set security key-manager profiles SPOKE_1_KM_PROFILE_1 static key ascii-text
qjwbdip139u5mcy89m28pcgowerefnkjsdg
```

```
set security ike proposal SPOKE_1_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_1_IKE_PROP dh-group group14
set security ike proposal SPOKE_1_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_1_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_1_IKE_PROP lifetime-seconds 3600
set security ike policy SPOKE_1_IKE_POL proposals SPOKE_1_IKE_PROP
set security ike policy SPOKE_1_IKE_POL certificate local-certificate SPOKE_1_CRT
set security ike gateway SPOKE_1_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_1_IKE_GW local-address 172.18.10.2
set security ike gateway SPOKE_1_IKE_GW ike-policy SPOKE_1_IKE_POL
set security ike gateway SPOKE_1_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_1_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW ppk-profile SPOKE_1_KM_PROFILE_1
set security ike gateway SPOKE_1_IKE_GW version v2-only
set security ipsec proposal SPOKE_1_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_1_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_1_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy SPOKE_1_IPSEC_POL proposals SPOKE_1_IPSEC_PROP
set security ipsec vpn SPOKE_1_IPSEC_VPN bind-interface st0.1
set security ipsec vpn SPOKE_1_IPSEC_VPN ike gateway SPOKE_1_IKE_GW
set security ipsec vpn SPOKE_1_IPSEC_VPN ike ipsec-policy SPOKE_1_IPSEC_POL
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.2/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.80.1/24
set interfaces st0 unit 1 family inet
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Set Commands on Spoke 2

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id SPOKE_2_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_2_CRT challenge-password
<different> domain-name spoke_2.juniper.net email spoke_2@juniper.net subject
DC=juniper,CN=spoke_2.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
set security key-manager profiles SPOKE_2_KM_PROFILE_1 static key-id ascii-text test-key-id
set security key-manager profiles SPOKE_2_KM_PROFILE_1 static key ascii-text
qjwbdip139u5mcy89m28pcgowerefnkjsdg
set security ike proposal SPOKE_2_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_2_IKE_PROP dh-group group14
set security ike proposal SPOKE_2_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_2_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_2_IKE_PROP lifetime-seconds 3600
set security ike policy SPOKE_2_IKE_POL proposals SPOKE_IKE_PROP
set security ike policy SPOKE_2_IKE_POL certificate local-certificate SPOKE_2_CRT
set security ike gateway SPOKE_2_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_2_IKE_GW local-address 172.18.10.3
set security ike gateway SPOKE_2_IKE_GW ike-policy SPOKE_2_IKE_POL
set security ike gateway SPOKE_2_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_2_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW ppk-profile SPOKE_2_KM_PROFILE_1
set security ike gateway SPOKE_2_IKE_GW version v2-only
set security ipsec proposal SPOKE_2_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_2_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_2_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy SPOKE_2_IPSEC_POL proposals SPOKE_2_IPSEC_PROP
set security ipsec vpn SPOKE_2_IPSEC_VPN bind-interface st0.2
set security ipsec vpn SPOKE_2_IPSEC_VPN ike gateway SPOKE_2_IKE_GW
set security ipsec vpn SPOKE_2_IPSEC_VPN ike ipsec-policy SPOKE_2_IPSEC_POL
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 local-ip 192.168.70.0/24
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.3/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.70.1/24
set interfaces st0 unit 2 family inet
```

```
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.2
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Appendix 2: Show Configuration Output on DUT

IN THIS SECTION

- Hub | **230**
- Spoke 1 | **234**
- Spoke 2 | **239**

Hub

From configuration mode, confirm your configuration by entering the `show security ike proposal HUB_IKE_PROP`, `show security ike policy HUB_IKE_POL`, `show security ike gateway HUB_IKE_GW`, `show security ipsec proposal HUB_IPSEC_PROP`, `show security ipsec policy HUB_IPSEC_POL`, `show security ipsec vpn HUB_IPSEC_VPN`, `show interfaces`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@hub# show security ike proposal HUB_IKE_PROP
dh-group group14;
authentication-algorithm sha-256;
```

```
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@hub# show security ike policy HUB_IKE_POL
proposals HUB_IKE_PROP;
certificate {
    local-certificate HUB_CRT;
```

```
user@hub# show security ike gateway HUB_IKE_GW
ike-policy HUB_IKE_POL;
dynamic {
    distinguished-name {
        wildcard C=us,DC=juniper;
    }
    ike-user-type group-ike-id;
}
local-identity distinguished-name;
external-interface ge-0/0/2.0;
local-address 172.18.10.1;
version v2-only;
ppk-profile HUB_KM_PROFILE_1;
```

```
user@hub# show security ipsec proposal HUB_IPSEC_PROP
protocol esp;
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
```

```
user@hub# show security ipsec policy HUB_IPSEC_POL
proposals HUB_IPSEC_PROP;
```

```
user@hub# show security ipsec vpn HUB_IPSEC_VPN
bind-interface st0.1;
ike {
    gateway HUB_IKE_GW;
    ipsec-policy HUB_IPSEC_POL;
}
```

```
traffic-selector ts1 {
    local-ip 192.168.90.0/24;
    remote-ip 0.0.0.0/0;
}
```

```
user@hub# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
        }
    }
}
ge-1/0/0 {
    unit 0 {
        family mpls;
    }
}
st0 {
    unit 1 {
        family inet;
```

```
    }
}
```

```
user@hub# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/0.0;
        ge-0/0/2.0;
    }
}
security-zone vpn {
    interfaces {
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

```
user@hub# show security policies
from-zone trust to-zone vpn {
    policy vpn_out {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
```

```
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn_in {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
```

Spoke 1

From configuration mode, confirm your configuration by entering the `show security pki ca-profile Root-CA`, `show security key-manager profiles SPOKE_1_KM_PROFILE_1`, `show security ike proposal SPOKE_1_IKE_PROP`, `show security ike policy SPOKE_1_IKE_POL`, `show security ike gateway SPOKE_1_IKE_GW`, `show security ipsec proposal SPOKE_1_IPSEC_PROP`, `show security ipsec policy SPOKE_1_IPSEC_POL`, `show security ipsec vpn SPOKE_1_IPSEC_VPN`, `show interfaces`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@spoke1# show security pki ca-profile Root-CA
ca-identity Root-CA;
enrollment {
    url https://ca-server.juniper.net/certsrv/mscep/mscep.dll;
}
revocation-check {
    disable;
}
```

```
user@spoke1# show security key-manager profiles SPOKE_1_KM_PROFILE_1
static {
    key-id ascii-text "$9$cJ5SvLdVYoZjs2qmTFAt1RhSMXoaZUjqWL"; ## SECRET-DATA
    key ascii-text "$9$mfF/IRSWX-9AORhyW8aZUj.PQFn/tuz3lKMXbwgoJGqf/
```

```
Ctu1RTzhSyeW8aZUHkPn6AIEyO1SeMWdVgoJUjqCA0IEyz3yKvW-d4aZ"; ## SECRET-DATA
}
```

```
user@spoke1# show security ike proposal SPOKE_1_IKE_PROP
authentication-method rsa-signatures;
dh-group group14;
authentication-algorithm sha-256;
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@spoke1# show security ike policy SPOKE_1_IKE_POL
proposals SPOKE_1_IKE_PROP;
certificate {
    local-certificate SPOKE_1_CRT;
}
```

```
user@spoke1# show security ike gateway SPOKE_1_IKE_GW
ike-policy SPOKE_1_IKE_POL;
address 172.18.10.1;
local-identity distinguished-name;
remote-identity distinguished-name;
external-interface ge-0/0/2.0;
local-address 172.18.10.2;
version v2-only;
ppk-profile SPOKE_1_KM_PROFILE_1;
```

```
user@spoke1# show security ipsec proposal SPOKE_1_IPSEC_PROP
protocol esp;
```

```
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
```

```
user@spoke1# show security ipsec policy SPOKE_1_IPSEC_POL
proposals SPOKE_1_IPSEC_PROP;
```

```
user@spoke1# show security ipsec vpn SPOKE_1_IPSEC_VPN
bind-interface st0.1;
ike {
    gateway SPOKE_1_IKE_GW;
    ipsec-policy SPOKE_1_IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.80.0/24;
    remote-ip 192.168.90.0/24;
}
```

```
user@spoke1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
```

```
            address 172.18.10.2/24;
        }
    }
}
ge-1/0/0 {
    unit 0 {
        family mpls;
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
```

```
user@spoke1# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
security-zone vpn {
    interfaces {
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
```

```
    }
}
```

```
user@spoke1# show security policies
from-zone trust to-zone vpn {
    policy vpn_out {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn_in {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
```

```
user@spoke1# show security pki
ca-profile Root-CA {
    ca-identity Root-CA;
    enrollment {
        url https://ca-server.juniper.net/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

Spoke 2

From configuration mode, confirm your configuration by entering the `show security pki`, `show security key-manager`, `show security ike proposal SPOKE_2_IKE_PROP`, `show security ike policy SPOKE_2_IKE_POL`, `show security ike gateway SPOKE_2_IKE_GW`, `show security ipsec proposal SPOKE_2_IPSEC_PROP`, `show security ipsec vpn SPOKE_2_IPSEC_VPN`, `show interfaces`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@spoke2# show security pki
ca-profile Root-CA {
    ca-identity Root-CA;
    enrollment {
        url https://ca-server.juniper.net/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

```
user@spoke2# show security key-manager
profiles {
    SPOKE_2_KM_PROFILE_1 {
        static {
            key-id ascii-text "$9$C4Y8ABEleWx-wM8goGjPf369A1hx7-VwgIE"; ## SECRET-DATA
            key ascii-text "$9$15SRyKdVYGjqvW7Vw2GUn/CtBIcylK8XSr4aZjPfz369ORKM8X-VhSwY2oGUn/
Cu0IleWdb27-YoZG.mz36CtOMWxdb2Sr2aJGq.Fn/"; ## SECRET-DATA
        }
    }
}
```

```
user@spoke2# show security ike proposal SPOKE_2_IKE_PROP
authentication-method rsa-signatures;
dh-group group14;
authentication-algorithm sha-256;
```

```
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@spoke2# show security ike policy SPOKE_2_IKE_POL
proposals SPOKE_IKE_PROP;
certificate {
    local-certificate SPOKE_2_CRT;
}
```

```
user@spoke2# show security ike gateway SPOKE_2_IKE_GW
ike-policy SPOKE_2_IKE_POL;
address 172.18.10.1;
local-identity distinguished-name;
remote-identity distinguished-name;
external-interface ge-0/0/2.0;
local-address 172.18.10.3;
version v2-only;
ppk-profile SPOKE_2_KM_PROFILE_1;
```

```
user@spoke2# show security ipsec proposal SPOKE_2_IPSEC_PROP
protocol esp;
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
```

```
user@spoke2# show security ipsec vpn SPOKE_2_IPSEC_VPN
bind-interface st0.2;
ike {
    gateway SPOKE_2_IKE_GW;
    ipsec-policy SPOKE_2_IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.70.0/24;
```

```
    remote-ip 192.168.90.0/24;
}
```

```
user@spoke2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
            address 192.168.70.1/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
            address 172.18.10.3/24;
        }
    }
}
ge-1/0/0 {
    unit 0 {
        family mpls;
    }
}
st0 {
    unit 1 {
        family inet;
    }
    unit 2 {
```

```
        family inet;
    }
}
```

```
user@spoke2# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
security-zone vpn {
    interfaces {
        st0.2;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

```
user@spoke2# show security policies
from-zone trust to-zone vpn {
    policy vpn_out {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
```

```
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn_in {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
```

Configure Quantum Key Manager Key Profile for Junos Key Manager

IN THIS SECTION

- Requirements | 243
- Overview | 244
- Configuration | 244
- Verification | 245

This example shows how to configure quantum key profile for Junos key manager. Configure the quantum key manager key profile to generate and send the generated keys to establish quantum safe IPsec VPN tunnel.

Requirements

Use Feature Explorer to confirm platform and release support for specific features.

1. Hardware requirements —Device that supports the feature.

2. Software requirements—OS with **Junos IKE** and **Junos Key Manager** packages that supports the feature..
3. Use any QKD device supporting ETSI Quantum Key Distribution (QKD) Rest API standard for communication.
4. Load the local certificates on the device. We recommended you to provide full path to the certificate.

Overview

The devices use the IPsec VPN to send traffic securely over the Internet. Configure the quantum key manager key profile in the IPsec VPN, to re-authenticate the existing IKE SA and a new key and key.

The quantum key manager key profile uses secure key distribution method based on QKD to generate and distribute keys that are quantum safe. These keys are dynamic.

Configuration

1. Configure the CA certificate.

```
user@host# set security pki ca-profile Root-CA ca-identity Root-CA
user@host# set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile Root-CA revocation-check disable
```

2. Load the CA certificate.

```
user@host> request security pki local-certificate load certificate-id SAE_A filename
SAE_A.cert key SAE_A.key
```

3. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile Root-CA
```

4. Configure the quantum key manager profile.

```
user@host# set security key-manager profiles KM_PROFILE_1 quantum-key-manager url https://
kme.juniper.net
user@host# set security key-manager profiles KM_PROFILE_1 quantum-key-manager local-sae-id
SAE_A
```

```
user@host# set security key-manager profiles KM_PROFILE_1 quantum-key-manager local-
certificate-id SAE_A_CERT
user@host# set security key-manager profiles KM_PROFILE_1 quantum-key-manager trusted-cas
Root-CA
```

Verification

Purpose

Verify the quantum key manager key profile and keys.

Action

From operational mode, enter the `show security pki ca-certificate ca-profile Root-CA` to view the CA profile and CA certificates.

```
user@host> show security pki ca-certificate ca-profile Root-CA
LSYS: root-logical-system
  CA profile: Root-CA
Certificate identifier: Root-CA
  Issued to: Root-CA, Issued by: C = IN, ST = WestBengal, O = JuniperNetworks, CN = Root-CA
  Validity:
    Not before: 09-11-2023 09:03 UTC
    Not after: 03-24-2044 09:03 UTC
  Public key algorithm: rsaEncryption(4096 bits)
  Keypair Location: Keypair generated locally
```

From operational mode, enter the `show security pki local-certificate certificate-id SAE_A_CERT` to view the PKI local certificates.

```
user@host> show security pki local-certificate certificate-id SAE_A_CERT
LSYS: root-logical-system
Certificate identifier: SAE_A_CERT
  Issued to: SAE_A, Issued by: C = IN, ST = WestBengal, O = JuniperNetworks, CN = ROOT_CA
  Validity:
    Not before: 08-28-2023 04:54 UTC
    Not after: 03-10-2044 04:54 UTC
  Public key algorithm: rsaEncryption(2048 bits)
  Keypair Location: Keypair generated locally
```

From operational mode, enter the `request security key-manager profiles get profile-keys name km_profile_1 peer-sae-id SAE_B` to view peer device key manager profile and keys.

```
user@host> request security key-manager profiles get profile-keys name km_profile_1 peer-sae-id
SAE_B

  - Response:
    - Status: SUCCESS
    - Name: km_profile_1
    - Type: quantum-key-manager
    - Key-size: 256 bits
    - Key-count: 1
    - Key-ids:
      - 002420bd-7a03-4725-9c41-6969d8e1815a
    - Keys:
      - 728d21c4a05fe2f73c7b2f58d1e3631dc68fcfaca16be12ca3fc7715079db0f9
```

From operational mode, enter the `show security key-manager profiles name KM_PROFILE_1 detail` to view key manager profile details.

```
user@host> show security key-manager profiles name KM_PROFILE_1 detail
Name: KM_PROFILE_1, Index: 2, Type: quantum-key-manager
  Configured-at: 11.09.23 (02:04:32)
  Time-elapsed: 0 hrs 20 mins 23 secs
  Url: https://kme.juniper.net
  Local-sae-id: SAE_A
  Local-certificate-id: SAE_A_CERT
  Trusted-cas: [ Root-CA ]
  Peer-sae-ids: N/A
  Default-key-size: N/A
  Request stats:
    Received: 0
    In-progress: 0
    Success: 0
    Failed: 0
```

Meaning

The `show security pki ca-certificate ca-profile Root-CA` displays PKI CA profile name, certificate identifier, validity, public key algorithm, and so on.

The `show security pki local-certificate certificate-id SAE_A_CERT` displays the local CA profile name, certificate identifier, validity, public key algorithm, and so on.

The `request security key-manager profiles get profile-keys name km_profile_1 peer-sae-id SAE_B` displays peer device key manager profile and keys.

The `show security key-manager profiles name KM_PROFILE_1 detail` displays the security key manager profile name, URL, requests, and so on.

Example: Configure Quantum Key Manager Key Profile for Site-To-Site IPsec VPN

SUMMARY

Use this configuration example to secure an IPsec Site-to-Site VPN infrastructure by configuring the quantum key manager key profile.

IN THIS SECTION

- Example Prerequisites | **248**
- Before You Begin | **248**
- Functional Overview | **249**
- Topology Overview | **253**
- Topology Illustration | **255**
- Step-By-Step Configuration on Devices | **255**
- Verification | **258**
- Appendix 1: Set Commands on all Devices | **263**
- Appendix 2: Show Configuration Output on DUT | **266**

You can secure an IPsec Site-to-Site VPN infrastructure by configuring the quantum key manager key profile.

In this configuration example, The SRX1 and SRX2 devices use the quantum key manager profile to fetch the QKD keys on IPsec VPN. The QKD keys help send traffic securely over the Internet.

TIP:

Table 45: Estimated Timers

Reading Time	Less than an hour
Configuration Time	Less than an hour

Example Prerequisites

Use Feature Explorer to confirm platform and release support for specific features.

Table 46: Hardware and Software Requirements

Hardware requirements	Device that supports the feature.
Software requirements	OS that supports the feature.

Before You Begin

Table 47: Benefits, Resources, and Additional Information

Benefits	• **Threat identification** By configuring quantum keys, you can establish a secure quantum channel between the QKD devices. This improves threat identification and secures the network. • **Extended security** You can merge the existing keys with quantum keys and encrypt and decrypt them over existing VPN tunnels. This improves the security of the IPsec VPN infrastructure. • **Enhanced cryptographic strength** RFC 8784 compliance provides you with an easy way to prevent attackers from eavesdropping on the connection and intercepting the keys. This also ensures interoperability with other devices that adhere to the standard. • **Interoperability support** You can use any QKD device that supports ETSI QKD Rest API.
Useful Resources	
Know more	• IPsec VPN • Route based IPsec VPN • Understanding Local Certificate Requests • Example: Loading CA and Local Certificates Manually
Hands-on experience	vLABs Sandbox
Learn more	RFC 8784 - Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security ETSI QKD Rest API

Functional Overview

Table 48: Quantum Key Manager Key Profile Functional Overview

IPsec VPN	Deploys a hub-and-spoke IPsec VPN topology where spokes are connected by VPN tunnels that send traffic through the hub. These VPN tunnels are later configured to use quantum keys making them quantum-safe VPN tunnels.
IKE gateway	Establishes a secure connection. The IKE gateway uses the IKE policy to limit itself to the configured group of certificate authority (CA) profiles while validating the certificate.
Proposals	
IKE proposal	Defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. IKE creates the dynamic security associations (SAs) and negotiates them for IPsec.
IPsec proposal	Lists protocols, algorithms, and security services to be negotiated with the remote IPsec peer.
Policies	
IKE policy	Defines a combination of security parameters (IKE proposals) to be used during IKE negotiation.
IPsec policy	Contains rules and security policies to allow group VPN traffic between the zones specified.

Security policy	Allows you to select the type of data traffic to secure through the IPsec SAs.
	• VPN-OUT—Permits traffic from the trust zone to the vpn zone, where the match criteria is:
	  • source-address: HOST-1-Net
	  • destination-address: HOST-2-Net
	  • application: any
	• VPN-IN—Permits traffic from the vpn zone to the trust zone, where the match criteria is:
	  • source-address: HOST-2-Net
	  • destination-address: HOST-1-Net
	  • application: any

Profiles	

Key profile	Defines how the devices communicate with the KME devices to retrieve QKD keys from the external KME server. Key profiles are configured on the hub (HUB_KM_PROFILE_1) and spokes (SPOKE_1_KM_PROFILE_1 and SPOKE_2_KM_PROFILE_1) separately. • Key profile—A quantum key manager profile *km_profile_1* is configured for applications and services to retrieve QKD keys from an external server. • IKE proposal—An IKE proposal *IKE_PROP* is configured with the required algorithms to establish an IKE SA. • IKE policy—An IKE policy *IKE_POL* is configured to set the runtime negotiation and authentication attributes. • IKE gateway—An IKE gateway *IKE_GW* is configured to manage the IPsec tunnels between endpoints. A *ppk-profile* indicates which key-profile to use to establish Quantum safe IKE or IPsec SA. • IPsec proposal—An IPsec proposal *IPSEC_PROP* is configured with the required algorithms to establish an IPsec SA. • IPsec policy—An IPsec policy *IPSEC_POL* is configured to set the runtime IPsec negotiation attributes. • IPsec VPN—An IPsec VPN policy *IPSEC_VPN* is configured to set the range of subnets that needs to be secured. • Security zone—Three different security zones *trust*, *untrust* and *vpn* are configured for better segregation of expected traffic within each of these zones. • Security policy—Security policies *trust to vpn* and *vpn to trust* are configured between the security zones to filter out which type of data traffic gets secured through the IPsec SAs.
PPK Profile	Indicates which key profile to use to establish quantum-safe IKE or IPsec SAs by referencing the key profile under the IKE gateway.
Certificates	

CA certificate	Verifies identity of devices and authenticate communication link.
Local certificate	Generates PKI and enroll it with the CA certificate for verification.
KME certificate	Third-party certificate generated by vendor.
Security Zones	
trust	Network segment at the host zone.
untrust	Network segment at the destination server zone.
vpn	Network segment through which the hub and spokes interact.
Primary verification tasks	Verify the established IKE and IPsec SAs are Quantum safe.

Topology Overview

In this example, we secure the SRX1 and SRX2 IPsec VPN tunnels by using quantum keys generated by third-party KME devices. The KME devices (KME-A and KME-B) are connected to each other through a quantum channel that is highly secure and capable of threat identification. Using this channel, the SRX1 and SRX2 devices retrieve quantum keys from their corresponding KME device and merge it with the existing keys to make the VPN tunnels quantum secure.

Table 49: Devices, Role, and Functionality used in this Configuration

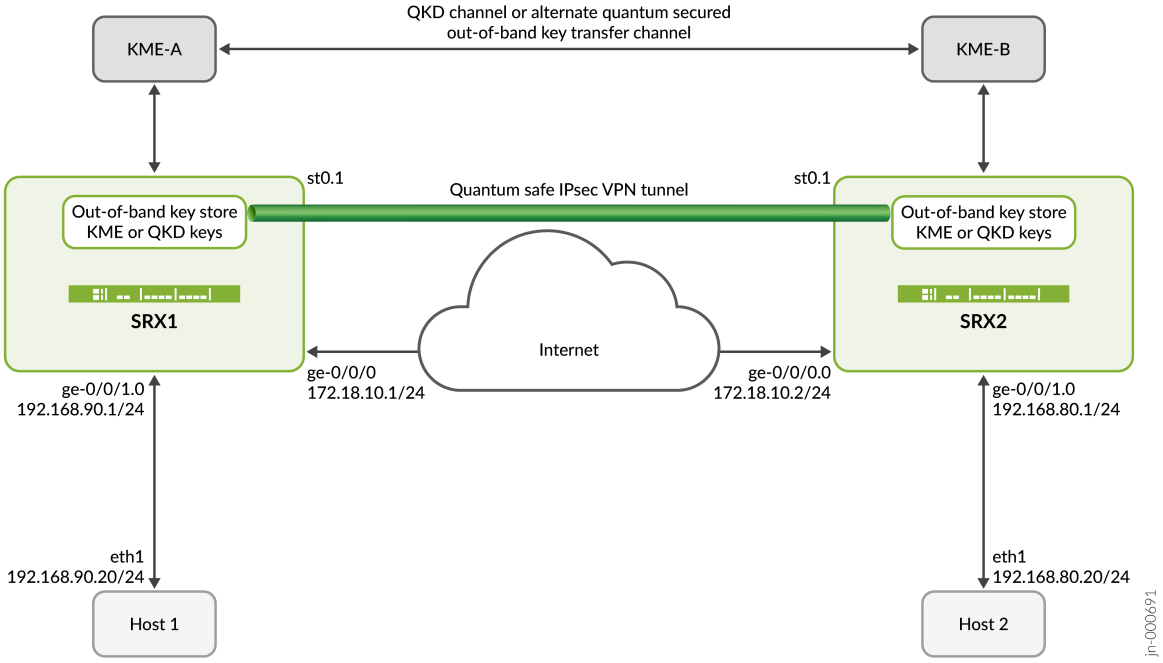
Hostname	Role	Function
SRX1	A device capable of establishing IPsec tunnels	Initiates IKE or IPsec SA negotiation and establishes quantum-safe IPsec tunnels with SRX2 using QKD key fetched from KME-A QKD device.

Table 49: Devices, Role, and Functionality used in this Configuration *(Continued)*

Hostname	Role	Function
SRX2	A device capable of establishing IPsec tunnels	Responds to IKE or IPsec SA negotiation and establishes quantum-safe IPsec tunnels using QKD key from KME-B QKD device.
HOST1	Host inside the trusted zone or LAN side of SRX1	Initiates client-side traffic toward HOST 2
HOST2	Host inside the trusted zone or LAN side of SRX2	Responds to client-side traffic from HOST 1.
KME-A	Third-party vendor QKD device	Provides QKD keys in response to key requests from SRX1.
KME-B	Third-party vendor QKD device	Provides QKD keys in response to key requests from SRX2.

Topology Illustration

Figure 21: Site-to-Site VPN



Step-By-Step Configuration on Devices

NOTE: For complete sample configurations on the DUT, see:

- "Set Commands on SRX1" on page 263
- "Set Commands on SRX2" on page 265

This configuration is applicable to SRX1 and SRX2 devices. For other devices, you must make the appropriate device-specific configuration changes.

1. Configure the interfaces.

```
[edit interfaces]
user@srx# set ge-0/0/0 unit 0 family inet address 172.18.10.1/24
user@srx# set st0 unit 1 family inet
user@srx# set ge-0/0/1 unit 0 family inet address 192.168.90.1/24
```

2. Configure a key profile of type quantum-key-manager with the must or recommended parameters.

   Define the CA certificate, configure the URL of the KME server, configure the SAE-ID to be used by the local end, configure the corresponding certificate for the local SAE-ID, and configure the previously defined CA certificate.

```
[edit security pki]
user@srx# set ca-profile ROOT_CA_CERT ca-identity RootCA
```

```
[edit security key-manager profiles]
user@srx# set km_profile_1 quantum-key-manager url https://www.kme_a-qkd-server.net
```

```
[edit security key-manager profiles]
user@srx# set km_profile_1 quantum-key-manager local-sae-id SAE_A
user@srx# set km_profile_1 quantum-key-manager local-certificate-id SAE_A_CERT
user@srx# set km_profile_1 quantum-key-manager trusted-cas ROOT_CA_CERT
```

3. Configure Site-to-Site IPsec VPN. This includes configuring the security zones, security policies, and relevant certificates for authenticating device identities and their communication links.

```
[edit security zones]
user@srx# set security-zone untrust host-inbound-traffic system-services ike
user@srx# set security-zone untrust interfaces ge-0/0/0.0
user@srx# set security-zone vpn interfaces st0.1
user@srx# set security-zone trust host-inbound-traffic system-services ping
user@srx# set security-zone trust interfaces ge-0/0/1.0
```

```
[edit security policies]
user@srx# set from-zone trust to-zone vpn policy vpn_out match source-address any
user@srx# set from-zone trust to-zone vpn policy vpn_out match destination-address any
user@srx# set from-zone trust to-zone vpn policy vpn_out match application any
user@srx# set from-zone trust to-zone vpn policy vpn_out then permit
user@srx# set from-zone vpn to-zone trust policy vpn_in match source-address any
user@srx# set from-zone vpn to-zone trust policy vpn_in match destination-address any
```

```
user@srx# set from-zone vpn to-zone trust policy vpn_in match application any
user@srx# set from-zone vpn to-zone trust policy vpn_in then permit
```

```
[edit security ike proposal]
user@srx# set IKE_PROP authentication-method pre-shared-keys
user@srx# set IKE_PROP dh-group group14
user@srx# set IKE_PROP authentication-algorithm sha-256
user@srx# set IKE_PROP encryption-algorithm aes-256-cbc
user@srx# set IKE_PROP lifetime-seconds 3600
```

```
[edit security ike policy]
user@srx# set IKE_POL proposals IKE_PROP
user@srx# set IKE_POL pre-shared-key ascii-text ipsec-test
```

```
[edit security ike gateway]
user@srx# set IKE_GW ike-policy IKE_POL
user@srx# set IKE_GW address 172.18.10.2
user@srx# set IKE_GW external-interface ge-0/0/0.0
user@srx# set IKE_GW local-address 172.18.10.1
user@srx# set IKE_GW version v2-only
user@srx# set IKE_GW ppk-profile km_profile_1
```

```
[edit security ipsec proposal]
user@srx# set IPSEC_PROP protocol esp
user@srx# set IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@srx# set IPSEC_PROP encryption-algorithm aes-256-cbc
user@srx# set IPSEC_PROP lifetime-seconds 2400
```

```
[edit security ipsec policy]
user@srx# set IPSEC_POL proposals IPSEC_PROP
```

```
[edit security ipsec vpn]
user@srx# set IPSEC_VPN bind-interface st0.1
user@srx# set IPSEC_VPN ike gateway IKE_GW
user@srx# set IPSEC_VPN ike ipsec-policy IPSEC_POL
```

```
user@srx# set IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
user@srx# set IPSEC_VPN traffic-selector ts1 remote-ip 192.168.80.0/24
user@srx# set IPSEC_VPN establish-tunnels immediately
```

Verification

IN THIS SECTION

- Verify IKE SAs | **259**
- Verify IPsec SAs | **260**
- Verify IPsec Statistics | **261**
- Verify Key Manager Profile | **262**
- Ping from HOST 1 to HOST 2 or vice versa | **263**

This section provides a list of show commands that you can use to verify the feature in this example.

Command	Verification Task
show security ike security-associations detail	"Verify that the IKE SAs are established." on page 259
show security ipsec security-associations detail	"Verify that the IPsec SAs are established." on page 260
show security ipsec statistics	"Verify IPsec encryption and decryption statistics." on page 261
show security key-manager profiles detail	"Verify key profile statistics." on page 262
ping 192.168.80.20 source 192.168.90.20 count 5	"Ping from HOST 1 to HOST 2 or vice versa." on page 263

Verify IKE SAs

Purpose

Verify the IKE SAs.

Action

From operational mode, enter the `show security ike security-associations detail` command to view the IKE SAs.

```
user@srx> show security ike security-associations detail
IKE peer 172.18.10.2, Index 21, Gateway Name: IKE_GW
Role: Initiator, State: UP
Initiator cookie: 5a417d46cef3207d, Responder cookie: 57b9a17516bee31b Exchange type: IKEv2,
Authentication method: Pre-shared-keys
Local gateway interface: ge-0/0/2.0 Routing instance: default
Local: 172.18.10.1:500, Remote: 172.18.10.2:500
Lifetime: Expires in 3445 seconds Reauth Lifetime: Disabled
IKE Fragmentation: Enabled, Size: 576 SRG ID: 0
Remote Access Client Info: Unknown Client Peer ike-id: 172.18.10.2
AAA assigned IP: 0.0.0.0 PPK-profile: km_profile_1
Optional: No State    : Used
Algorithms:
Authentication    : hmac-sha256-128
Encryption    : aes256-cbc Pseudo random function: hmac-sha256 Diffie-Hellman group : DH-group-14
Traffic statistics:
Input bytes :    783
Output bytes :    831
Input packets:        2
Output packets:        2
Input fragmented packets:    0
Output fragmented packets:    0
IPSec security associations: 2 created, 0 deleted Phase 2 negotiations in progress: 1
IPSec Tunnel IDs: 500003
Negotiation type: Quick mode, Role: Initiator, Message ID: 0 Local: 172.18.10.1:500, Remote:
172.18.10.2:500
Local identity: 172.18.10.1
Remote identity: 172.18.10.2 Flags: IKE SA is created
IPsec SA Rekey CREATE_CHILD_SA exchange stats:
Initiator stats:        Responder stats:
Request Out    : 0    Request In    : 0
```

```
Response In    : 0    Response Out    : 0
No Proposal Chosen In    : 0    No Proposal Chosen Out    : 0
Invalid KE In    : 0    Invalid KE Out    : 0
TS Unacceptable In    : 0    TS Unacceptable Out    : 0
Res DH Compute Key Fail : 0    Res DH Compute Key Fail: 0 Res Verify SA Fail    : 0
Res Verify DH Group Fail: 0 Res Verify TS Fail    : 0
```

Meaning

The `Role: Initiator, State: UP`, `PPK-profile: km_profile_1`, `IPSec security associations: 2 created, 0 deleted Phase 2 negotiations in progress: 1`, and `Flags: IKE SA is created` fields shows the IKE SAs are created successfully.

Verify IPsec SAs

Purpose

Verify the IPsec SAs.

Action

From operational mode, enter the `show security ipsec security-associations detail` command to view the IPsec SAs.

```
user@srx> show security ipsec security-associations detail

ID: 500003 Virtual-system: root, VPN Name: IPSEC_VPN Local Gateway: 172.18.10.1, Remote Gateway:
172.18.10.2 Traffic Selector Name: ts1
Local Identity: ipv4(192.168.90.0-192.168.90.255)
Remote Identity: ipv4(192.168.80.0-192.168.80.255) TS Type: traffic-selector
Version: IKEv2 Quantum Secured: Yes PFS group: N/A
SRG ID: 0
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: IPSEC_POL Port:
500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
Multi-sa, Configured SAs# 0, Negotiated SAs#: 0 Tunnel events:
Fri Mar 31 2023 01:41:52: IPsec SA negotiation succeeds (1 times)
Location: FPC 0, PIC 0, KMD-Instance 0
Anchorship: Thread 1
Distribution-Profile: default-profile Direction: inbound, SPI: 0xd1e1549c, AUX-SPI: 0
, VPN Monitoring: - Hard lifetime: Expires in 1916 seconds Lifesize Remaining: Unlimited
Soft lifetime: Expires in 1349 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
```

```
Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits) Anti-replay
service: counter-based enabled, Replay window size: 64
Extended-Sequence-Number: Disabled
tunnel-establishment: establish-tunnels-immediately IKE SA Index: 21
Direction: outbound, SPI: 0xb5883167, AUX-SPI: 0
, VPN Monitoring: - Hard lifetime: Expires in 1916 seconds Lifesize Remaining: Unlimited
Soft lifetime: Expires in 1349 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits) Anti-replay
service: counter-based enabled, Replay window size: 64
Extended-Sequence-Number: Disabled
tunnel-establishment: establish-tunnels-immediately IKE SA Index: 21
```

Meaning

The `Quantum Secured: Yes`, `Policy-name: IPSEC_POL`, `IPsec SA negotiation succeeds (1 times)`, and `tunnel-establishment: establish-tunnels-immediately IKE SA Index: 21` fields shows the IPsec SAs are created successfully.

Verify IPsec Statistics

Purpose

Verify the IPsec statistics.

Action

From operational mode, enter the `show security ipsec statistics` command to view the IPsec statistics.

```
user@srx> show security ipsec statistics
ESP Statistics:
Encrypted bytes:    780
Decrypted bytes:    780
Encrypted packets:    5
Decrypted packets:    5
AH Statistics:
Input bytes:    0
Output bytes:    0
Input packets:    0
Output packets:    0
Errors:
```

```
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0 Bad headers: 0, Bad trailers: 0
Invalid SPI: 0, TS check fail: 0 Exceeds tunnel MTU: 0
Discarded: 0
```

Meaning

The `ESP Statistics` and `AH Statistics` fields shows the IPsec statistics.

Verify Key Manager Profile

Purpose

Verify the key manager profile.

Action

From operational mode, enter the `show security key-manager profiles detail` command to view the key manager profile.

```
user@srx> show security key-manager profiles detail
Name: km_profile_1, Index: 3, Type: Quantum-key-manager
Configured-at: 31.03.23 (01:40:50)
Time-elapsed: 0 hrs 11 mins 30 secs Url: https://www.kme_a-qkd-server.net Local-sae-id: SAE_A
Local-certificate-id: SAE_A_CERT Trusted-cas: [ ROOT_CA_CERT ] Peer-sae-ids: N/A
Default-key-size: N/A Request stats:
Received: 1
In-progress: 0
Success: 1
Failed: 0
```

Meaning

The `Name: km_profile_1` and `Quantum-key-manager` fields shows the key manager profile.

Ping from HOST 1 to HOST 2 or vice versa

Purpose

Verify the connectivity from HOST 1 to HOST 2.

Action

From operational mode, enter the `ping 192.168.80.20 source 192.168.90.20` count 5 to view the connectivity from HOST 1 to HOST 2.

```
user@HOST1# ping 192.168.80.20 source 192.168.90.20 count 5
PING 192.168.80.20 (192.168.80.20): 56 data bytes count 5
64 bytes from 192.168.80.1: icmp_seq=0 ttl=64 time=0.998 ms
64 bytes from 192.168.80.1: icmp_seq=1 ttl=64 time=1.594 ms
64 bytes from 192.168.80.1: icmp_seq=2 ttl=64 time=1.395 ms
64 bytes from 192.168.80.1: icmp_seq=3 ttl=64 time=1.536 ms
64 bytes from 192.168.80.1: icmp_seq=4 ttl=64 time=1.838 ms

--- 192.168.80.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss round-trip min/avg/max/stddev =
0.998/1.472/1.838/0.277 ms

Data traffic is successfully flowing between the HOSTs
```

Meaning

The `PING 192.168.80.20 (192.168.80.20): 56 data bytes count 5` confirms the connectivity from HOST 1 to HOST 2.

Appendix 1: Set Commands on all Devices

Set command output on all devices.

Set Commands on SRX1

```
set security pki ca-profile ROOT_CA_CERT ca-identity RootCA
set security key-manager profiles km_profile_1 quantum-key-manager url https://www.kme_a-qkd-
server.net
set security key-manager profiles km_profile_1 quantum-key-manager local-sae-id SAE_A
```

```
set security key-manager profiles km_profile_1 quantum-key-manager local-certificate-id
SAE_A_CERT
set security key-manager profiles km_profile_1 quantum-key-manager trusted-cas ROOT_CA_CERT
set interfaces ge-0/0/0 unit 0 family inet address 172.18.10.1/24
set interfaces st0 unit 1 family inet
set interfaces ge-0/0/1 unit 0 family inet address 192.168.90.1/24
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group14
set security ike proposal IKE_PROP authentication-algorithm sha-256
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text ipsec-test
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 172.18.10.2
set security ike gateway IKE_GW external-interface ge-0/0/0.0
set security ike gateway IKE_GW local-address 172.18.10.1
set security ike gateway IKE_GW version v2-only
set security ike gateway IKE_GW ppk-profile km_profile_1
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal IPSEC_PROP lifetime-seconds 2400
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN bind-interface st0.1
set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
```

```
set security ipsec vpn IPSEC_VPN traffic-selector ts1 remote-ip 192.168.80.0/24
set security ipsec vpn IPSEC_VPN establish-tunnels immediately
```

Set Commands on SRX2

```
set security pki ca-profile ROOT_CA_CERT ca-identity RootCA
set security key-manager profiles km_profile_1 quantum-key-manager url https://www.kme_a-qkd-
server.net
set security key-manager profiles km_profile_1 quantum-key-manager local-sae-id SAE_B
set security key-manager profiles km_profile_1 quantum-key-manager local-certificate-id
SAE_B_CERT
set security key-manager profiles km_profile_1 quantum-key-manager trusted-cas ROOT_CA_CERT
set interfaces ge-0/0/0 unit 0 family inet address 172.18.10.2/24
set interfaces st0 unit 1 family inet
set interfaces ge-0/0/1 unit 0 family inet address 192.168.80.1/24
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group14
set security ike proposal IKE_PROP authentication-algorithm sha-256
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text ipsec-test
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 172.18.10.1
set security ike gateway IKE_GW external-interface ge-0/0/0.0
set security ike gateway IKE_GW local-address 172.18.10.2
set security ike gateway IKE_GW version v2-only
set security ike gateway IKE_GW ppk-profile km_profile_1
```

```
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal IPSEC_PROP lifetime-seconds 2400
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN bind-interface st0.1
set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
set security ipsec vpn IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set security ipsec vpn IPSEC_VPN establish-tunnels immediately
```

Appendix 2: Show Configuration Output on DUT

IN THIS SECTION

- SRX1 | 266
- SRX 2 | 270

Show command output on the DUT.

SRX1

```
user@srk1# show security pki
ca-profile ROOT_CA_CERT {
    ca-identity RootCA;
}
```

```
user@srk1# show security key-manager
profiles {
    km_profile_1 {
        quantum-key-manager {
            url https://www.kme_a-qkd-server.net;
            local-sae-id SAE_A;
            local-certificate-id SAE_A_CERT;
            trusted-cas ROOT_CA_CERT;
        }
```

```
    }
}
```

```
user@srk1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
            address 192.168.70.1/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
            address 172.18.10.3/24;
        }
    }
}
ge-1/0/0 {
    unit 0 {
        family mpls;
    }
}
st0 {
    unit 1 {
        family inet;
    }
    unit 2 {
```

```
        family inet;
    }
}
```

```
user@srk1# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn {
    interfaces {
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

```
user@srk1# show security policies
from-zone trust to-zone vpn {
    policy vpn_out {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
```

```
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn_in {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
```

```
user@srk1# show security ike proposal IKE_PROP
authentication-method pre-shared-keys;
dh-group group14;
authentication-algorithm sha-256;
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@srk1# show security ike policy IKE_POL
proposals IKE_PROP;
pre-shared-key ascii-text "$9$Nadwg4aUH.5Nds4aUiHuO1RhrvWxVs4"; ## SECRET-DATA
```

```
user@srk1# show security ike gateway IKE_GW
ike-policy IKE_POL;
address 172.18.10.2;
external-interface ge-0/0/0.0;
local-address 172.18.10.1;
```

```
version v2-only;
ppk-profile km_profile_1;
```

```
user@srk1# show security ipsec proposal IPSEC_PROP
protocol esp;
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
lifetime-seconds 2400;
```

```
user@srk1# show security ipsec policy IPSEC_POL
proposals IPSEC_PROP;
```

```
user@srk1# show security ipsec vpn IPSEC_VPN
bind-interface st0.1;
ike {
    gateway IKE_GW;
    ipsec-policy IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.90.0/24;
    remote-ip 192.168.80.0/24;
}
establish-tunnels immediately;
```

SRX 2

```
user@srx2# show security key-manager
profiles {
    km_profile_1 {
        quantum-key-manager {
            url https://www.kme_a-qkd-server.net;
            local-sae-id SAE_B;
            local-certificate-id SAE_B_CERT;
            trusted-cas ROOT_CA_CERT;
        }
```

```
    }
}
```

```
user@srx2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
            address 192.168.70.1/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
            address 172.18.10.3/24;
        }
    }
}
ge-1/0/0 {
    unit 0 {
        family mpls;
    }
}
st0 {
    unit 1 {
        family inet;
    }
    unit 2 {
```

```
        family inet;
    }
}
```

```
user@srx2# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn {
    interfaces {
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

```
user@srx2# show security policies
from-zone trust to-zone vpn {
    policy vpn_out {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
```

```
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn_in {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
```

```
user@srx2# show security ike proposal IKE_PROP
authentication-method pre-shared-keys;
dh-group group14;
authentication-algorithm sha-256;
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@srx2# show security ike gateway IKE_GW
ike-policy IKE_POL;
address 172.18.10.1;
external-interface ge-0/0/0.0;
local-address 172.18.10.2;
```

```
version v2-only;
ppk-profile km_profile_1;
```

```
user@srx2# show security ike policy IKE_POL
proposals IKE_PROP;
pre-shared-key ascii-text "$9$P5z6/Cu1EyP5F/CuB1-VwYgJDi.TF/"; ## SECRET-DATA
```

```
user@srx2# show security ipsec policy IPSEC_POL
proposals IPSEC_PROP;
```

```
user@srx2# show security ipsec vpn IPSEC_VPN
bind-interface st0.1;
ike {
    gateway IKE_GW;
    ipsec-policy IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.80.0/24;
    remote-ip 192.168.90.0/24;
}
establish-tunnels immediately;
```

Example: Configure Quantum-Secured IPsec AutoVPN Topology Using Quantum Key Manager Key Profile

SUMMARY

Use this configuration example to secure an IPsec AutoVPN infrastructure by configuring the quantum key manager key profile.

IN THIS SECTION

- Example Prerequisites | **275**
- Before You Begin | **276**
- Functional Overview | **276**
- Topology Overview | **281**

- Topology Illustration | **284**
- Step-By-Step Configuration on Hub | **284**
- Step-By-Step Configuration on Spoke Devices | **288**
- Verification | **291**
- Appendix 1: Set Commands on all Devices | **301**
- Appendix 2: Show Configuration Output on DUT | **305**

The Hub, Spoke 1, and Spoke 2 use quantum key manager key profiles to communicate with KME Hub, KME Spoke 1, and KME Spoke 2 to fetch the QKD keys and establish then IPsec VPN tunnels.

TIP:

Table 50: Estimated Timers

Reading Time	Less than an hour.
Configuration Time	Less than an hour.

Example Prerequisites

Use Feature Explorer to confirm platform and release support for specific features.

Table 51: Hardware and Software Requirements

Hardware requirements	• Device that supports the feature. • Third-party Key Management Entity (KME) or Quantum Key Distribution (QKD) devices. The KME parameters are as per ETSI GS QKD 014 specification.
Software requirements	OS that supports the feature.

Before You Begin

Table 52: Benefits, Resources, and Additional Information

Benefits	• **Threat identification** Establish a secure quantum channel between the QKD devices that guarantees threat identification with the help of quantum keys. • **Extend security** Merge existing keys with quantum keys and encrypt and decrypt them over existing VPN tunnels thereby extending security of the IPsec VPN infrastructure. • **RFC 8784 compliant** Extend the already standardized RFC 8784 procedure. • **Interoperability support** Use any QKD device supporting ETSI QKD Rest API.
Useful Resources	
Know more	• IPsec VPN • AutoVPN on Hub-and-Spoke Devices
Hands-on Experience	vLab Sandbox: IPsec VPN
Learn more	• RFC 8784 - Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security • ETSI QKD Rest API

Functional Overview

Table 53 on page 276 provides a quick summary of the configuration components deployed in this example.

Table 53: Quantum Key Manager Functional Overview

IPsec VPN	Deploy a hub-and-spoke IPsec VPN topology where spokes are connected by VPN tunnels that send traffic through the hub. These VPN tunnels are later configured to use quantum keys making them quantum-safe VPN tunnels.
IKE gateway	Establish a secure connection, the IKE gateway uses the IKE policy to limit itself to the configured group of CAs (ca-profiles) while validating the certificate.
Proposals	
IKE proposal	Define the algorithms and keys used to establish the secure IKE connection with the peer security gateway. IKE creates the dynamic SAs and negotiates them for IPsec.
IPsec proposal	List protocols, algorithms, and security services to be negotiated with the remote IPsec peer.
Policies	
IKE policy	Define a combination of security parameters (IKE proposals) to be used during IKE negotiation.
IPsec policy	Contain rules and security policies to allow group VPN traffic between the zones specified.

Security policy	Allows you to select the type of data traffic to secure through the IPsec SAs.
	• VPN-OUT – Permits traffic from the trust zone to the vpn zone, where the match criteria is:
	  • source-address: HOST-1-Net
	  • destination-address: HOST-2-Net
	  • application: any
	• VPN-IN – Permits traffic from the vpn zone to the trust zone, where the match criteria is:
	  • source-address: HOST-2-Net
	  • destination-address: HOST-1-Net
	  • application: any

Profiles	

Key profile	Define how the SRX devices communicate with the KME devices to retrieve QKD keys from the external KME server. Key profiles are configured on the hub (HUB_KM_PROFILE_1) and spokes (SPOKE_1_KM_PROFILE_1 and SPOKE_2_KM_PROFILE_1) separately.
	
	Configure *SPOKE-1* and *SPOKE-2* for applications and services to retrieve QKD keys from external server.
	
	• Key profile—Configure the following quantum key manager key profiles on the Hub.
	  • *HUB_KM_PROFILE_1*
	  • *SPOKE_1_KM_PROFILE_1*
	  • *SPOKE_2_KM_PROFILE_1*
	• Configure *SPOKE-1* and *SPOKE-2* with the required algorithms to establish an IKE SAs.
	  IKE proposal—Configure the following IKE proposals on the Hub.
	  • *HUB_IKE_PROP*
	  • *SPOKE_1_IKE_PROP*
	  • *SPOKE_2_IKE_PROP*
	• Configure *SPOKE-1* and *SPOKE-2* to set the runtime negotiation and authentication attributes.
	  IKE policy—Configure the following IKE policies on the Hub.
	  • *HUB_IKE_POL*
	  • *SPOKE_1_IKE_POL*
	  • *SPOKE_3_IKE_POL*
	• Configure *SPOKE-1* and *SPOKE-2* to set the endpoints between the IPsec tunnels.
	  IKE gateway—Configure the following IKE gateways on the Hub.

	A *ppk-profile* indicates which key-profile to use to establish quantum-safe IKE or IPsec SA. • *HUB_IKE_GW* • *SPOKE_1_IKE_GW* • *SPOKE_2_IKE_GW* • Configure *SPOKE-1* and *SPOKE-2* with the required algorithms to establish an IPsec SA. IPsec proposal—Configure the following IPsec proposals on the Hub. • *HUB_IPSEC_PROP* • *SPOKE_1_IPSEC_PROP* • *SPOKE_2_IPSEC_PROP* • Configure *SPOKE-1* and *SPOKE-2* to set the runtime IPsec negotiation attributes. IPsec policy—Configure the following IPsec policies on the Hub. • *HUB_IPSEC_POL* • *SPOKE_1_IPSEC_POL* • *SPOKE_2_IPSEC_POL* • Configure *SPOKE-1* and *SPOKE-2* to set the range of subnets that need to be secured. IPsec VPN—Configure the following IPsec VPNs on the Hub. • *HUB_IPSEC_VPN* • *SPOKE_1_IPSEC_VPN* • *SPOKE_2_IPSEC_VPN* • Security zone—Configure three different security zones to segregate the traffic.

	• *trust* • *untrust* • *vpn* • Security policy—Configure the security policies *trust to vpn* and *vpn to trust* to select the type of data traffic that is secured through the IPsec SAs.
PPK Profile	Indicate which key profile to use to establish quantum-safe IKE or IPsec SAs by referencing the key profile under the IKE gateway.
Certificates	
CA certificate	Verify identity of devices and authenticate communication link between them.
Local certificate	Generate PKI and enroll it with the CA certificate for verification.
KME certificate	Third-party certificate generated by vendor.
Security Zones	
trust	Network segment at the host zone.
untrust	Network segment at the destination server zone.
vpn	Network segment through which the hub and spokes interact.
Primary verification tasks	Verify the established IKE and IPsec SAs are Quantum safe.

Topology Overview

In this example, we secure the hub-and-spoke IPsec VPN tunnels using quantum keys generated by third-party KME devices. The KME devices (KME-Hub, KME-Spoke 1, and KME-Spoke 2) are connected to each other through a quantum channel that is highly secure and capable of threat identification. Using

this channel, the Hub and Spoke devices retrieve quantum keys from their corresponding KME device and merge it with the existing keys to make the VPN tunnels quantum secure.

Table 54: Quantum Key Manager Topology Components

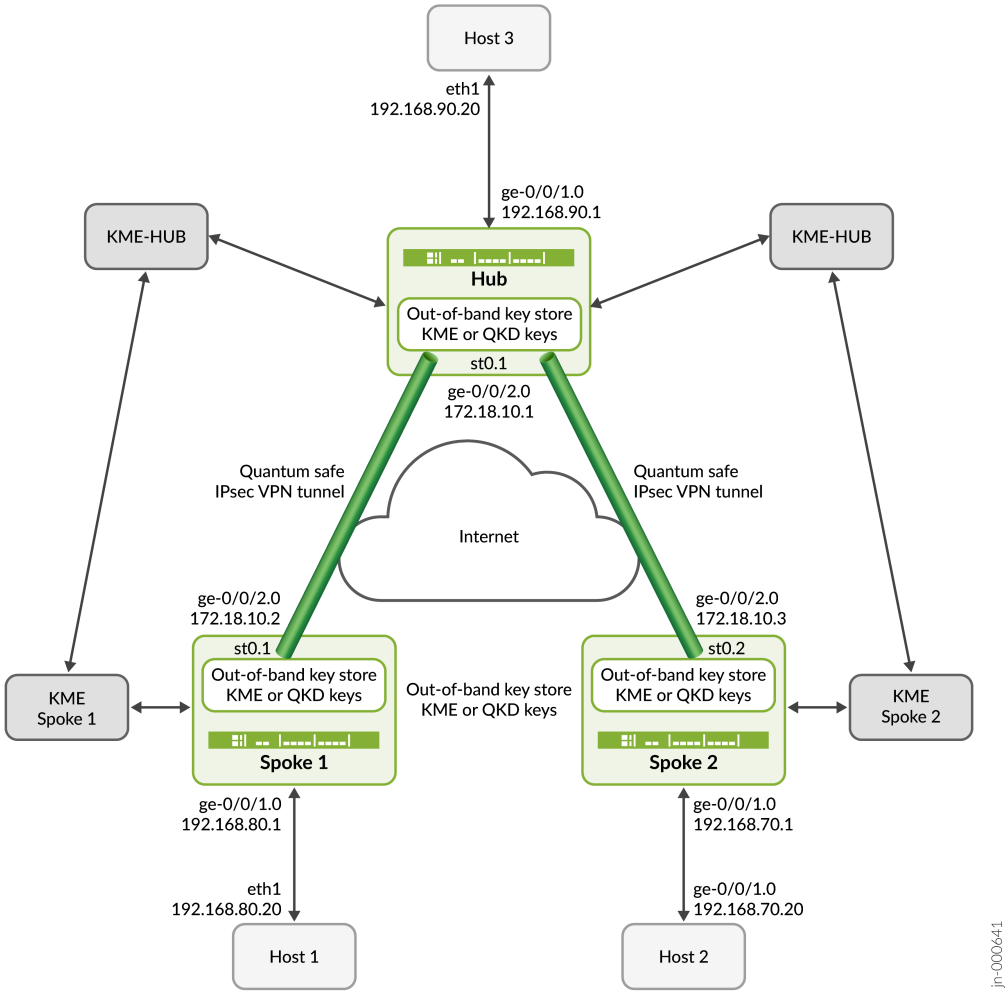
Topology Components	Role	Function
Hub	A device capable of establishing IPsec tunnels	Responds to IKE or IPsec SA negotiation and establishes Quantum-safe IPsec tunnels using QKD key from KME-HUB QKD device on SPOKE-1 and SPOKE-2.
SPOKE-1	A device capable of establishing IPsec tunnels	Initiates IKE or IPsec SA negotiation and establishes Quantum-safe IPsec tunnels with hub using QKD key from KME-SPOKE-1 QKD device.
SPOKE-2	A device capable of establishing IPsec tunnels	Initiates IKE or IPsec SA negotiation and establishes Quantum-safe IPsec tunnels with hub using QKD key from KME-SPOKE-2 QKD device.
HOST-1	Host inside the trusted zone or LAN side of SPOKE 1. Host 1 is secured by SPOKE 1.	Initiates client-side traffic toward HOST-3
HOST-2	Host inside the trusted zone or LAN side of SPOKE 2. Host 2 is secured by SPOKE 2.	Initiates client-side traffic toward HOST-3
HOST- 3	Host inside the trusted zone or LAN side of hub. Host 3 is secured by Hub.	Responds to client-side traffic from HOST-1 and HOST-2
KME-HUB	Third-party QKD device	Provides QKD keys in response to key requests from HUB

Table 54: Quantum Key Manager Topology Components *(Continued)*

Topology Components	Role	Function
KME-SPOKE-1	Third-party QKD device	Provides QKD keys in response to key requests from SPOKE-1
KME-SPOKE-2	Third-party QKD device	Provides QKD keys in response to key requests from SPOKE-2

Topology Illustration

Figure 22: Quantum Key Manager with AutoVPN



Step-By-Step Configuration on Hub

NOTE: For complete sample configurations on the hub and spoke devices, see:

- "Appendix 1: Set Commands on all Devices" on page 301
- "Appendix 2: Show Configuration Output on DUT" on page 305

1. Configure the hub interfaces.

```
[edit interfaces]
user@hub# set ge-0/0/2 unit 0 family inet address 172.18.10.1/24
user@hub# set ge-0/0/1 unit 0 family inet address 192.168.90.1/24
user@hub# set st0 unit 1 family inet
```

2. Configure hub-spoke the IPsec VPN. This includes configuring the security zones, security policies, and relevant certificates for authenticating device identities and their communication links.

   Configure the hub to fetch the CA certificate from the CA server, or load a locally available CA certificate from the device.

   NOTE: The KME certificates need to configured as per third-party vendor instructions.

   Configure the IPsec proposal and policy. Configure the IKE policy, proposal and gateway for the IPsec VPN.

```
[edit security zones]
user@hub# set security-zone untrust host-inbound-traffic system-services ike
user@hub# set security-zone untrust interfaces ge-0/0/2.0
user@hub# set security-zone vpn interfaces st0.1
user@hub# set security-zone trust host-inbound-traffic system-services ping
user@hub# set security-zone trust interfaces ge-0/0/1.0
```

```
[edit security policies]
user@hub# set from-zone trust to-zone vpn policy vpn_out match source-address any
user@hub# set from-zone trust to-zone vpn policy vpn_out match destination-address any
user@hub# set from-zone trust to-zone vpn policy vpn_out match application any
user@hub# set from-zone trust to-zone vpn policy vpn_out then permit
user@hub# set from-zone vpn to-zone trust policy vpn_in match source-address any
user@hub# set from-zone vpn to-zone trust policy vpn_in match destination-address any
user@hub# set from-zone vpn to-zone trust policy vpn_in match application any
user@hub# set from-zone vpn to-zone trust policy vpn_in then permit
```

```
[edit security pki]
user@hub# set ca-profile Root-CA ca-identity Root-CA
```

```
user@hub# set ca-profile Root-CA enrollment url url-to-CA-server
user@hub# set ca-profile Root-CA revocation-check disable
```

```
user@hub> request security pki ca-certificate enroll ca-profile Root-CA
```

```
user@hub> request security pki generate-key-pair certificate-id HUB_CRT size 2048 type rsa
user@hub> request security pki local-certificate enroll certificate-id HUB_CRT challenge-
password password domain-name hub.juniper.net email hub@juniper.net subject
DC=juniper,CN=hub.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-profile
Root-CA
user@hub> request security pki local-certificate load certificate-id SAE_HUB filename
SAE_HUB.cert key SAE_HUB.key
```

```
[edit security ike proposal]
user@hub# set HUB_IKE_PROP authentication-method rsa-signatures
user@hub# set HUB_IKE_PROP dh-group group14
user@hub# set HUB_IKE_PROP authentication-algorithm sha-256
user@hub# set HUB_IKE_PROP encryption-algorithm aes-256-cbc
user@hub# set HUB_IKE_PROP lifetime-seconds 3600
```

```
[edit security ike policy]
user@hub# set HUB_IKE_POL proposals HUB_IKE_PROP
user@hub# set HUB_IKE_POL certificate local-certificate HUB_CRT
```

```
[edit security ike gateway]
user@hub# set HUB_IKE_GW local-address 172.18.10.1
user@hub# set HUB_IKE_GW ike-policy HUB_IKE_POL
user@hub# set HUB_IKE_GW external-interface ge-0/0/2.0
user@hub# set HUB_IKE_GW local-identity distinguished-name
user@hub# set HUB_IKE_GW dynamic ike-user-type group-ike-id
```

```
user@hub# set HUB_IKE_GW dynamic distinguished-name wildcard C=us,DC=juniper
user@hub# set HUB_IKE_GW version v2-only
```

```
[edit security ipsec proposal]
user@hub# set HUB_IPSEC_PROP protocol esp
user@hub# set HUB_IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@hub# set HUB_IPSEC_PROP encryption-algorithm aes-256-cbc
```

```
[edit security ipsec vpn]
user@hub# set HUB_IPSEC_VPN bind-interface st0.1
user@hub# set HUB_IPSEC_VPN ike gateway HUB_IKE_GW
user@hub# set HUB_IPSEC_VPN ike ipsec-policy HUB_IPSEC_POL
user@hub# set HUB_IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
user@hub# set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 remote-ip 0.0.0.0/0
```

```
[edit security ipsec policy]
user@hub# set HUB_IPSEC_POL proposals HUB_IPSEC_PROP
```

3. Configure the quantum key manager key profile to retrieve quantum keys from the corresponding KME-Hub device.

```
[edit security key-manager profiles]
user@hub# set HUB_KM_PROFILE_1 quantum-key-manager url kme-server-urlset security key-manager
profiles HUB_KM_PROFILE_1 quantum-key-manager local-sae-id SAE_HUB
user@hub# set HUB_KM_PROFILE_1 quantum-key-manager local-certificate-id SAE_HUB_CERT
user@hub# set HUB_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
```

4. Bind the quantum key manager key profile as the IKE gateway ppk-profile to make the VPN tunnels quantum-safe.

```
[edit security ike gateway]
user@hub# set HUB_IKE_GW ppk-profile HUB_KM_PROFILE_1
```

If you are done configuring the device, enter `commit` from configuration mode.

Step-By-Step Configuration on Spoke Devices



NOTE: For complete sample configurations on the devices, see:

- "Appendix 1: Set Commands on all Devices" on page 301
- "Appendix 2: Show Configuration Output on DUT" on page 305

This configuration is applicable for Spoke 1 and Spoke 2 devices, you must make the appropriate device-specific configuration changes.

1. Configure the spoke interfaces.

```
[edit interfaces]
user@spoke# set ge-0/0/2 unit 0 family inet address 172.18.10.2/24
user@spoke# set ge-0/0/1 unit 0 family inet address 192.168.80.1/24
user@spoke# set st0 unit 1 family inet
```

2. Configure hub-spoke the IPsec VPN. This includes configuring the security zones, security policies, and relevant certificates for authenticating device identities and their communication links.

   Configure the hub to fetch the CA certificate from the CA server, or load a locally available CA certificate from the device.



NOTE: The KME certificates need to configured as per third-party vendor instructions.

   Configure the IPsec proposal and policy. Configure the IKE policy, proposal and gateway for the IPsec VPN.

```
[edit security zones]
user@spoke# set security-zone untrust host-inbound-traffic system-services ike
user@spoke# set security-zone untrust interfaces ge-0/0/2.0
user@spoke# set security-zone vpn interfaces st0.1
user@spoke# set security-zone trust host-inbound-traffic system-services ping
user@spoke# set security-zone trust interfaces ge-0/0/1.0
```

```
[edit security policies]
user@spoke# set from-zone trust to-zone vpn policy vpn_out match source-address any
user@spoke# set from-zone trust to-zone vpn policy vpn_out match destination-address any
user@spoke# set from-zone trust to-zone vpn policy vpn_out match application any
```

```
user@spoke# set from-zone trust to-zone vpn policy vpn_out then permit
user@spoke# set from-zone vpn to-zone trust policy vpn_in match source-address any
user@spoke# set from-zone vpn to-zone trust policy vpn_in match destination-address any
user@spoke# set from-zone vpn to-zone trust policy vpn_in match application any
user@spoke# set from-zone vpn to-zone trust policy vpn_in then permit
```

```
[edit security pki]
user@spoke# set ca-profile Root-CA ca-identity Root-CA
user@spoke# set ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
user@spoke# set ca-profile Root-CA revocation-check disable
```

```
user@spoke> request security pki ca-certificate enroll ca-profile Root-CA
```

```
user@spoke> request security pki generate-key-pair certificate-id SPOKE_1_CRT size 2048 type
rsa
user@spoke> request security pki local-certificate enroll certificate-id SPOKE_1_CRT
challenge-password <password> domain-name spoke_1.juniper.net email spoke_1@juniper.net
subject
DC=juniper,CN=spoke_1.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
user@spoke> request security pki local-certificate load certificate-id SAE_SPOKE_1 filename
SAE_SPOKE_1.cert key SAE_SPOKE_1.key
```

```
[edit security ike proposal]
user@spoke# set SPOKE_1_IKE_PROP authentication-method rsa-signatures
user@spoke# set SPOKE_1_IKE_PROP dh-group group14
user@spoke# set SPOKE_1_IKE_PROP authentication-algorithm sha-256
```

```
user@spoke# set SPOKE_1_IKE_PROP encryption-algorithm aes-256-cbc
user@spoke# set SPOKE_1_IKE_PROP lifetime-seconds 3600
```

```
[edit security ike policy]
user@spoke# set SPOKE_1_IKE_POL proposals SPOKE_1_IKE_PROP
user@spoke# set SPOKE_1_IKE_POL certificate local-certificate SPOKE_1_CRT
```

```
[edit security ike gateway]
user@spoke# set SPOKE_1_IKE_GW address 172.18.10.1
user@spoke# set SPOKE_1_IKE_GW local-address 172.18.10.2
user@spoke# set SPOKE_1_IKE_GW ike-policy SPOKE_1_IKE_POL
user@spoke# set SPOKE_1_IKE_GW external-interface ge-0/0/2.0
user@spoke# set SPOKE_1_IKE_GW local-identity distinguished-name
user@spoke# set SPOKE_1_IKE_GW remote-identity distinguished-name
user@spoke# set SPOKE_1_IKE_GW version v2-only
```

```
[edit security ipsec proposal]
user@spoke# set SPOKE_1_IPSEC_PROP protocol esp
user@spoke# set SPOKE_1_IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@spoke# set SPOKE_1_IPSEC_PROP encryption-algorithm aes-256-cbc
```

```
[edit security ipsec vpn]
user@spoke# set SPOKE_1_IPSEC_VPN bind-interface st0.1
user@spoke# set SPOKE_1_IPSEC_VPN ike gateway SPOKE_1_IKE_GW
user@spoke# set SPOKE_1_IPSEC_VPN ike ipsec-policy SPOKE_1_IPSEC_POL
user@spoke# set SPOKE_1_IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
user@spoke# set SPOKE_1_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
```

```
[edit security ipsec policy]
user@spoke# set SPOKE_1_IPSEC_POL proposals SPOKE_1_IPSEC_PROP
```

3. Configure the quantum key manager key profile to retrieve quantum keys from the corresponding spoke device.

```
[edit security key-manager profiles]
user@spoke# set SPOKE_1_KM_PROFILE_1 quantum-key-manager url https://www.kme_spoke_1-qkd-
server.net
user@spoke# set SPOKE_1_KM_PROFILE_1 quantum-key-manager local-sae-id SAE_SPOKE_1
user@spoke# set SPOKE_1_KM_PROFILE_1 quantum-key-manager local-certificate-id SAE_SPOKE_1_CERT
user@spoke# set profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
```

4. Bind the quantum key manager key profile as the IKE gateway ppk-profile to make the VPN tunnels quantum-safe.

```
[edit security ike gateway]
user@spoke# set SPOKE_1_IKE_GW ppk-profile SPOKE_1_KM_PROFILE_1
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verify IKE SAs | **292**
- Verify IPsec SAs | **295**
- Verify IPsec Statistics | **298**
- Verify Key Manager Profile | **299**
- Ping from Host 1 to Host 3 | **299**
- Ping from Host 2 to Host 3 | **300**

This section provides a list of show commands that you can use to verify the feature in this example.

Table 55: Verification Tasks

Command	Verification Task
show security ike security-associations detail	"Verify the IKE SAs." on page 292
show security ipsec security-associations detail	"Verify the IPsec SAs." on page 295
show security ipsec statistics	"Verify IPsec encryption and decryption statistics." on page 298
show security key-manager profiles detail	"Verify key profile statistics." on page 299
ping 192.168.90.20 source 192.168.80.20 count 4	"Ping from Host 1 to Host 3." on page 299
ping 192.168.90.20 source 192.168.70.20 count 4	"Ping from Host 2 to Host 3." on page 300

Verify IKE SAs

Purpose

Verify the IKE SAs.

Action

From operational mode, enter the `show security ike security-associations detail` command to view the IKE SAs.

```
user@hub> show security ike security-associations detail

IKE peer 172.18.10.3, Index 2161, Gateway Name: HUB_IKE_GW
  Role: Responder, State: UP
  Initiator cookie: bccc74c70f0b81b9, Responder cookie: 872d364f15b29c28
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/2.0
  Routing instance: default
  Local: 172.18.10.1:500, Remote: 172.18.10.3:500
  Lifetime: Expires in 3464 seconds
```

```
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke_2.juniper.net
  AAA assigned IP: 0.0.0.0
  PPK-profile: HUB_KM_PROFILE_1
     Optional: No
     State   : Used
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                 2661
   Output bytes  :                 2586
   Input  packets:                    5
   Output packets:                    5
   Input  fragmented packets:       4
   Output fragmented packets:       4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500446

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.18.10.1:500, Remote: 172.18.10.3:500
    Local identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=hub.juniper.net
    Remote identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke_2.juniper.net
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                Responder stats:
    Request Out             : 0                     Request In             :
0
    Response In             : 0                     Response Out           :
0
    No Proposal Chosen In   : 0                     No Proposal Chosen Out :
0
    Invalid KE In           : 0                     Invalid KE Out         :
0
```

```
    TS Unacceptable In      : 0                       TS Unacceptable Out    :
0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0

IKE peer 172.18.10.2, Index 2162, Gateway Name: HUB_IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 5e17d5924c619788, Responder cookie: 15f1e3c4252ba6f8
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/2.0
  Routing instance: default
  Local: 172.18.10.1:500, Remote: 172.18.10.2:500
  Lifetime: Expires in 3464 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke.juniper.net
  AAA assigned IP: 0.0.0.0
  PPK-profile: HUB_KM_PROFILE_1
     Optional: No
     State   : Used
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                 2645
   Output bytes  :                 2586
   Input  packets:                    5
   Output packets:                    5
   Input  fragmented packets:       4
   Output fragmented packets:       4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500447

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.18.10.1:500, Remote: 172.18.10.2:500
```

```
    Local identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=hub.juniper.net
    Remote identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke.juniper.net
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                Responder stats:
    Request Out             : 0                     Request In             :
0
    Response In             : 0                     Response Out           :
0
    No Proposal Chosen In   : 0                     No Proposal Chosen Out :
0
    Invalid KE In           : 0                     Invalid KE Out         :
0
    TS Unacceptable In      : 0                     TS Unacceptable Out    :
0
    Res DH Compute Key Fail : 0                     Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

Meaning

The sample output confirms the IKE SAs.

Verify IPsec SAs

Purpose

Verify the IPsec SAs.

Action

From operational mode, enter the `show security ipsec security-associations detail` command to view the IPsec SAs.

```
user@hub> show security ipsec security-associations detail
```

```
ID: 500446 Virtual-system: root, VPN Name: HUB_IPSEC_VPN
  Local Gateway: 172.18.10.1, Remote Gateway: 172.18.10.3
  Traffic Selector Name: ts1
  Local Identity: ipv4(192.168.90.0-192.168.90.255)
  Remote Identity: ipv4(192.168.70.0-192.168.70.255)
  TS Type: traffic-selector
  Version: IKEv2
  Quantum Secured: Yes
  PFS group: N/A
  Passive mode tunneling: Disabled
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: HUB_IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Fri Jul 21 2023 00:31:08: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0xcf48c0c9, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3464 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2778 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2161
  Direction: outbound, SPI: 0x86c9ba76, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3464 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2778 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2161

ID: 500447 Virtual-system: root, VPN Name: HUB_IPSEC_VPN
```

```
Local Gateway: 172.18.10.1, Remote Gateway: 172.18.10.2
Traffic Selector Name: ts1
Local Identity: ipv4(192.168.90.0-192.168.90.255)
Remote Identity: ipv4(192.168.80.0-192.168.80.255)
TS Type: traffic-selector
Version: IKEv2
Quantum Secured: Yes
PFS group: N/A
Passive mode tunneling: Disabled
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: HUB_IPSEC_POL
Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
Tunnel events:
  Fri Jul 21 2023 00:31:08: IPsec SA negotiation succeeds (1 times)
Location: FPC 0, PIC 0
Anchorship: Thread 1
Distribution-Profile: default-profile
Direction: inbound, SPI: 0x4275d756, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3464 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 2772 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Extended-Sequence-Number: Disabled
  tunnel-establishment: establish-tunnels-responder-only
  IKE SA Index: 2162
Direction: outbound, SPI: 0xe37b5568, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3464 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 2772 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Extended-Sequence-Number: Disabled
  tunnel-establishment: establish-tunnels-responder-only
  IKE SA Index: 2162
```

Meaning

The sample output confirms the IPsec SAs.

Verify IPsec Statistics

Purpose

Verify the IPsec statistics.

Action

From operational mode, enter the `show security ipsec statistics` command to view the IPsec statistics.

```
user@hub> show security ipsec statistics

ESP Statistics:
  Encrypted bytes:             1560
  Decrypted bytes:             1560
  Encrypted packets:             10
  Decrypted packets:             10
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Exceeds tunnel MTU: 0
  Discarded: 0
```

Meaning

The sample output confirms the IPsec statistics.

Verify Key Manager Profile

Purpose

Verify the key manager profile.

Action

From operational mode, enter the `show security key-manager profiles detail` command and verify the `Success` field in the `Request stats` option.

```
user@hub> show security key-manager profiles detail

Name: HUB_KM_PROFILE_1, Index: 6, Type: Quantum-key-manager
  Configured-at: 21.07.23 (00:14:00)
  Time-elapsed: 0 hrs 19 mins 24 secs
  Url: https://kme.juniper.net:8080
  Local-sae-id: SAE_HUB
  Local-certificate-id: SAE_HUB_CERT
  Trusted-cas: [ ROOT_CA_CERT ]
  Peer-sae-ids: N/A
  Default-key-size: N/A
  Request stats:
    Received: 2
    In-progress: 0
    Success: 2
    Failed: 0
```

Meaning

The sample output confirms the quantum key manager profile.

Ping from Host 1 to Host 3

Purpose

Verify the connectivity from Host 1 to Host 3.

Action

From operational mode, enter the `ping 192.168.90.20 source 192.168.80.20 count 5` command to view the connectivity from Host 1 to Host 3.

```
user@host1# ping 192.168.90.20 source 192.168.80.20 count 5
PING 192.168.90.20 (192.168.90.20): 56 data bytes
64 bytes from 192.168.90.20: icmp_seq=0 ttl=64 time=2.151 ms
64 bytes from 192.168.90.20: icmp_seq=1 ttl=64 time=1.710 ms
64 bytes from 192.168.90.20: icmp_seq=2 ttl=64 time=1.349 ms
64 bytes from 192.168.90.20: icmp_seq=3 ttl=64 time=1.597 ms
64 bytes from 192.168.90.20: icmp_seq=4 ttl=64 time=1.515 ms
--- 192.168.90.20 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.349/1.702/2.151/0.290 ms

Data traffic is successfully flowing between the HOSTs
```

Meaning

The sample output confirms the connectivity from Host 1 to Host 3.

Ping from Host 2 to Host 3

Purpose

Verify the connectivity from Host 2 to Host 3.

Action

From operational mode, enter the `ping 192.168.90.20 source 192.168.80.20 count 5` command to view the connectivity from Host 2 to Host 3.

```
user@host2# ping 192.168.90.20 source 192.168.70.20 count 5
PING 192.168.90.20 (192.168.90.20): 56 data bytes
64 bytes from 192.168.90.20: icmp_seq=0 ttl=64 time=2.151 ms
64 bytes from 192.168.90.20: icmp_seq=1 ttl=64 time=1.710 ms
64 bytes from 192.168.90.20: icmp_seq=2 ttl=64 time=1.349 ms
64 bytes from 192.168.90.20: icmp_seq=3 ttl=64 time=1.597 ms
64 bytes from 192.168.90.20: icmp_seq=4 ttl=64 time=1.759 ms
```

```
--- 192.168.90.20 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.349/1.702/2.151/0.290 ms

Data traffic is successfully flowing between the HOSTs
```

Meaning

The sample output confirms the connectivity from Host 2 to Host 3.

Appendix 1: Set Commands on all Devices

Set command output on all devices.

Set Commands on Hub

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id HUB_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id HUB_CRT challenge-password
<password> domain-name hub.juniper.net email hub@juniper.net subject
DC=juniper,CN=hub.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-profile
Root-CA
request security pki local-certificate load certificate-id SAE_HUB filename SAE_HUB.cert key
SAE_HUB.key
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager url https://www.kme_hub-
qkd-server.net
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager local-sae-id SAE_HUB
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_HUB_CERT
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
set security ike proposal HUB_IKE_PROP authentication-method rsa-signatures
set security ike proposal HUB_IKE_PROP dh-group group14
set security ike proposal HUB_IKE_PROP authentication-algorithm sha-256
set security ike proposal HUB_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal HUB_IKE_PROP lifetime-seconds 3600
set security ike policy HUB_IKE_POL proposals HUB_IKE_PROP
```

```
set security ike policy HUB_IKE_POL certificate local-certificate HUB_CRT
set security ike gateway HUB_IKE_GW local-address 172.18.10.1
set security ike gateway HUB_IKE_GW ike-policy HUB_IKE_POL
set security ike gateway HUB_IKE_GW external-interface ge-0/0/2.0
set security ike gateway HUB_IKE_GW local-identity distinguished-name
set security ike gateway HUB_IKE_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_IKE_GW dynamic distinguished-name wildcard C=us,DC=juniper
set security ike gateway HUB_IKE_GW ppk-profile HUB_KM_PROFILE_1
set security ike gateway HUB_IKE_GW version v2-only
set security ipsec proposal HUB_IPSEC_PROP protocol esp
set security ipsec proposal HUB_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal HUB_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy HUB_IPSEC_POL proposals HUB_IPSEC_PROP
set security ipsec vpn HUB_IPSEC_VPN bind-interface st0.1
set security ipsec vpn HUB_IPSEC_VPN ike gateway HUB_IKE_GW
set security ipsec vpn HUB_IPSEC_VPN ike ipsec-policy HUB_IPSEC_POL
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 remote-ip 0.0.0.0/0
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.1/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.90.1/24
set interfaces st0 unit 1 family inet
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Set Commands on Spoke 1

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
```

```
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id SPOKE_1_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_1_CRT challenge-password
<password> domain-name spoke_1.juniper.net email spoke_1@juniper.net subject
DC=juniper,CN=spoke_1.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
request security pki local-certificate load certificate-id SAE_SPOKE_1 filename SAE_SPOKE_1.cert
key SAE_SPOKE_1.key
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager url https://
www.kme_spoke_1-qkd-server.net
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager local-sae-id
SAE_SPOKE_1
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_SPOKE_1_CERT
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
set security ike proposal SPOKE_1_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_1_IKE_PROP dh-group group14
set security ike proposal SPOKE_1_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_1_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_1_IKE_PROP lifetime-seconds 3600
set security ike policy SPOKE_1_IKE_POL proposals SPOKE_1_IKE_PROP
set security ike policy SPOKE_1_IKE_POL certificate local-certificate SPOKE_1_CRT
set security ike gateway SPOKE_1_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_1_IKE_GW local-address 172.18.10.2
set security ike gateway SPOKE_1_IKE_GW ike-policy SPOKE_1_IKE_POL
set security ike gateway SPOKE_1_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_1_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW ppk-profile SPOKE_1_KM_PROFILE_1
set security ike gateway SPOKE_1_IKE_GW version v2-only
set security ipsec proposal SPOKE_1_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_1_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_1_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy SPOKE_1_IPSEC_POL proposals SPOKE_1_IPSEC_PROP
set security ipsec vpn SPOKE_1_IPSEC_VPN bind-interface st0.1
set security ipsec vpn SPOKE_1_IPSEC_VPN ike gateway SPOKE_1_IKE_GW
set security ipsec vpn SPOKE_1_IPSEC_VPN ike ipsec-policy SPOKE_1_IPSEC_POL
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.2/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.80.1/24
set interfaces st0 unit 1 family inet
```

```
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Set Commands on Spoke 2

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id SPOKE_2_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_2_CRT challenge-password
<password> domain-name spoke_2.juniper.net email spoke_2@juniper.net subject
DC=juniper,CN=spoke_2.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
request security pki local-certificate load certificate-id SAE_SPOKE_2 filename SAE_SPOKE_2.cert
key SAE_SPOKE_2.key
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager url https://
www.kme_spoke_2-qkd-server.net
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager local-sae-id
SAE_SPOKE_2
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_SPOKE_2_CERT
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
set security ike proposal SPOKE_2_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_2_IKE_PROP dh-group group14
set security ike proposal SPOKE_2_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_2_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_2_IKE_PROP lifetime-seconds 3600
```

```
set security ike policy SPOKE_2_IKE_POL proposals SPOKE_IKE_PROP
set security ike policy SPOKE_2_IKE_POL certificate local-certificate SPOKE_2_CRT
set security ike gateway SPOKE_2_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_2_IKE_GW local-address 172.18.10.3
set security ike gateway SPOKE_2_IKE_GW ike-policy SPOKE_2_IKE_POL
set security ike gateway SPOKE_2_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_2_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW ppk-profile SPOKE_2_KM_PROFILE_1
set security ike gateway SPOKE_2_IKE_GW version v2-only
set security ipsec proposal SPOKE_2_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_2_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_2_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy SPOKE_2_IPSEC_POL proposals SPOKE_2_IPSEC_PROP
set security ipsec vpn SPOKE_2_IPSEC_VPN bind-interface st0.2
set security ipsec vpn SPOKE_2_IPSEC_VPN ike gateway SPOKE_2_IKE_GW
set security ipsec vpn SPOKE_2_IPSEC_VPN ike ipsec-policy SPOKE_2_IPSEC_POL
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 local-ip 192.168.70.0/24
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.3/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.70.1/24
set interfaces st0 unit 2 family inet
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.2
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Appendix 2: Show Configuration Output on DUT

Show command output on the DUT.

Hub

From configuration mode, confirm your configuration by entering the `show security pki ca-profile Root-CA`, `show security key-manager`, `show security ike proposal HUB_IKE_PROP`, `show security ike policy HUB_IKE_POL`, `show security ike gateway HUB_IKE_GW`, `show security ipsec proposal HUB_IPSEC_PROP`, `show security ipsec policy HUB_IPSEC_POL`, `show security ipsec vpn HUB_IPSEC_VPN`, `show security zones security-zone untrust`, `show security zones security-zone trust`, `show security policies from-zone trust to-zone vpn`, `show security policies from-zone vpn to-zone trust`, and `show interfaces` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@hub# show security pki ca-profile Root-CA
ca-identity Root-CA;
enrollment {
    url https://ca-server.juniper.net/certsrv/mscep/mscep.dll;
}
revocation-check {
    disable;
}
```

```
user@hub# show security key-manager
profiles {
    km_profile_1 {
        static {
            key-id ascii-text "$9$7VNs4UDkPT3Hq9A01yrWLxNYoPfTz3924"; ## SECRET-DATA
            key ascii-text "$9$RraElM7NbwgJ-
VkPTFAtxNdws4GUHqmTaZ36AtOBwY24UHfTz9A0JGu1IhrlGDjHmTFn/9p0fT39p0hc-
VwgGiPfzn9pJGqfQnpurev8xds2aDjqX7"; ## SECRET-DATA
        }
    }
    HUB_KM_PROFILE_1 {
        quantum-key-manager {
            url https://www.kme_hub-qkd-server.net;
            local-sae-id SAE_HUB;
            local-certificate-id SAE_HUB_CERT;
            trusted-cas Root-CA;
        }
```

```
    }
}
```

```
user@hub# show security ike proposal HUB_IKE_PROP
authentication-method rsa-signatures;
dh-group group14;
authentication-algorithm sha-256;
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@hub# show security ike policy HUB_IKE_POL
proposals HUB_IKE_PROP;
certificate {
    local-certificate HUB_CRT;
}
```

```
user@hub# show security ike gateway HUB_IKE_GW
ike-policy HUB_IKE_POL;
dynamic {
    distinguished-name {
        wildcard C=us,DC=juniper;
    }
    ike-user-type group-ike-id;
}
local-identity distinguished-name;
external-interface ge-0/0/2.0;
local-address 172.18.10.1;
version v2-only;
ppk-profile HUB_KM_PROFILE_1;
```

```
user@hub# show security ipsec proposal HUB_IPSEC_PROP
protocol esp;
```

```
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
```

```
user@hub# show security ipsec policy HUB_IPSEC_POL
proposals HUB_IPSEC_PROP;
```

```
user@hub# show security ipsec vpn HUB_IPSEC_VPN
bind-interface st0.1;
ike {
    gateway HUB_IKE_GW;
    ipsec-policy HUB_IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.90.0/24;
    remote-ip 0.0.0.0/0;
}
```

```
user@hub# show security zones security-zone untrust
host-inbound-traffic {
    system-services {
        ike;
    }
}
interfaces {
    ge-0/0/0.0;
    ge-0/0/2.0;
}
```

```
user@hub# show security zones security-zone trust
host-inbound-traffic {
    system-services {
        ping;
    }
}
interfaces {
```

```
    ge-0/0/1.0;
}
```

```
user@hub# show security policies from-zone trust to-zone vpn
policy vpn_out {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
```

```
user@hub# show security policies from-zone vpn to-zone trust
policy vpn_in {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
```

```
user@hub# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
```

```
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
        }
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
```

Spoke 1

From configuration mode, confirm your configuration by entering the `show security pki ca-profile Root-CA`, `show security key-manager profiles SPOKE_1_KM_PROFILE_1`, `show security ike proposal SPOKE_1_IKE_PROP`, `show security ike policy SPOKE_1_IKE_POL`, `show security ike gateway SPOKE_1_IKE_GW`, `show security ipsec proposal SPOKE_1_IPSEC_PROP`, `show security ipsec policy SPOKE_1_IPSEC_POL`, `show security ipsec vpn SPOKE_1_IPSEC_VPN`, `show interfaces`, `show security zones security-zone untrust`, `show security zones security-zone trust`, `show security policies from-zone trust to-zone vpn`, and `show security policies from-zone vpn to-zone trust` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@spoke1# show security pki ca-profile Root-CA
ca-identity Root-CA;
enrollment {
    url https://ca-server.juniper.net/certsrv/mscep/mscep.dll;
}
revocation-check {
    disable;
}
```

```
user@spoke1# show security key-manager profiles SPOKE_1_KM_PROFILE_1
quantum-key-manager {
    url https://www.kme_spoke_1-qkd-server.net;
    local-sae-id SAE_SPOKE_1;
```

```
    local-certificate-id SAE_SPOKE_1_CERT;
    trusted-cas Root-CA;
}
```

```
user@spoke1# show security ike proposal SPOKE_1_IKE_PROP
authentication-method rsa-signatures;
dh-group group14;
authentication-algorithm sha-256;
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@spoke1# show security ike policy SPOKE_1_IKE_POL
proposals SPOKE_1_IKE_PROP;
certificate {
    local-certificate SPOKE_1_CRT;
}
```

```
user@spoke1# show security ike gateway SPOKE_1_IKE_GW
ike-policy SPOKE_1_IKE_POL;
address 172.18.10.1;
local-identity distinguished-name;
remote-identity distinguished-name;
external-interface ge-0/0/2.0;
local-address 172.18.10.2;
version v2-only;
ppk-profile SPOKE_1_KM_PROFILE_1;
```

```
user@spoke1# show security ipsec proposal SPOKE_1_IPSEC_PROP
protocol esp;
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
```

```
user@spoke1# show security ipsec vpn SPOKE_1_IPSEC_VPN
bind-interface st0.1;
ike {
    gateway SPOKE_1_IKE_GW;
```

```
    ipsec-policy SPOKE_1_IPSEC_POL;
}
traffic-selector ts1 {
    local-ip 192.168.80.0/24;
    remote-ip 192.168.90.0/24;
}
```

```
user@spoke1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
        }
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
```

```
user@spoke1# show security zones security-zone untrust
host-inbound-traffic {
    system-services {
```

```
        ike;
    }
}
interfaces {
    ge-0/0/0.0;
    ge-0/0/2.0;
}
```

```
user@spoke1# show security zones security-zone trust
host-inbound-traffic {
    system-services {
        ping;
    }
}
interfaces {
    ge-0/0/1.0;
}
```

```
user@spoke1# show security policies from-zone trust to-zone vpn
policy vpn_out {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
```

```
user@spoke1# security policies from-zone vpn to-zone trust
policy vpn_in {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
```

```
    }
}
```

Spoke 2

From configuration mode, confirm your configuration by entering the `show security pki ca-profile Root-CA`, `show security key-manager profiles SPOKE_1_KM_PROFILE_1`, `show security ike proposal SPOKE_1_IKE_PROP`, `show security ike policy SPOKE_1_IKE_POL`, `show security ike gateway SPOKE_1_IKE_GW`, `show security ipsec proposal SPOKE_1_IPSEC_PROP`, `show security ipsec policy SPOKE_1_IPSEC_POL`, `show security ipsec vpn SPOKE_1_IPSEC_VPN`, `show interfaces`, `show security zones security-zone untrust`, `show security zones security-zone trust`, `show security policies from-zone trust to-zone vpn`, and `show security policies from-zone vpn to-zone trust` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@spoke2# show security pki ca-profile Root-CA
ca-identity Root-CA;
enrollment {
    url https://ca-server.juniper.net/certsrv/mscep/mscep.dll;
}
revocation-check {
    disable;
}
```

```
user@spoke2# show security key-manager profiles SPOKE_2_KM_PROFILE_1
quantum-key-manager {
    url https://www.kme_spoke_2-qkd-server.net;
    local-sae-id SAE_SPOKE_2;
    local-certificate-id SAE_SPOKE_2_CERT;
    trusted-cas Root-CA;
}
```

```
user@spoke2# show security ike proposal SPOKE_2_IKE_PROP
authentication-method rsa-signatures;
dh-group group14;
authentication-algorithm sha-256;
```

```
encryption-algorithm aes-256-cbc;
lifetime-seconds 3600;
```

```
user@spoke2# show security ike policy SPOKE_2_IKE_POL
##
## Warning: Referenced proposal is not defined
##
proposals SPOKE_IKE_PROP;
certificate {
    local-certificate SPOKE_2_CRT;
}
```

```
user@spoke2# show security ike gateway SPOKE_2_IKE_GW
ike-policy SPOKE_2_IKE_POL;
address 172.18.10.1;
local-identity distinguished-name;
remote-identity distinguished-name;
external-interface ge-0/0/2.0;
local-address 172.18.10.3;
version v2-only;
ppk-profile SPOKE_2_KM_PROFILE_1;
```

```
user@spoke2# show security ipsec proposal SPOKE_2_IPSEC_PROP
protocol esp;
authentication-algorithm hmac-sha-256-128;
encryption-algorithm aes-256-cbc;
```

```
user@spoke2# show security ipsec policy SPOKE_2_IPSEC_POL
proposals SPOKE_2_IPSEC_PROP;

[edit]
user@spoke2# show security ipsec vpn SPOKE_2_IPSEC_VPN
bind-interface st0.2;
ike {
    gateway SPOKE_2_IKE_GW;
    ipsec-policy SPOKE_2_IPSEC_POL;
}
```

```
traffic-selector ts1 {
    local-ip 192.168.70.0/24;
    remote-ip 192.168.90.0/24;
}
```

```
user@spoke2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.90.1/24;
            address 192.168.80.1/24;
            address 192.168.70.1/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.18.10.1/24;
            address 172.18.10.2/24;
            address 172.18.10.3/24;
        }
    }
}
st0 {
    unit 1 {
        family inet;
    }
    unit 2 {
        family inet;
```

```
    }
}
```

```
user@spoke2# show security zones security-zone untrust
host-inbound-traffic {
    system-services {
        ike;
    }
}
interfaces {
    ge-0/0/0.0;
    ge-0/0/2.0;
}
```

```
user@spoke2# show security zones security-zone vpn
interfaces {
    st0.1;
    st0.2;
}
```

```
user@spoke2# show security zones security-zone trust
host-inbound-traffic {
    system-services {
        ping;
    }
}
interfaces {
    ge-0/0/1.0;
}
```

```
user@spoke2# show security policies from-zone trust to-zone vpn
policy vpn_out {
    match {
        source-address any;
        destination-address any;
        application any;
    }
```

```
    then {
        permit;
    }
}
```

```
user@spoke2# show security policies from-zone vpn to-zone trust
policy vpn_in {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
```

4

CHAPTER

Route Based VPN

IN THIS CHAPTER

- Route-Based IPsec VPNs | **320**
- Route-Based VPN with IKEv2 | **347**
- Secure Tunnel Interface in a Virtual Router | **409**
- Dual Stack Tunnels over an External Interface | **421**
- IPsec VPN Tunnels with Chassis Clusters | **440**
- Traffic Selectors in Route-Based VPNs | **451**
- Migrate to Route-Based IPsec VPNs | **480**

Route-Based IPsec VPNs

IN THIS SECTION

- Understanding Route-Based IPsec VPNs | 320
- Example: Configuring a Route-Based VPN | 321

A route-based VPN is a configuration in which an IPsec VPN tunnel created between two end points is referenced by a route that determines which traffic is sent through the tunnel based on a destination IP address.

Understanding Route-Based IPsec VPNs

With route-based VPNs, you can configure dozens of security policies to regulate traffic flowing through a single VPN tunnel between two sites, and there is just one set of IKE and IPsec SAs at work. Unlike policy-based VPNs, for route-based VPNs, a policy refers to a destination address, not a VPN tunnel. When Junos OS looks up a route to find the interface to use to send traffic to the packet's destination address, it finds a route through a secure tunnel interface (st0.*x*). The tunnel interface is bound to a specific VPN tunnel, and the traffic is routed to the tunnel if the policy action is permit.

A secure tunnel (st0) interface supports only one IPv4 address and one IPv6 address at the same time. This applies to all route-based VPNs. The `disable` option is not supported on st0 interfaces.



NOTE: A secure tunnel interface (st0) from st0.16000 to st0.16385 is reserved for Multinode High Availability and for HA control link encryption in Chassis Cluster. These interfaces are not user configurable interfaces. You can only use interfaces from st0.0 to st0.15999.

Examples of where route-based VPNs can be used:

- There are overlapping subnets or IP addresses between the two LANs.
- A hub-and-spoke VPN topology is used in the network, and spoke-to-spoke traffic is required.
- Primary and backup VPNs are required.

- A dynamic routing protocol (for example, OSPF, RIP, or BGP) is running across the VPN.

  Configuring RIP demand circuits over point-to-multipoint VPN interfaces is not supported.

We recommend that you use route-based VPN when you want to configure VPN between multiple remote sites. Route-based VPN allows for routing between the spokes between multiple remote sites; it is easier to configure, monitor, and troubleshoot.

SEE ALSO

Class of Service User Guide (Security Devices)

IPsec Overview | 12

Example: Configuring a Route-Based VPN

IN THIS SECTION

- Requirements | 321
- Overview | 322
- Configuration | 325
- Verification | 340

This example shows how to configure a route-based IPsec VPN to allow data to be securely transferred between two sites.

Requirements

This example uses the following hardware:

- Any SRX Series Firewall
  - Updated and revalidated using vSRX Virtual Firewall on Junos OS Release 20.4R1.

NOTE: Are you interested in getting hands-on experience with the topics and operations covered in this guide? Visit the IPsec Route-Based VPN demonstration in Juniper

Networks Virtual Labs and reserve your free sandbox today! You'll find the IPsec VPN Route-Based sandbox in the Security category.

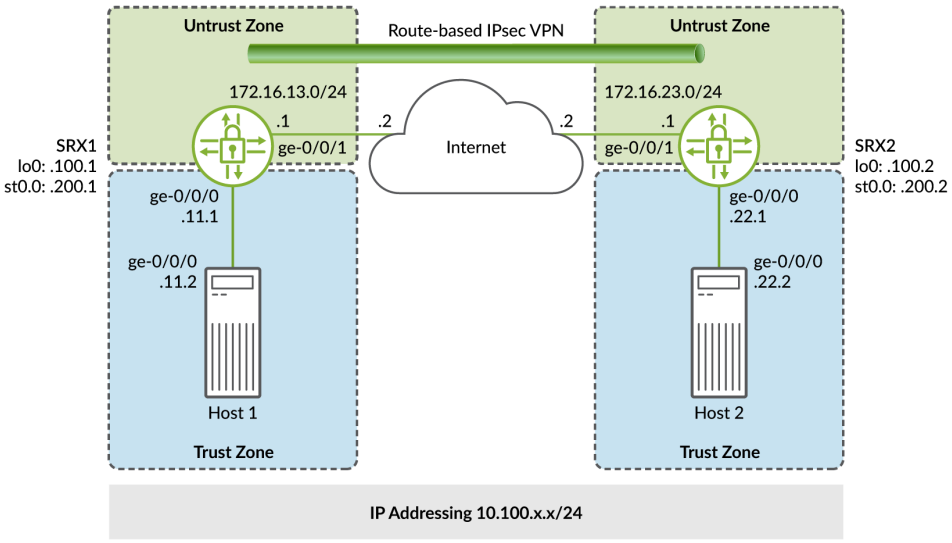
Before you begin, read "IPsec Overview" on page 12.

Overview

In this example, you configure a route-based VPN on SRX1 and SRX2. Host1 and Host2 use the VPN to send traffic securely over the Internet between both hosts.

Figure 23 on page 322 shows an example of a route-based VPN topology.

Figure 23: Route-Based VPN Topology



In this example, you configure interfaces, an IPv4 default route, and security zones. Then you configure IKE, IPsec, security policy, and TCP-MSS parameters. See Table 56 on page 322 through Table 60 on page 325 for specific configuration parameters used in this example.

Table 56: Interface, Static Route, Security Zone, and Security Policy Information for SRX1

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	10.100.11.1/24

Table 56: Interface, Static Route, Security Zone, and Security Policy Information for SRX1 *(Continued)*

Feature	Name	Configuration Parameters
	ge-0/0/1.0	172.16.13.1/24
	st0.0 (tunnel interface)	10.100.200.1/24
Static routes	10.100.22.0/24 0.0.0.0/0	The next hop is st0.0. The next hop is 172.16.13.2.
Security zones	trust	• The ge-0/0/0.0 interface is bound to this zone.
	untrust	• The ge-0/0/1.0 interface is bound to this zone.
	vpn	• The st0.0 interface is bound to this zone.

Table 57: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	standard	• Authentication method: pre-shared-keys
Policy	IKE-POL	• Mode: main • Proposal reference: standard • IKE policy authentication method: pre-shared-keys

Table 57: IKE Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Gateway	IKE-GW	• IKE policy reference: IKE-POL • External interface: ge-0/0/1 • Gateway address: 172.16.23.1

Table 58: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	standard	• Using default configuration
Policy	IPSEC-POL	• Proposal reference: standard
VPN	VPN-to-Host2	• IKE gateway reference: IKE-GW • IPsec policy reference: IPSEC-POL • Bind to interface: st0.0 • establish-tunnels immediately

Table 59: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

The security policy permits traffic from the trust zone to the VPN zone.	VPN-OUT	• Match criteria:
		  • source-address Host1-Net
		  • destination-address Host2-Net
		  • application any
		• Action: permit

Table 59: Security Policy Configuration Parameters *(Continued)*

Purpose	Name	Configuration Parameters

The security policy permits traffic from the VPN zone to the trust zone.	VPN-IN	• Match criteria:
		  • source-address Host2-Net
		  • destination-address Host1-Net
		  • application any
		• Action: permit

Table 60: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. For VPN traffic, the IPsec encapsulation overhead, along with the IP and the frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, which causes fragmentation. Fragmentation increases bandwidth and the device resources. We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.	MSS value: 1350

Configuration

IN THIS SECTION

- Configure Basic Network and Security Zone Information | **326**
- Configuring IKE | **330**

- Configuring IPsec | **332**
- Configuring Security Policies | **334**
- Configuring TCP-MSS | **338**
- Configuring SRX2 | **339**

Configure Basic Network and Security Zone Information

CLI Quick Configuration

To quickly configure this section of the example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.100.11.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.13.1/24
set interfaces lo0 unit 0 family inet address 10.100.100.1/32
set interfaces st0 unit 0 family inet address 10.100.200.1/24
set routing-options static route 10.100.22.0/24 next-hop st0.0
set routing-options static route 0.0.0.0/0 next-hop 172.16.13.2
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone VPN host-inbound-traffic system-services ping
set security zones security-zone VPN interfaces st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure interface, static route, and security zone information:

1. Configure the interfaces.

```
[edit]
user@SRX1# set interfaces ge-0/0/0 unit 0 family inet address 10.100.11.1/24
user@SRX1# set interfaces ge-0/0/1 unit 0 family inet address 172.16.13.1/24
user@SRX1# set interfaces lo0 unit 0 family inet address 10.100.100.1/32
user@SRX1# set interfaces st0 unit 0 family inet address 10.100.200.1/24
```

2. Configure the static routes.

```
[edit]
user@SRX1# set routing-options static route 10.100.22.0/24 next-hop st0.0
user@SRX1# set routing-options static route 0.0.0.0/0 next-hop 172.16.13.2
```

3. Assign the Internet facing interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@SRX1# set interfaces ge-0/0/1.0
```

4. Specify the allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@SRX1# set host-inbound-traffic system-services ike
user@SRX1# set host-inbound-traffic system-services ping
```

5. Assign the Host1 facing interface to the trust security zone.

```
[edit security zones security-zone trust]
user@SRX1# set interfaces ge-0/0/0.0
```

6. Specify the allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@SRX1# set host-inbound-traffic system-services all
```

7. Assign the secure tunnel interface to the VPN security zone.

```
[edit security zones security-zone VPN]
user@SRX1# set interfaces st0.0
```

8. Specify the allowed system services for the VPN security zone.

```
[edit security zones security-zone VPN]
user@SRX1# set host-inbound-traffic system-services ping
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.100.11.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.16.13.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.100.100.1/32;
        }
    }
}
```

```
st0 {
    unit 0 {
        family inet {
            address 10.100.200.1/24;
        }
    }
}
```

```
[edit]
user@SRX1# show routing-options
static {
    route 10.100.22.0/24 next-hop st0.0;
    route 0.0.0.0/0 next-hop 172.16.13.2;
}
```

```
[edit]
user@SRX1# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone VPN {
    host-inbound-traffic {
```

```
        system-services {
            ping;
        }
    }
    interfaces {
        st0.0;
    }
}
```

Configuring IKE

CLI Quick Configuration

To quickly configure this section of the example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set security ike proposal standard authentication-method pre-shared-keys
set security ike policy IKE-POL mode main
set security ike policy IKE-POL proposals standard
set security ike policy IKE-POL pre-shared-key ascii-text $ABC123
set security ike gateway IKE-GW ike-policy IKE-POL
set security ike gateway IKE-GW address 172.16.23.1
set security ike gateway IKE-GW external-interface ge-0/0/1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure IKE:

1. Create the IKE proposal.

```
[edit security ike]
user@SRX1# set proposal standard
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal standard]
user@SRX1# set authentication-method pre-shared-keys
```

3. Create an IKE policy.

```
[edit security ike]
user@SRX1# set policy IKE-POL
```

4. Set the IKE policy mode.

```
[edit security ike policy IKE-POL]
user@SRX1# set mode main
```

5. Specify a reference to the IKE proposal.

```
[edit security ike policy IKE-POL]
user@SRX1# set proposals standard
```

6. Define the IKE policy authentication method.

```
[edit security ike policy IKE-POL]
user@SRX1# set pre-shared-key ascii-text $ABC123
```

7. Create an IKE gateway and define its external interface.

```
[edit security ike]
user@SRX1# set gateway IKE-GW external-interface ge-0/0/1
```

8. Define the IKE policy reference.

```
[edit security ike gateway IKE-GW]
user@SRX1# set ike-policy IKE-POL
```

9. Define the IKE gateway address.

```
[edit security ike gateway IKE-GW]
user@SRX1# set address 172.16.23.1
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show security ike
proposal standard {
    authentication-method pre-shared-keys;
}
policy IKE-POL {
    mode main;
    proposals standard;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway IKE-GW {
    ike-policy IKE-POL;
    address 172.16.23.1;
    external-interface ge-0/0/1;
}
```

Configuring IPsec

CLI Quick Configuration

To quickly configure this section of the example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set security ipsec proposal standard
set security ipsec policy IPSEC-POL proposals standard
```

```
set security ipsec vpn VPN-to-Host2 bind-interface st0.0
set security ipsec vpn VPN-to-Host2 ike gateway IKE-GW
set security ipsec vpn VPN-to-Host2 ike ipsec-policy IPSEC-POL
set security ipsec vpn VPN-to-Host2 establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure IPsec:

1. Create an IPsec proposal.

```
[edit]
user@SRX1# set security ipsec proposal standard
```

2. Create the IPsec policy.

```
[edit security ipsec]
user@SRX1# set policy IPSEC-POL
```

3. Specify the IPsec proposal reference.

```
[edit security ipsec policy IPSEC-POL]
user@SRX1# set proposals standard
```

4. Specify the IKE gateway.

```
[edit security ipsec]
user@SRX1# set vpn VPN-to-Host2 ike gateway IKE-GW
```

5. Specify the IPsec policy.

```
[edit security ipsec]
user@host# set vpn VPN-to-Host2 ike ipsec-policy IPSEC-POL
```

6. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn VPN-to-Host2 bind-interface st0.0
```

7. Configure the tunnel to establish immediately.

```
[edit security ipsec]
user@host# set vpn VPN-to-Host2 establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal standard;
policy IPSEC-POL {
    proposals standard;
}
vpn VPN-to-Host2 {
    bind-interface st0.0;
    ike {
        gateway IKE-GW;
        ipsec-policy IPSEC-POL;
    }
    establish-tunnels immediately;
}
```

Configuring Security Policies

CLI Quick Configuration

To quickly configure security policies for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and

paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set security address-book Host1 address Host1-Net 10.100.11.0/24
set security address-book Host1 attach zone trust
set security address-book Host2 address Host2-Net 10.100.22.0/24
set security address-book Host2 attach zone VPN
set security policies from-zone trust to-zone untrust policy default-permit match source-address
any
set security policies from-zone trust to-zone untrust policy default-permit match destination-
address any
set security policies from-zone trust to-zone untrust policy default-permit match application any
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies from-zone trust to-zone VPN policy VPN-OUT match source-address Host1-Net
set security policies from-zone trust to-zone VPN policy VPN-OUT match destination-address Host2-
Net
set security policies from-zone trust to-zone VPN policy VPN-OUT match application any
set security policies from-zone trust to-zone VPN policy VPN-OUT then permit
set security policies from-zone VPN to-zone trust policy VPN-IN match source-address Host2-Net
set security policies from-zone VPN to-zone trust policy VPN-IN match destination-address Host1-
Net
set security policies from-zone VPN to-zone trust policy VPN-IN match application any
set security policies from-zone VPN to-zone trust policy VPN-IN then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure security policies:

1. Create address book entries for the networks that will be used in the security policies.

```
[edit]
user@SRX1# set security address-book Host1 address Host1-Net 10.100.11.0/24
user@SRX1# set security address-book Host1 attach zone trust
user@SRX1# set security address-book Host2 address Host2-Net 10.100.22.0/24
user@SRX1# set security address-book Host2 attach zone VPN
```

2. Create a security policy to permit traffic from the trust zone to the untrust zone for traffic to the Internet.

```
[edit security policies from-zone trust to-zone untrust]
user@SRX1# set policy default-permit match source-address any
user@SRX1# set policy default-permit match destination-address any
user@SRX1# set policy default-permit match application any
user@SRX1# set policy default-permit then permit
```

3. Create a security policy to permit traffic from Host1 in the trust zone destined to Host2 in the VPN zone.

```
[edit security policies from-zone trust to-zone VPN]
user@SRX1# set policy VPN-OUT match source-address Host1-Net
user@SRX1# set policy VPN-OUT match destination-address Host2-Net
user@SRX1# set policy VPN-OUT match application any
user@SRX1# set policy VPN-OUT then permit
```

4. Create a security policy to permit traffic from Host2 in the VPN zone to Host1 in the trust zone.

```
[edit security policies from-zone VPN to-zone trust]
user@host# set policy VPN-IN match source-address Host2-Net
user@host# set policy VPN-IN match destination-address Host1-Net
user@host# set policy VPN-IN match application any
user@host# set policy VPN-IN then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security address-book` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security address-book
Host1 {
    address Host1-Net 10.100.11.0/24;
    attach {
        zone trust;
```

```
    }
}
Host2 {
    address Host2-Net 10.100.22.0/24;
    attach {
        zone VPN;
    }
}
user@host# show security policies
from-zone trust to-zone untrust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone trust to-zone VPN {
    policy VPN-OUT {
        match {
            source-address Host1-Net;
            destination-address Host2-Net;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone VPN to-zone trust {
    policy VPN-IN {
        match {
            source-address Host2-Net;
            destination-address Host1-Net;
            application any;
        }
        then {
            permit;
        }
```

```
    }
}
```

Configuring TCP-MSS

CLI Quick Configuration

To quickly configure the TCP MSS for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure TCP-MSS information:

1. Configure the TCP-MSS information.

```
[edit]
user@SRX1# set security flow tcp-mss ipsec-vpn mss 1350
```

Results

From configuration mode, confirm your configuration by entering the `show security flow` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
```

```
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring SRX2

CLI Quick Configuration

For reference, the configuration for the SRX2 is provided.

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set security ike proposal standard authentication-method pre-shared-keys
set security ike policy IKE-POL mode main
set security ike policy IKE-POL proposals standard
set security ike policy IKE-POL pre-shared-key ascii-text $ABC123
set security ike gateway IKE-GW ike-policy IKE-POL
set security ike gateway IKE-GW address 172.16.13.1
set security ike gateway IKE-GW external-interface ge-0/0/1
set security ipsec proposal standard
set security ipsec policy IPSEC-POL proposals standard
set security ipsec vpn VPN-to-Host1 bind-interface st0.0
set security ipsec vpn VPN-to-Host1 ike gateway IKE-GW
set security ipsec vpn VPN-to-Host1 ike ipsec-policy IPSEC-POL
set security ipsec vpn VPN-to-Host1 establish-tunnels immediately
set security address-book Host1 address Host1-Net 10.100.11.0/24
set security address-book Host1 attach zone VPN
set security address-book Host2 address Host2-Net 10.100.22.0/24
set security address-book Host2 attach zone trust
set security flow tcp-mss ipsec-vpn mss 1350
set security policies from-zone trust to-zone untrust policy default-permit match source-address
any
set security policies from-zone trust to-zone untrust policy default-permit match destination-
address any
set security policies from-zone trust to-zone untrust policy default-permit match application any
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies from-zone trust to-zone VPN policy VPN-OUT match source-address Host2-Net
```

```
set security policies from-zone trust to-zone VPN policy VPN-OUT match destination-address Host1-
Net
set security policies from-zone trust to-zone VPN policy VPN-OUT match application any
set security policies from-zone trust to-zone VPN policy VPN-OUT then permit
set security policies from-zone VPN to-zone trust policy VPN-IN match source-address Host1-Net
set security policies from-zone VPN to-zone trust policy VPN-IN match destination-address Host2-
Net
set security policies from-zone VPN to-zone trust policy VPN-IN match application any
set security policies from-zone VPN to-zone trust policy VPN-IN then permit
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone VPN host-inbound-traffic system-services ping
set security zones security-zone VPN interfaces st0.0
set interfaces ge-0/0/0 unit 0 family inet address 10.100.22.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.23.1/24
set interfaces lo0 unit 0 family inet address 10.100.100.2/32
set interfaces st0 unit 0 family inet address 10.100.200.2/24
set routing-options static route 10.100.11.0/24 next-hop st0.0
set routing-options static route 0.0.0.0/0 next-hop 172.16.23.2
```

Verification

IN THIS SECTION

- Verify the IKE Status | **341**
- Verify the IPsec Status | **343**
- Test Traffic Flow Across the VPN | **345**
- Review Statistics and Errors for an IPsec Security Association | **346**

Perform these tasks to confirm that the configuration is working properly:

Verify the IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@SRX1> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
1859340 UP     b153dc24ec214da9  5af2ee0c2043041a  Main           172.16.23.1
```

```
user@SRX1> show security ike security-associations index 1859340 detail
IKE peer 172.16.23.1, Index 1859340, Gateway Name: IKE-GW
  Role: Responder, State: UP
  Initiator cookie: b153dc24ec214da9, Responder cookie: 5af2ee0c2043041a
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 172.16.13.1:500, Remote: 172.16.23.1:500
  Lifetime: Expires in 23038 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: 172.16.23.1
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                 1236
   Output bytes  :                  868
   Input  packets:                    9
   Output packets:                    5
   Input  fragmentated packets:       0
   Output fragmentated packets:       0
```

```
IPSec security associations: 2 created, 2 deleted
Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Responder, Message ID: 0
  Local: 172.16.13.1:500, Remote: 172.16.23.1:500
  Local identity: 172.16.13.1
  Remote identity: 172.16.23.1
  Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE SAs. If no SAs are listed, there was a problem with IKE establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The IKE SA has been established.
  - DOWN—There was a problem establishing the IKE SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Proposal parameters (must match on both peers)

The `show security ike security-associations index 1859340 detail` command lists additional information about the security association with an index number of 1859340:

- Authentication and encryption algorithms used
- lifetime

- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of negotiations in progress

Verify the IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@SRX1> show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131074 ESP:3des/sha1   912f9063 3403/ unlim  -   root 500   172.16.23.1
  >131074 ESP:3des/sha1   71dbaa56 3403/ unlim  -   root 500   172.16.23.1
```

```
user@SRX1> show security ipsec security-associations index 131074 detail
ID: 131074 Virtual-system: root, VPN Name: VPN-to-Host2
  Local Gateway: 172.16.13.1, Remote Gateway: 172.16.23.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 500, Nego#: 26, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Fri Jul 23 2021 10:46:34 -0700: IPSec SA negotiation successfully completed (23 times)
    Fri Jul 23 2021 09:07:24 -0700: IKE SA negotiation successfully completed (3 times)
```

```
    Thu Jul 22 2021 16:34:17 -0700: Negotiation failed with INVALID_SYNTAX error (3 times)
    Thu Jul 22 2021 16:33:50 -0700: Tunnel configuration changed. Corresponding IKE/IPSec SAs
are deleted (1 times)
    Thu Jul 22 2021 16:23:49 -0700: IPSec SA negotiation successfully completed (2 times)
    Thu Jul 22 2021 15:34:12
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Thu Jul 22 2021 15:33:25 -0700: IPSec SA negotiation successfully completed (1 times)
    Thu Jul 22 2021 15:33:25
    : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
    Thu Jul 22 2021 15:33:25 -0700: External interface's address received. Information updated
(1 times)
    Thu Jul 22 2021 15:33:25 -0700: Bind-interface's zone received. Information updated (1 times)
    Thu Jul 22 2021 10:34:55 -0700: IKE SA negotiation successfully completed (1 times)
    Thu Jul 22 2021 10:34:46 -0700: No response from peer. Negotiation failed (16 times)
  Direction: inbound, SPI: 912f9063, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3302 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2729 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 71dbaa56, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3302 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2729 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 131074. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.
- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3403/ unlim value indicates that the lifetime expires in 3403 seconds, and that no lifesize has been

specified, which indicates that it is unlimited. Lifetime can differ from lifetime, as IPsec is not dependent on IKE after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 131074 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a IPsec failure. If no IPsec SA is listed, confirm that IPsec proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Another common reason for IPsec failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

Test Traffic Flow Across the VPN

Purpose

Verify the traffic flow across the VPN.

Action

Use the `ping` command from the Host1 device to test traffic flow to Host2.

```
user@Host1> ping 10.100.22.1 rapid count 100
PING 10.100.22.1 (10.100.22.1): 56 data bytes
!!!
```

Meaning

If the `ping` command fails from Host1, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

Review Statistics and Errors for an IPsec Security Association

Purpose

Review ESP and authentication header counters and errors for an IPsec security association.

Action

From operational mode, enter the `show security ipsec statistics index` *`index_number`* command, using the index number of the VPN for which you want to see statistics.

```
user@SRX1> show security ipsec statistics index 131074
ESP Statistics:
  Encrypted bytes:            13600
  Decrypted bytes:             8400
  Encrypted packets:            100
  Decrypted packets:            100
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

Meaning

If you see packet loss issues across a VPN, run the `show security ipsec statistics` or `show security ipsec statistics detail` command several times to confirm if the encrypted and decrypted packet counters are incrementing. Look in the command output for any incrementing error counters.

SEE ALSO

IPsec Overview | 12

IPsec Route Based VPN in Juniper Networks Virtual Labs

Route-Based VPN with IKEv2

IN THIS SECTION

- Example: Configuring a Route-Based VPN for IKEv2 | 348
- Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload | 374
- IKE Policy with a Trusted CA | 407

Internet Key Exchange version 2 (IKEv2) is an IPsec based tunneling protocol that provides a secure VPN communication channel between peer VPN devices and defines negotiation and authentication for IPsec security associations (SAs) in a protected manner.

Table 61 on page 347 describes the IPsec Radius xAuth or CP values.

Table 61: IPsec Radius xAuth or CP values

Radius Attribute	Attribute ID	Attribute Name	Vendor ID (Dictionary)	Vendor Attribute ID	Attribute Value	Type
Standard	8	Framed IP address	NA	NA	IP address	IPv4 address
Standard	9	Framed IP netmask	NA	NA	IP address	IPv4 address
Standard	88	Framed pool	NA	NA	Name	Text
Standard	100	Framed IPv6 pool	NA	NA	Name	Text

Table 61: IPsec Radius xAuth or CP values *(Continued)*

Radius Attribute	Attribute ID	Attribute Name	Vendor ID (Dictionary)	Vendor Attribute ID	Attribute Value	Type
Vendor	26	Primary DNS	4874 (Juniper ERX)	4	IP address	IPv4 address
Vendor	26	Secondary DNS	4874 (Juniper ERX)	5	IP address	IPv4 address
Vendor	26	Primary WINS (NBNS)	4874 (Juniper ERX)	6	IP address	IPv4 address
Vendor	26	Secondary WINS (NBNS)	4874 (Juniper ERX)	7	IP address	IPv4 address
Vendor	26	IPv6 primary DNS	4874 (Juniper ERX)	47	IP address	hex-string or octets
Vendor	26	IPv6 secondary DNS	4874 (Juniper ERX)	48	IP address	hex-string or octets

Example: Configuring a Route-Based VPN for IKEv2

IN THIS SECTION

- Requirements | 349
- Overview | 349
- Configuration | 353
- Verification | 368

This example shows how to configure a route-based IPsec VPN to allow data to be securely transferred between a branch office and a corporate office.

Requirements

This example uses the following hardware:

- SRX240 device
- SSG140 device

Before you begin, read "IPsec Overview" on page 12.

Overview

In this example, you configure a route-based VPN for a branch office in Chicago, Illinois, because you want to conserve tunnel resources but still get granular restrictions on VPN traffic. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

In this example, you configure interfaces, an IPv4 default route, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, a security policy, and TCP-MSS parameters. See Table 62 on page 349 through Table 66 on page 352 for specific configuration parameters used in this example.

Table 62: Interface, Static Route, Security Zone, and Address Book Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	192.168.10.1/24
	ge-0/0/3.0	10.1.1.2/30
	st0.0 (tunnel interface)	10.11.11.10/24
Static routes	0.0.0.0/0 (default route)	The next hop is 10.1.1.1.
	192.168.168.0/24	The next hop is st0.0.
Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.

Table 62: Interface, Static Route, Security Zone, and Address Book Information *(Continued)*

Feature	Name	Configuration Parameters
	untrust	• IKE is the only allowed system service. • The ge-0/0/3.0 interface is bound to this zone.
	vpn-chicago	The st0.0 interface is bound to this zone.
Address book entries	sunnyvale	• This address is for the trust zone's address book. • The address for this address book entry is 192.168.10.0/24.
	chicago	• This address is for the untrust zone's address book. • The address for this address book entry is 192.168.168.0/24.

Table 63: IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc

Table 63: IKE Phase 1 Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw-chicago	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2

Table 64: IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2
VPN	ipsec-vpn-chicago	• IKE gateway reference: gw-chicago • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0

Table 65: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

The security policy permits traffic from the trust zone to the vpn-chicago zone.	vpn-tr-chi	• Match criteria:
		  • source-address sunnyvale
		  • destination-address chicago
		  • application any
		• Action: permit

The security policy permits traffic from the vpn-chicago zone to the trust zone.	vpn-chi-tr	• Match criteria:
		  • source-address chicago
		  • destination-address sunnyvale
		  • application any
		• Action: permit

Table 66: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. For VPN traffic, the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, which causes fragmentation. Fragmentation increases bandwidth and device resources. We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.	MSS value: 1350

Configuration

IN THIS SECTION

- Configuring Interface, Static Route, Security Zone, and Address Book Information | 353
- Configuring IKE | 358
- Configuring IPsec | 361
- Configuring Security Policies | 364
- Configuring TCP-MSS | 366
- Configuring the SSG Series Device | 367

Configuring Interface, Static Route, Security Zone, and Address Book Information

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
set interfaces st0 unit 0 family inet address 10.11.11.10/24
set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
set routing-options static route 192.168.168.0/24 next-hop st0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust address-book address sunnyvale 192.168.10.0/24
set security zones security-zone vpn-chicago interfaces st0.0
set security zones security-zone vpn-chicago address-book address chicago 192.168.168.0/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interface, static route, security zone, and address book information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.1/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
user@host# set interfaces st0 unit 0 family inet address 10.11.11.10/24
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
user@host# set routing-options static route 192.168.168.0/24 next-hop st0.0
```

3. Configure the untrust security zone.

```
[edit ]
user@host# edit security zones security-zone untrust
```

4. Assign an interface to the security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/3.0
```

5. Specify allowed system services for the security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services ike
```

6. Configure the trust security zone.

```
[edit]
user@host# edit security zones security-zone trust
```

7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/0.0
```

8. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

9. Configure the address book entry for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set address-book address sunnyvale 192.168.10.0/24
```

10. Configure the vpn-chicago security zone.

```
[edit]
user@host# edit security zones security-zone vpn-chicago
```

11. Assign an interface to the security zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set interfaces st0.0
```

12. Configure the address book entry for the vpn-chicago zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set address-book address chicago 192.168.168.0/24
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.168.10.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.1.2/30
        }
    }
}
st0{
    unit 0 {
        family inet {
            address 10.11.11.10/24
        }
    }
}
```

```
[edit]
user@host# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.1.1.1;
    route 192.168.168.0/24 next-hop st0.0;
}
```

```
[edit]
user@host# show security zones
```

```
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone trust {
    address-book {
        address sunnyvale 192.168.10.0/24;
    }
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn-chicago {
    address-book {
        address chicago 192.168.168.0/24;
    }
    interfaces {
        st0.0;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text “$ABC123”
set security ike gateway gw-chicago external-interface ge-0/0/3.0
set security ike gateway gw-chicago ike-policy ike-phase1-policy
set security ike gateway gw-chicago address 10.2.2.2
set security ike gateway gw-chicago version v2-only
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
user@host# set proposal ike-phase1-proposal
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@host# set policy ike-phase1-policy
```

7. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@host# set proposals ike-phase1-proposal
```

8. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike-phase1-policy]
user@host# set pre-shared-key ascii-text “$ABC123”
```

9. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@host# set gateway gw-chicago external-interface ge-0/0/3.0
```

10. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gw-chicago]
user@host# set ike-policy ike-phase1-policy
```

11. Define the IKE Phase 1 gateway address.

```
[edit security ike gateway gw-chicago]
user@host# set address 10.2.2.2
```

12. Define the IKE Phase 1 gateway version.

```
[edit security ike gateway gw-chicago]
user@host# set version v2-only
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike-phase1-proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
    proposals ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway gw-chicago {
    ike-policy ike-phase1-policy;
    address 10.2.2.2;
    external-interface ge-0/0/3.0;
```

```
    version v2-only;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ipsec-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ipsec-vpn-chicago ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn ipsec-vpn-chicago bind-interface st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@host# set security ipsec proposal ipsec-phase2-proposal
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set policy ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set proposals ipsec-phase2-proposal
```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn ipsec-vpn-chicago ike gateway gw-chicago
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn ipsec-vpn-chicago ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn ipsec-vpn-chicago bind-interface st0.0
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec-phase2-proposal;
}
vpn ipsec-vpn-chicago {
    bind-interface st0.0;
    ike {
        gateway gw-chicago;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match source-address
sunnyvale
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match destination-
address chicago
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match application
any
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi then permit
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match source-address
chicago
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match destination-
address sunnyvale
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match application
any
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the vpn-chicago zone.

```
[edit security policies from-zone trust to-zone vpn-chicago]
user@host# set policy vpn-tr-chi match source-address sunnyvale
user@host# set policy vpn-tr-chi match destination-address chicago
user@host# set policy vpn-tr-chi match application any
user@host# set policy vpn-tr-chi then permit
```

2. Create the security policy to permit traffic from the vpn-chicago zone to the trust zone.

```
[edit security policies from-zone vpn-chicago to-zone trust]
user@host# set policy vpn-chi-tr match source-address sunnyvale
user@host# set policy vpn-chi-tr match destination-address chicago
user@host# set policy vpn-chi-tr match application any
user@host# set policy vpn-chi-tr then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone vpn-chicago {
    policy vpn-tr-vpn {
        match {
            source-address sunnyvale;
            destination-address chicago;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn-chicago to-zone trust {
    policy vpn-tr-vpn {
        match {
            source-address chicago;
            destination-address sunnyvale;
            application any;
        }
        then {
            permit;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring TCP-MSS

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure TCP-MSS information:

1. Configure TCP-MSS information.

```
[edit]
user@host# set security flow tcp-mss ipsec-vpn mss 1350
```

Results

From configuration mode, confirm your configuration by entering the `show security flow` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the SSG Series Device

CLI Quick Configuration

For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts & Examples ScreenOS Reference Guide*, which is located at https://www.juniper.net/documentation.

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set zone name vpn-chicago
set interface ethernet0/6 zone Trust
set interface ethernet0/0 zone Untrust
set interface tunnel.1 zone vpn-chicago
set interface ethernet0/6 ip 192.168.168.1/24
set interface ethernet0/6 route
set interface ethernet0/0 ip 10.2.2.2/30
set interface ethernet0/0 route
set interface tunnel.1 ip 10.11.11.11/24
set flow tcp-mss 1350
set address Trust “192.168.168-net” 192.168.168.0 255.255.255.0
set address vpn-chicago "192.168.10-net" 192.168.10.0 255.255.255.0
set ike gateway corp-ike address 10.1.1.2 IKEv2 outgoing-interface ethernet0/0 preshare
395psksecr3t sec-level standard
set vpn corp-vpn gateway corp-ike replay tunnel idletime 0 sec-level standard
set vpn corp-vpn monitor optimized rekey
set vpn corp-vpn bind interface tunnel.1
set policy from Trust to Untrust “ANY” “ANY” “ANY” nat src permit
set policy from Trust to vpn-chicago “192.168.168-net” “192.168.10-net” “ANY” permit
set policy from vpn-chicago to Trust “192.168.10-net” “192.168.168-net” “ANY” permit
set route 192.168.10.0/24 interface tunnel.1
set route 0.0.0.0/0 interface ethernet0/0 gateway 10.2.2.1
```

Verification

IN THIS SECTION

- Verifying the IKE Phase 1 Status | 368
- Verifying the IPsec Phase 2 Status | 370
- Reviewing Statistics and Errors for an IPsec Security Association | 372
- Testing Traffic Flow Across the VPN | 373

Confirm that the configuration is working properly.

Verifying the IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

Before starting the verification process, you need to send traffic from a host in the 192.168.10/24 network to a host in the 192.168.168/24 network. For route-based VPNs, traffic can be initiated by the SRX Series Firewall through the tunnel. We recommend that when testing IPsec tunnels, test traffic be sent from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate a ping from 192.168.10.10 to 192.168.168.10.

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@host> show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
1       10.2.2.2        UP     744a594d957dd513  1e1307db82f58387  IKEv2
```

```
user@host> show security ike security-associations index 1 detail
IKE peer 10.2.2.2, Index 1,
  Role: Responder, State: UP
  Initiator cookie: 744a594d957dd513, Responder cookie: 1e1307db82f58387
```

```
Exchange type: IKEv2, Authentication method: Pre-shared-keys
Local: 10.1.1.2:500, Remote: 10.2.2.2:500
Lifetime: Expires in 28570 seconds
Algorithms:
 Authentication        : sha1
 Encryption            : aes-cbc (128 bits)
 Pseudo random function: hmac-sha1
Traffic statistics:
 Input bytes    :                 852
 Output bytes   :                 940
 Input packets  :                   5
 Output packets :                   5
Flags: Caller notification sent
IPSec security associations: 1 created, 0 deleted
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets).
- IKE policy parameters.
- Preshared key information.
- Phase 1 proposal parameters (must match on both peers).

The `show security ike security-associations index 1 detail` command lists additional information about the SA with an index number of 1:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created

Verifying the IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@host> show security ipsec security-associations
  total configured sa: 2
  ID    Gateway     Port  Algorithm       SPI      Life:sec/kb  Mon vsys
  <16384 10.2.2.2    500   ESP:aes-128/sha1   76d64d1d 3363/ unlim   -   0
  >16384 10.2.2.2    500   ESP:aes-128/sha1   a1024ee2 3363/ unlim   -   0
```

```
user@host> show security ipsec security-associations index 16384 detail
  Virtual-system: Root
  Local Gateway: 10.1.1.2, Remote Gateway: 10.2.2.2
  Local Identity: ipv4_subnet(any:0,[0..7]=192.168.10.0/24)
  Remote Identity: ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
  Version: IKEv2

    DF-bit: clear
```

```
    Direction: inbound, SPI: 1993755933, AUX-SPI: 0
    Hard lifetime: Expires in 3352 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 2775 seconds
    Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: enabled, Replay window size: 32

    Direction: outbound, SPI: 2701283042, AUX-SPI: 0
    Hard lifetime: Expires in 3352 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 2775 seconds
    Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc
(128 bits)
    Anti-replay service: enabled, Replay window size: 32
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 16384. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.
- There is one IPsec SA pair using port 500.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3363/ unlim value indicates that the Phase 2 lifetime expires in 3363 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, because Phase 2 is not dependent on Phase 1 after the VPN is up.
- The vsys is the root system, and it is always listed as 0.
- The IKEv2 allows connections from a version 2 peer and will initiate a version 2 negotiation.

The output from the `show security ipsec security-associations index 16384 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each

IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

Reviewing Statistics and Errors for an IPsec Security Association

Purpose

Review ESP and authentication header counters and errors for an IPsec SA.

Action

From operational mode, enter the `show security ipsec statistics index` *`index_number`* command, using the index number of the VPN for which you want to see statistics.

```
user@host> show security ipsec statistics index 16384
ESP Statistics:
  Encrypted bytes:              920
  Decrypted bytes:             6208
  Encrypted packets:              5
  Decrypted packets:             87
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

Meaning

If you see packet loss issues across a VPN, you can run the `show security ipsec statistics` or `show security ipsec statistics detail` command several times to confirm that the encrypted and decrypted packet counters are incrementing. You should also check that the other error counters are incrementing.

Testing Traffic Flow Across the VPN

Purpose

Verify the traffic flow across the VPN.

Action

You can use the `ping` command from the SRX Series Firewall to test traffic flow to a remote host PC. Make sure that you specify the source interface so that the route lookup is correct and the appropriate security zones are referenced during policy lookup.

From operational mode, enter the `ping` command.

```
ssg-> ping 192.168.168.10 interface ge-0/0/0 count 5
PING 192.168.168.10 (192.168.168.10): 56 data bytes
64 bytes from 192.168.168.10: icmp_seq=0 ttl=127 time=8.287 ms
64 bytes from 192.168.168.10: icmp_seq=1 ttl=127 time=4.119 ms
64 bytes from 192.168.168.10: icmp_seq=2 ttl=127 time=5.399 ms
64 bytes from 192.168.168.10: icmp_seq=3 ttl=127 time=4.361 ms
64 bytes from 192.168.168.10: icmp_seq=4 ttl=127 time=5.137 ms

--- 192.168.168.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.119/5.461/8.287/1.490 ms
```

You can also use the `ping` command from the SSG Series device.

```
user@host> ping 192.168.10.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.10.10, timeout is 1 seconds from ethernet0/6
!!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=4/4/5 ms
```

Meaning

If the `ping` command fails from the SRX Series or SSG Series device, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

SEE ALSO

IPsec Overview | 12

Example: Configuring a Hub-and-Spoke VPN

Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload

IN THIS SECTION

- Requirements | 374
- Overview | 374
- Configuration | 380
- Verification | 401

In networks where many devices are being deployed, managing the network needs to be simple. The IKEv2 configuration payload feature supports the provisioning of these devices without touching either the device configuration or the SRX Series configuration. This example shows how to configure an SRX Series to support pico cell provisioning using the IKEv2 configuration payload feature.

Requirements

This example uses the following hardware and software components:

- Two SRX Series Firewalls configured in a chassis cluster
- One SRX Series Firewall configured as an intermediate router
- Two pico cell clients
- One RADIUS server configured with pico cell client provisioning information
- Junos OS Release 12.1X46-D10 or later for IKEv2 configuration payload support

Overview

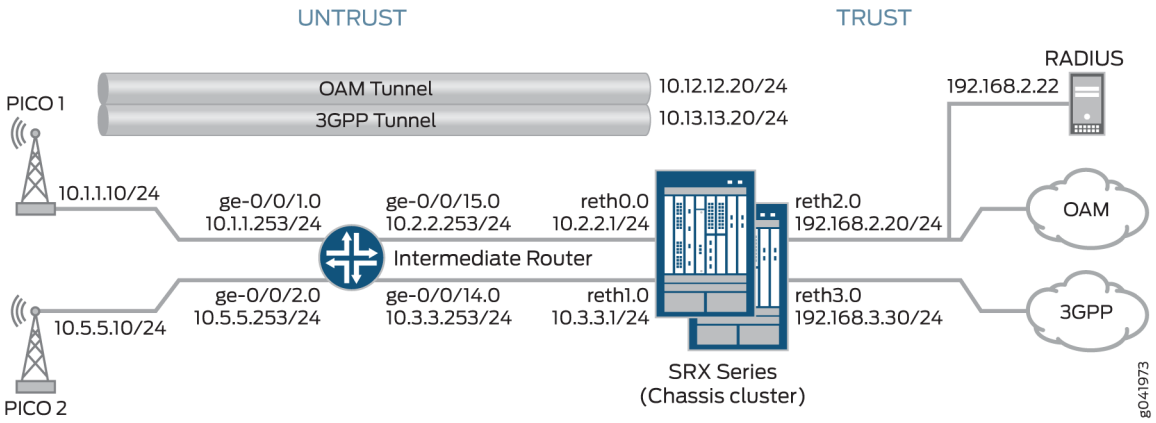
In this example, an SRX Series uses the IKEv2 configuration payload feature to propagate provisioning information to a series of pico cells. The pico cells ship from the factory with a standard configuration that allows them to connect to the SRX Series, but the pico cell provisioning information is stored on an

external RADIUS server. The pico cells receive full provisioning information after establishing secure connections with provisioning servers in a protected network. IKEv2 configuration payload is supported for both IPv4 and IPV6. This example covers IKEv2 configuration payload for IPv4, however you can configure with IPv6 addresses as well.

Starting in Junos OS Release 20.3R1, we support IKEv2 IPv6 configuration payload for assigning IPv6 address on SRX5000 line running iked process. The same support is included in vSRX Virtual Firewall running iked process starting from Junos OS Release 21.1R1.

Figure 24 on page 375 shows a topology in which the SRX Series supports pico cell provisioning using the IKEv2 configuration payload feature.

Figure 24: SRX Series Support for Pico Cell Provisioning with IKEv2 Configuration Payload



Each pico cell in this topology initiates two IPsec VPNs: one for management and one for data. In this example, management traffic uses the tunnel labeled OAM Tunnel, while the data traffic flows through the tunnel labeled 3GPP Tunnel. Each tunnel supports connections with OAM and 3GPP provisioning servers on separate, configurable networks, requiring separate routing instances and VPNs. This example provides the IKE Phase 1 and Phase 2 options for establishing the OAM and 3GPP VPNs.

In this example, the SRX Series acts as the IKEv2 configuration payload server, acquiring provisioning information from the RADIUS server and providing that information to the pico cell clients. The SRX Series returns the provisioning information for each authorized client in the IKEv2 configuration payload during tunnel negotiation. The SRX Series cannot be used as a client device.

Additionally, the SRX Series uses the IKEv2 configuration payload information to update the Traffic Selector initiator (TSi) and Traffic Selector responder (TSr) values exchanged with the client during tunnel negotiation. The configuration payload uses the TSi and TSr values that are configured on the SRX Series using the `proxy-identity` statement at the [`edit security ipsec vpn` *`vpn-name`* `ike`] hierarchy level. The TSi and TSr values define the network traffic for each VPN.

The intermediate router routes pico cell traffic to the appropriate interfaces on the SRX Series.

The following process describes the connection sequence:

1. The pico cell initiates an IPsec tunnel with the SRX Series using the factory configuration.
2. The SRX Series authenticates the client using the client certificate information and the root certificate of the CA that is enrolled in the SRX Series. After authentication, the SRX Series passes the IKE identity information from the client certificate to the RADIUS server in an authorization request.
3. After authorizing the client, the RADIUS server responds to the SRX Series with the client provisioning information:
   - IP address (TSi value)
   - IP subnet mask (optional; the default is 32 bit)
   - DNS address (optional)
4. The SRX Series returns the provisioning information in the IKEv2 configuration payload for each client connection, and exchanges final TSi and TSr values with the pico cells. In this example, the SRX Series provides the following TSi and TSr information for each VPN:

VPN Connection	TSi/TSr Values Provided by SRX
Pico 1 OAM	TSi: 10.12.1.201/32, TSr: 192.168.2.0/24
Pico 1 3GPP	TSi: 10.13.1.201/32, TSr: 192.168.3.0/24, TSr: 10.13.0.0/16
Pico 2 OAM	TSi: 10.12.1.205/32, TSr: 192.168.2.0/24
Pico 2 3GPP	TSi: 10.13.1.205/32, TSr: 192.168.3.0/24, TSr: 10.13.0.0/16

If the provisioning information supplied by the RADIUS server includes a subnet mask, the SRX Series returns a second TSr value for the client connection that includes the IP subnet. This enables intrapeer communication for devices on that subnet. In this example, intrapeer communication is enabled for the subnet associated with the 3GPP VPN (13.13.0.0/16).

The IKEv2 configuration payload feature is supported for both point-to-multipoint secure tunnel (st0) interfaces and point-to-point interfaces. For point-to-multipoint interfaces, the interfaces must be numbered, and the addresses provided in the configuration payload must be within the subnetwork range of the associated point-to-multipoint interface.

Starting in Junos OS Release 20.1R1, we support IKEv2 configuration payload feature with point-to-point interfaces on SRX5000 line and vSRX Virtual Firewall running iked.

Multinode High Availability supports IKEv2 configuration payload feature with point-to-point interfaces for secure tunnel (st0).

Table 67 on page 377 shows the Phase 1 and Phase 2 options configured on the SRX Series, including information for establishing both OAM and 3GPP tunnels.

Table 67: Phase 1 and Phase 2 Options for the SRX Series

Option	Value
IKE proposal:	
Proposal name	IKE_PROP
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	group5
Authentication algorithm	SHA-1
Encryption algorithm	AES 256 CBC
IKE policy:	
IKE Policy name	IKE_POL
Local certificate	Example_SRX
IKE gateway (OAM):	
IKE policy	IKE_POL
Remote IP address	dynamic
IKE user type	group-ike-id
Local IKE ID	hostname srx_series.example.net

Table 67: Phase 1 and Phase 2 Options for the SRX Series *(Continued)*

Option	Value
Remote IKE ID	hostname .pico_cell.net
External interface	reth0.0
Access profile	radius_pico
IKE version	v2-only
IKE gateway (3GPP):	
IKE policy	IKE_POL
Remote IP address	Dynamic
IKE user type	group-ike-id
Local IKE ID	distinguished-name wildcard OU=srx_series
Remote IKE ID	distinguished-name wildcard OU=pico_cell
External interface	reth1
Access profile	radius_pico
IKE version	v2-only
IPsec proposal:	
Proposal name	IPSEC_PROP
Protocol	ESP

Table 67: Phase 1 and Phase 2 Options for the SRX Series *(Continued)*

Option	Value
Authentication algorithm	HMAC SHA-1 96
Encryption algorithm	AES 256 CBC
IPsec policy:	
Policy name	IPSEC_POL
Perfect Forward Secrecy (PFS) keys	group5
IPsec proposals	IPSEC_PROP
IPsec VPN (OAM):	
Bind interface	st0.0
IKE gateway	OAM_GW
Local proxy-identity	192.168.2.0/24
Remote proxy-identity	0.0.0.0/0
IPsec policy	IPSEC_POL
IPsec VPN (3GPP):	
Bind interface	st0.1
IKE gateway	3GPP_GW
Local proxy-identity	192.168.3.0/24

Table 67: Phase 1 and Phase 2 Options for the SRX Series *(Continued)*

Option	Value
Remote proxy-identity	0.0.0.0/0
IPsec policy	IPSEC_POL

Certificates are stored on the pico cells and the SRX Series.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Configuration

IN THIS SECTION

- Configuring the SRX Series | **380**
- Configuring the Intermediate Router | **393**
- Configuring the Pico Cell (Sample Configuration) | **397**
- Configuring the RADIUS Server (Sample Configuration using a FreeRADIUS) | **399**

Configuring the SRX Series

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set chassis cluster reth-count 5
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 250
set chassis cluster redundancy-group 0 node 1 priority 150
set chassis cluster redundancy-group 1 node 0 priority 220
set chassis cluster redundancy-group 1 node 1 priority 149
```

```
set chassis cluster redundancy-group 1 interface-monitor ge-3/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/2/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/2/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/2/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/2/1 weight 255
set interfaces ge-3/0/0 gigether-options redundant-parent reth0
set interfaces ge-3/0/1 gigether-options redundant-parent reth1
set interfaces ge-3/2/0 gigether-options redundant-parent reth2
set interfaces ge-3/2/1 gigether-options redundant-parent reth3
set interfaces ge-8/0/0 gigether-options redundant-parent reth0
set interfaces ge-8/0/1 gigether-options redundant-parent reth1
set interfaces ge-8/2/0 gigether-options redundant-parent reth2
set interfaces ge-8/2/1 gigether-options redundant-parent reth3
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.2.2.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.3.3.1/24
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth2 unit 0 family inet address 192.168.2.20/24
set interfaces reth3 redundant-ether-options redundancy-group 1
set interfaces reth3 unit 0 family inet address 192.168.3.20/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.12.1.20/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 10.13.1.20/24
set routing-options static route 10.1.0.0/16 next-hop 10.2.2.253
set routing-options static route 10.5.0.0/16 next-hop 10.2.2.253
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth0.0
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone oam-trust host-inbound-traffic system-services all
set security zones security-zone oam-trust host-inbound-traffic protocols all
set security zones security-zone oam-trust interfaces reth2.0
set security zones security-zone oam-trust interfaces st0.0
set security zones security-zone 3gpp-trust host-inbound-traffic system-services all
set security zones security-zone 3gpp-trust host-inbound-traffic protocols all
set security zones security-zone 3gpp-trust interfaces reth3.0
set security zones security-zone 3gpp-trust interfaces st0.1
set access profile radius_pico authentication-order radius
```

```
set access profile radius_pico radius-server 192.168.2.22 secret "$ABC123"
set access profile radius_pico radius-server 192.168.2.22 routing-instance VR-OAM
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate example_SRX
set security ike gateway OAM_GW ike-policy IKE_POL
set security ike gateway OAM_GW dynamic hostname .pico_cell.net
set security ike gateway OAM_GW dynamic ike-user-type group-ike-id
set security ike gateway OAM_GW local-identity hostname srx_series.example.net
set security ike gateway OAM_GW external-interface reth0.0
set security ike gateway OAM_GW aaa access-profile radius_pico
set security ike gateway OAM_GW version v2-only
set security ike gateway 3GPP_GW ike-policy IKE_POL
set security ike gateway 3GPP_GW dynamic distinguished-name wildcard OU=pico_cell
set security ike gateway 3GPP_GW dynamic ike-user-type group-ike-id
set security ike gateway 3GPP_GW local-identity distinguished-name wildcard OU=srx_series
set security ike gateway 3GPP_GW external-interface reth1.0
set security ike gateway 3GPP_GW aaa access-profile radius_pico
set security ike gateway 3GPP_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal IPSEC_PROP lifetime-seconds 300
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn OAM_VPN bind-interface st0.0
set security ipsec vpn OAM_VPN ike gateway OAM_GW
set security ipsec vpn OAM_VPN ike proxy-identity local 192.168.2.0/24
set security ipsec vpn OAM_VPN ike proxy-identity remote 0.0.0.0/0
set security ipsec vpn OAM_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn 3GPP_VPN bind-interface st0.1
set security ipsec vpn 3GPP_VPN ike gateway 3GPP_GW
set security ipsec vpn 3GPP_VPN ike proxy-identity local 192.168.3.0/24
set security ipsec vpn 3GPP_VPN ike proxy-identity remote 0.0.0.0/0
set security ipsec vpn 3GPP_VPN ike ipsec-policy IPSEC_POL
set routing-instances VR-OAM instance-type virtual-router
set routing-instances VR-OAM interface reth2.0
set routing-instances VR-OAM interface st0.0
set routing-instances VR-3GPP instance-type virtual-router
set routing-instances VR-3GPP interface reth3.0
```

```
set routing-instances VR-3GPP interface st0.1
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the SRX Series:

1. Configure the chassis cluster.

```
[edit chassis cluster]
user@host# set reth-count 5
user@host# set node 0
user@host# set node 1
user@host#set redundancy-group 0 node 0 priority 250
user@host#set redundancy-group 0 node 1 priority 150
user@host#set redundancy-group 1 node 0 priority 220
user@host#set redundancy-group 1 node 1 priority 149
user@host# set redundancy-group 1 interface-monitor ge-3/0/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-3/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-3/2/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/2/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-3/2/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/2/1 weight 255
```

2. Configure interfaces.

```
[edit interfaces]
user@host# set ge-3/0/0 gigether-options redundant-parent reth0
user@host# set ge-3/0/1 gigether-options redundant-parent reth1
user@host# set ge-3/2/0 gigether-options redundant-parent reth2
user@host# set ge-3/2/1 gigether-options redundant-parent reth3
user@host# set ge-8/0/0 gigether-options redundant-parent reth0
user@host# set ge-8/0/1 gigether-options redundant-parent reth1
user@host# set ge-8/2/0 gigether-options redundant-parent reth2
user@host# set ge-8/2/1 gigether-options redundant-parent reth3
user@host# set reth0 redundant-ether-options redundancy-group 1
```

```
user@host# set reth0 unit 0 family inet address 10.2.2.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 10.3.3.1/24
user@host# set reth2 redundant-ether-options redundancy-group 1
user@host# set reth2 unit 0 family inet address 192.168.2.20/24
user@host# set reth3 redundant-ether-options redundancy-group 1
user@host# set reth3 unit 0 family inet address 192.169.3.20/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.12.1.20/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 10.13.1.20/24
```

3. Configure routing options.

```
[edit routing-options]
user@host# set static route 10.1.0.0/16 next-hop 10.2.2.253
user@host# set static route 10.5.0.0/16 next-hop 10.2.2.253
```

4. Specify security zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
user@host# set interfaces reth0.0
user@host# set interfaces reth1.0
[edit security zones security-zone oam-trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth2.0
user@host# set interfaces st0.0
[edit security zones security-zone 3gpp-trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth3.0
user@host# set interfaces st0.1
```

5. Create the RADIUS profile.

```
[edit access profile radius_pico]
user@host# set authentication-order radius
```

```
user@host# set radius-server 192.168.2.22 secret “$ABC123”
user@host# set radius-server 192.168.2.22 routing-instance VR-OAM
```

6. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate example_SRX
[edit security ike gateway OAM_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic hostname .pico_cell.net
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity hostname srx.example.net
user@host# set external-interface reth0.0
user@host# set aaa access-profile radius_pico
user@host# set version v2-only
[edit security ike gateway 3GPP_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=pico_cell
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name wildcard OU=srx_series
user@host# set external-interface reth1.0
user@host# set aaa access-profile radius_pico
user@host# set version v2-only
```

7. Specify Phase 2 options.

```
[edit set security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 300
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn OAM_VPN]
```

```
user@host# set bind-interface st0.0
user@host# set ike gateway OAM_GW
user@host# set ike proxy-identity local 192.168.2.0/24
user@host# set ike proxy-identity remote 0.0.0.0/0
user@host# set ike ipsec-policy IPSEC_POL
[edit security ipsec vpn 3GPP_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway 3GPP_GW
user@host# set ike proxy-identity local 192.168.3.0/24
user@host# set ike proxy-identity remote 0.0.0.0/0
user@host# set ike ipsec-policy IPSEC_POL
```

8. Specify the routing instances.

```
[edit routing-instances VR-OAM]
user@host# set instance-type virtual router
user@host# set interface reth2.0
user@host# set interface st0.0
[edit routing-instances VR-3GPP]
user@host# set instance-type virtual router
user@host# set interface reth3.0
user@host# set interface st0.1
```

9. Specify security policies to permit site-to-site traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show chassis cluster`, `show interfaces`, `show security zones`, `show access profile radius_pico`, `show security ike`, `show security ipsec`, `show routing-instances`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis cluster
reth-count 5
node 0
```

```
node 1
redundancy-group 0{
    node 0 priority 250;
    node 1 priority 150;
    redundancy-group 1 {
    node 0 priority 220;
    node 1 priority 149;
    interface-monitor {
        ge-3/0/0 weight 255;
        ge-8/0/0 weight 255;
        ge-3/0/1 weight 255;
        ge-8/0/1 weight 255;
        ge-3/2/0 weight 255;
        ge-8/2/0 weight 255;
        ge-3/2/1 weight 255;
        ge-8/2/1 weight 255;
    }
}
[edit]
user@host# show interfaces
ge-3/0/0 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-3/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-3/2/0 {
    gigether-options {
        redundant-parent reth2;
    }
}
ge-3/2/1 {
    gigether-options {
        redundant-parent reth3;
    }
}
ge-8/0/0 {
    gigether-options {
        redundant-parent reth0;
```

```
    }
}
ge-8/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-8/2/0 {
    gigether-options {
        redundant-parent reth2;
    }
}
ge-8/2/1 {
    gigether-options {
        redundant-parent reth3;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.2.2.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.3.3.1/24;
        }
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
```

```
            address 192.168.2.20/24;
        }
    }
}
reth3 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.168.3.20/24;
        }
    }
}
st0 {
    unit 0{
        multipoint;
        family inet {
            address 12.12.1.20/24;
        }
    }
    unit 1{
        multipoint;
        family inet {
            address 13.13.1.20/24;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.0.0/16 next-hop 10.2.2.253;
    route 10.5.0.0/16 next-hop 10.2.2.253;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
```

```
        }
    }
    interfaces {
        reth1.0;
        reth0.0;
    }
}
security-zone oam-trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth2.0;
        st0.0;
    }
}
security-zone 3gpp-trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth3.0;
        st0.1;
    }
}
[edit]
user@host# show access profile radius_pico
authentication-order radius;
radius-server {
    192.168.2.22 {
        secret "$ABC123";
        routing-instance VR-OAM;
    }
```

```
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate example_SRX;
    }
}
gateway OAM_GW {
    ike-policy IKE_POL;
    dynamic {
        hostname .pico_cell.net;
        ike-user-type group-ike-id;
    }
    local-identity hostname srx_series.example.net;
    external-interface reth0.0;
    aaa access-profile radius_pico;
    version v2-only;
}
gateway 3GPP_GW {
    ike-policy IKE_POL;
    dynamic {
        distinguished-name {
            wildcard OU=pico_cell;
        }
        ike-user-type group-ike-id;
    }
    local-identity distinguished-name;
    external-interface reth1.0;
    aaa access-profile radius_pico;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
```

```
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 300;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn OAM_VPN {
    bind-interface st0.0;
    ike {
        gateway OAM_GW;
        proxy-identity {
            local 192.168.2.0/24;
            remote 0.0.0.0/0;
        }
        ipsec-policy IPSEC_POL;
    }
}
vpn 3GPP_VPN {
    bind-interface st0.1;
    ike {
        gateway 3GPP_GW;
        proxy-identity {
            local 192.168.3.0/24;
            remote 0.0.0.0/0;
        }
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show routing-instances
VR-OAM {
    instance-type virtual-router;
    interface reth2.0;
    interface st0.0;
}
VR-3GPP {
    instance-type virtual-router;
    interface reth3.0;
    interface st0.1;
```

```
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Intermediate Router

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.253/24
set interfaces ge-0/0/2 unit 0 family inet address 10.5.5.253/24
set interfaces ge-0/0/14 unit 0 family inet address 10.3.3.253/24
set interfaces ge-0/0/15 unit 0 family inet address 10.2.2.253/24
set routing-options static route 192.168.3.0/24 next-hop 10.2.2.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/14.0
set security zones security-zone trust interfaces ge-0/0/15.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/2.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the intermediate router:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.253/24
user@host# set ge-0/0/2 unit 0 family inet address 10.5.5.253/24
user@host# set ge-0/0/14 unit 0 family inet address 10.3.3.253/24
user@host# set ge-0/0/15 unit 0 family inet address 10.2.2.253/24
```

2. Configure routing options.

```
[edit routing-options]
user@host# set static route 192.168.3.0/24  next-hop 10.2.2.1
```

3. Specify security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
user@host# set interfaces ge-0/0/14.0
user@host# set interfaces ge-0/0/15.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces ge-0/0/2.0
```

4. Specify security policies.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.253/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.5.5.253/24;
        }
    }
}
ge-0/0/14 {
    unit 0 {
        family inet {
            address 10.3.3.253/24;
        }
    }
}
ge-0/0/15 {
    unit 0 {
        family inet {
            address 10.2.2.253/24;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 192.168.3.0/24 next-hop 10.2.2.1;
}
[edit]
```

```
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/14.0;
        ge-0/0/15.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        ge-0/0/2.0;
    }
}
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Pico Cell (Sample Configuration)

Step-by-Step Procedure

The pico cell information in this example is provided for reference. Detailed pico cell configuration information is beyond the scope of this document. The pico cell factory configuration must include the following information:

- Local certificate (X.509v3) and IKE identity information
- Traffic Selector (TSi, TSr) values set to any/any (0.0.0.0/0)
- SRX Series IKE identity information and public IP address
- Phase 1 and Phase 2 proposals that match the SRX Series configuration

The pico cells in this example use strongSwan open source software for IPsec-based VPN connections. This information is used by the SRX Series for pico cell provisioning using the IKEv2 configuration payload feature. In networks where many devices are being deployed, the pico cell configuration can be identical except for the certificate (leftcert) and identity (leftid) information. The following sample configurations illustrate factory settings.

1. Review the Pico 1 configuration:

   Pico 1: Sample Configuration

```
conn %default
        ikelifetime=8h
        keylife=1h
        rekeymargin=1m
        keyingtries=1
        keyexchange=ikev2
        authby=pubkey
        mobike=no

conn oam
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid=pico1.pico_cell.net
        leftfirewall=yes
        reauth=yes
        right=10.2.2.1/24
        rightid=srx_series.example.net
```

```
        rightsubnet=0.0.0.0/0 #peer net for proxy id
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add

conn 3gpp
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid="C=US, ST=CA, L=Sunnyvale, O=org, OU=pico_cell, CN=pico1"
        leftfirewall=yes
        reauth=yes
        right=10.3.3.1/24
        rightid="OU=srx_series"
        rightsubnet=0.0.0.0/0 #peer net for proxy id
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add
```

2. Review the Pico 2 configuration:

Pico 2 Sample Configuration

```
conn %default
        ikelifetime=8h
        keylife=1h
        rekeymargin=1m
        keyingtries=1
        keyexchange=ikev2
        authby=pubkey
        mobike=no

conn oam
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid=pico2.pico_cell.net
        leftfirewall=yes
        #reauth=no
        right=10.2.2.1/24
        rightid=srx_series.example.net
        rightsubnet=0.0.0.0/0 #peer net for proxy id
```

```
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add

conn 3gpp
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid=”C=US, ST=CA, L=Sunnyvale, O=org, OU=pico_cell, CN=pico2”
        leftfirewall=yes
        #reauth=no
        right=10.3.3.1/24
        rightid=”OU=srx_series”
        rightsubnet=0.0.0.0/0 #peer net for proxy id
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add
```

Configuring the RADIUS Server (Sample Configuration using a FreeRADIUS)

Step-by-Step Procedure

The RADIUS server information in this example is provided for reference. Complete RADIUS server configuration information is beyond the scope of this document. The following information is returned to the SRX Series by the RADIUS server:

- Framed-IP-Address
- Framed-IP-Netmask (optional)
- Primary-DNS and Secondary-DNS (optional)

In this example, the RADIUS server has separate provisioning information for the OAM and 3GPP connections. The User-Name is taken from the client certificate information provided in the SRX Series authorization request.

If the RADIUS server acquires client provisioning information from a DHCP server, the client identity information relayed to the DHCP server by the RADIUS server must be consistent with the client IKE identity information relayed to the RADIUS server by the SRX Series Firewall. This ensures the continuity of the client identity across the various protocols.

The communication channel between the SRX Series Firewall and the RADIUS server is protected by a RADIUS shared secret.

1. Review the RADIUS configuration for the Pico 1 OAM VPN. The RADIUS server has the following information:

   Sample RADIUS configuration in Junos OS Releases 12.3X48 and Junos OS releases prior to 15.1X49-D160, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

   FreeRADIUS configuration example:

   ```
   DEFAULT User-Name =~ "device@example.net", Cleartext-Password := "juniper"
           Service-Type = Framed-User,
           Framed-IP-Address = 10.12.1.201,
           Framed-IP-Netmask = 255.255.255.255,
           Primary-Dns = 192.168.2.104,
           Secondary-Dns = 192.168.2.106,
   ```

   Sample RADIUS configuration starting from Junos OS Releases 15.X49-D161, 15.1X49-D170, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

   FreeRADIUS configuration example:

   ```
   DEFAULT User-Name =~ "device@example.net", Auth-Type := "Accept"
           Service-Type = Framed-User,
           Framed-IP-Address = 10.12.1.201,
           Framed-IP-Netmask = 255.255.255.255,
           Primary-Dns = 192.168.2.104,
           Secondary-Dns = 192.168.2.106,
   ```

   In this case, the RADIUS server provides the default subnet mask (255.255.255.255), which blocks intrapeer traffic.

2. Review the RADIUS configuration for the Pico 1 3GPP VPN. The RADIUS server has the following information:

   Sample RADIUS configuration in Junos OS Releases 12.3X48 and Junos OS releases prior to 15.1X49-D160, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

   FreeRADIUS configuration example:

   ```
   DEFAULT User-Name =~ "device@example.net", Cleartext-Password := "juniper"
          Service-Type = Framed-User,
           Framed-IP-Address = 10.13.1.201.10,
           Framed-IP-Netmask = 255.255.0.0,
   ```

```
        Primary-Dns = 192.168.2.104,
        Secondary-Dns = 192.168.2.106,
```

Sample RADIUS configuration starting from Junos OS Releases 15.X49-D161, 15.1X49-D170, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

FreeRADIUS configuration example:

```
DEFAULT User-Name =~ "device@example.net", Auth-Type := "Accept"
       Service-Type = Framed-User,
        Framed-IP-Address = 10.13.1.201.10,
        Framed-IP-Netmask = 255.255.0.0,
        Primary-Dns = 192.168.2.104,
        Secondary-Dns = 192.168.2.106,
```

In this case, the RADIUS server provides a subnet mask value (255.255.0.0), which enables intrapeer traffic.

Starting in Junos OS Release 20.1R1, you can configure a common password for IKEv2 configuration payload requests for an IKE gateway configuration. The common password in the range of 1 to 128 characters allows the administrator to define a common password. This password is used between the SRX Series Firewall and the RADIUS server when the SRX Series Firewall requesting an IP address on behalf of a remote IPsec peer using IKEv2 configuration payload. RADIUS server validate the credentials before it provides any IP information to the SRX Series Firewall for the configuration payload request. You can configure the common password using `config-payload-password` *`configured-password`* configuration statement at `[edit security ike gateway gateway-name aaa access-profile access-profile-name]` hierarchy level. Additionally, this example creates two tunnels from the same client certificate by using different parts of the certificate for User-Name (IKE identity) information.

Verification

IN THIS SECTION

- Verifying the IKE Phase 1 Status for the SRX Series | **402**
- Verifying IPsec Security Associations for the SRX Series | **404**

Confirm that the configuration is working properly.

Verifying the IKE Phase 1 Status for the SRX Series

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode on node 0, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations detail** command.

```
user@host# show security ike security-associations
node0:
--------------------------------------------------------------------------
Index      State  Initiator cookie  Responder cookie  Mode  Remote Address
553329718  UP     99919a471d1a5278  3be7c5a49172e6c2  IKEv2 10.1.1.1
1643848758 UP     9e31d4323195a195  4d142438106d4273  IKEv2 10.1.1.1
```

```
user@host# show security ike security-associations index 553329718 detail
node0:
--------------------------------------------------------------------------
IKE peer 10.1.1.1, Index 553329718, Gateway Name: OAM_GW
  Location: FPC 2, PIC 0, KMD-Instance 1
  Role: Responder, State: UP
  Initiator cookie: 99919a471d1a5278, Responder cookie: 3be7c5a49172e6c2
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 10.2.2.1:500, Remote: 10.1.1.1:500
  Lifetime: Expires in 28738 seconds
  Peer ike-id: C=US, ST=CA, L=Sunnyvale, O=org, OU=pico_cell, CN=pico1
  aaa assigned IP: 10.12.1.201
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                2104
   Output bytes  :                 425
  Input  packets:                    2
```

```
 Output packets:                    1
IPSec security associations: 0 created, 0 deleted
Phase 2 negotiations in progress: 1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs with pico cells devices. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. This example shows only the IKE Phase 1 SA for the OAM VPN; however, a separate IKE Phase 1 SA will be displayed showing the IKE Phase 1 parameters for the 3GPP VPN.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA: you can use the `show security ike security-associations index detail` command to get more information about the SA.
- Remote address—Verify that the local IP address is correct and that port 500 is being used for peer-to-peer communication.
- Role responder state:
  - Up—The Phase 1 SA has been established.
  - Down—There was a problem establishing the Phase 1 SA.
- Peer (remote) IKE ID—Verify the certificate information is correct.
- Local identity and remote identity—Verify these addresses are correct.
- Mode—Verify that the correct mode is being used.

Verify that the following items are correct in your configuration:

- External interfaces (the interface must be the one that sends IKE packets)
- IKE policy parameters
- Phase 1 proposal parameters (must match between peers)

The `show security ike security-associations` command lists the following additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)

- Role information

  Troubleshooting is best performed on the peer using the responder role.

- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations for the SRX Series

Purpose

Verify the IPsec status.

Action

From operational mode on node 0, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations detail** command.

```
user@host# show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID         Algorithm             SPI     Life:sec/kb Mon lsys Port Gateway
  <214171651 ESP:aes-cbc-256/sha1 cc2869e2 3529/        -  root 500  10.1.1.1
  >214171651 ESP:aes-cbc-256/sha1 c0a54936 3529/        -  root 500  10.1.1.1
  <205520899 ESP:aes-cbc-256/sha1 84e49026 3521/        -  root 500  10.1.1.1
  >205520899 ESP:aes-cbc-256/sha1 c4ed1849 3521/        -  root 500  10.1.1.1
```

```
user@host# show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------
Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x604a29
Last Tunnel Down Reason: SA not initiated
  ID: 214171651 Virtual-system: root, VPN Name: 3GPP_VPN
  Local Gateway: 10.3.3.1, Remote Gateway: 10.1.1.1
  Local Identity: list(any:0,ipv4_subnet(any:0-65535,[0..7]=192.168.3.0/24),
ipv4_subnet(any:0-65535,[0..7]=10.13.0.0/16))
  Remote Identity: ipv4(any:0,[0..3]=10.13.1.201)
```

```
  DF-bit: clear
  Bind-interface: st0.1

Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
Last Tunnel Down Reason: SA not initiated
  Location: FPC 6, PIC 0, KMD-Instance 2
  Direction: inbound, SPI: cc2869e2, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3523 seconds
  Lifesize Remaining:
  Soft lifetime: Expires in 2965 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64

  Location: FPC 6, PIC 0, KMD-Instance 2
  Direction: outbound, SPI: c0a54936, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3523 seconds
  Lifesize Remaining:
  Soft lifetime: Expires in 2965 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64

ID: 205520899 Virtual-system: root, VPN Name: OAM_VPN
Local Gateway: 10.2.2.1, Remote Gateway: 10.1.1.1
Local Identity: ipv4_subnet(any:0-65535,[0..7]=192.168.2.0/24)
Remote Identity: ipv4(any:0,[0..3]=10.12.1.201)
Version: IKEv2
  DF-bit: clear
  Bind-interface: st0.0

Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
Last Tunnel Down Reason: SA not initiated
  Location: FPC 2, PIC 0, KMD-Instance 1
  Direction: inbound, SPI: 84e49026, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3515 seconds
  Lifesize Remaining:
  Soft lifetime: Expires in 2933 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
```

```
    Anti-replay service: counter-based enabled, Replay window size: 64

    Location: FPC 2, PIC 0, KMD-Instance 1
    Direction: outbound, SPI: c4ed1849, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3515 seconds
    Lifesize Remaining:
    Soft lifetime: Expires in 2933 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

This examples shows the active IKE Phase 2 SAs for Pico 1. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IPsec policy parameters in your configuration. For each Phase 2 SA (OAM and 3GPP), information is provided in both the inbound and outboard direction. The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an IP address of 10.1.1.1.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3529/ value indicates that the Phase 2 lifetime expires in 3529 seconds, and that no lifesize has been specified, which indicates that it is unlimited. The Phase 2 lifetime can differ from the Phase 1 lifetime, because Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

The above output from the `show security ipsec security-associations index` *`index_id`* `detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Authentication and encryption algorithms used.

- Phase 2 proposal parameters (must match between peers).
- Secure tunnel (st0.0 and st0.1) bindings to the OAM and 3GPP gateways.

SEE ALSO

IPsec Overview | 12

PKI in Junos

IKE Policy with a Trusted CA

This example shows how to bind a trusted CA server to an IKE policy of the peer.

Before you begin, you must have a list of all the trusted CAs you want to associate with the IKE policy of the peer.

You can associate an IKE policy to a single trusted CA profile or a trusted CA group. For establishing a secure connection, the IKE gateway uses the IKE policy to limit itself to the configured group of CAs (ca-profiles) while validating the certificate. A certificate issued by any source other than the trusted CA or trusted CA group is not validated. If there is a certificate validation request coming from an IKE policy then the associated CA profile of the IKE policy will validate the certificate. If an IKE policy is not associated with any CA then by default the certificate is validated by any one of the configured CA profiles.

In this example, a CA profile named `root-ca` is created and a `root-ca-identity` is associated to the profile.

You can configure a maximum of 20 CA profiles that you want to add to a trusted CA group. You cannot commit your configuration if you configure more than 20 CA profiles in a trusted CA group.

1. Create a CA profile and associate a CA identifier to the profile.

```
[edit]
user@host# set security pki ca-profile root-ca ca-identity root-ca
```

2. Define an IKE proposal and the IKE proposal authentication method.

```
[edit]
user@host# set security ike proposal ike_prop authentication-method rsa-signatures
```

3. Define the Diffie-Hellman group, authentication algorithm, an encryption algorithm for the IKE proposal.

```
[edit]
user@host# set security ike proposal ike_prop dh-group group2
user@host# set security ike proposal ike_prop authentication-algorithm sha-256
user@host# set security ike proposal ike_prop encryption-algorithm aes-256-cbc
```

4. Configure an IKE policy and associate the policy with the IKE proposal.

```
[edit]
user@host# set security ike policy ike_policy proposals ike_prop
```

5. Configure a local certificate identifier for the IKE policy.

```
[edit]
user@host# set security ike policy ike_policy certificate local-certificate SPOKE
```

6. Define the CA to be used for the IKE policy.

```
[edit]
user@host# set security ike policy ike_policy certificate trusted-ca ca-profile root-ca
```

To view the CA profiles and the trusted CA groups configured on your device, run `show security pki` command.

```
user@host# show security ike
    proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy ike_policy {
    proposals ike_prop;
    certificate {
        local-certificate SPOKE;
        trusted-ca ca-profile root-ca;
```

```
    }
}
```

The `show security ike` command displays the CA profile group under the IKE policy named `ike_policy` and the certificate associated with the IKE policy.

SEE ALSO

Basic Elements of PKI in Junos OS

Secure Tunnel Interface in a Virtual Router

SUMMARY

Read this topic to learn about secure tunnel (st0) interface in a virtual router.

IN THIS SECTION

- Understanding Virtual Router Support for Route-Based VPNs | **409**
- Example: Configuring an st0 Interface in a Virtual Router | **411**

A secure tunnel interface (st0) is an internal interface that is used by route-based VPNs to route cleartext traffic to an IPsec VPN tunnel.

Understanding Virtual Router Support for Route-Based VPNs

IN THIS SECTION

- Understanding Virtual Router Limitations | **410**

This feature includes routing-instance support for route-based VPNs. In previous releases, when an st0 interface was put in a nondefault routing instance, the VPN tunnels on this interface did not work

properly. You can place the st0 interface in a routing instance and configure each unit in point-to-point mode or multipoint mode. Therefore, VPN traffic now works correctly in a nondefault VR. You can now configure different subunits of the st0 interface in different routing instances. The following functions are supported for nondefault routing instances:

- Manual key management
- Transit traffic
- Self-traffic
- VPN monitoring
- Hub-and-spoke VPNs
- Encapsulating Security Payload (ESP) protocol
- Authentication Header (AH) protocol
- Aggressive mode or main mode
- st0 anchored on the loopback (lo0) interface
- Maximum number of virtual routers (VRs) supported on an SRX Series Firewall
- Applications such as Application Layer Gateway (ALG), Intrusion Detection and Prevention (IDP), and Content Security
- Dead peer detection (DPD)
- *Chassis cluster* active/backup
- Open Shortest Path First (OSPF) over st0
- Routing Information Protocol (RIP) over st0

Understanding Virtual Router Limitations

When you configure VPN on SRX Series Firewalls, overlapping of IP addresses across virtual routers is supported with the following limitations:

- An IKE external interface address cannot overlap with any other virtual router.
- An internal or trust interface address can overlap across any other virtual router.
- An st0 interface address cannot overlap in route-based VPN in point-to-multipoint tunnels such as NHTB.
- An st0 interface address can overlap in route-based VPN in point-to-point tunnels.

SEE ALSO

IPsec Overview | 12

Example: Configuring an st0 Interface in a Virtual Router

IN THIS SECTION

- Requirements | 411
- Overview | 411
- Configuration | 414
- Verification | 420

This example shows how to configure an st0 interface in a virtual router.

Requirements

Before you begin, configure the interfaces and assign the interfaces to security zones. See "*Security Zones Overview*".

Overview

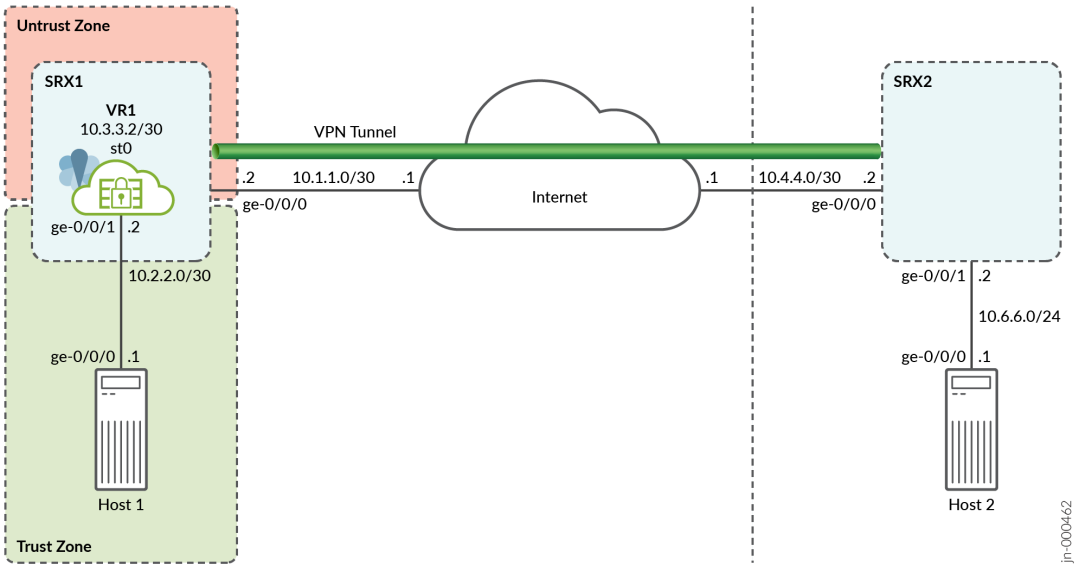
In this example, you perform the following operations:

- Configure the interfaces.
- Configure IKE Phase 1 proposals.
- Configure IKE policies, and reference the proposals.
- Configure an IKE gateway, and reference the policy.
- Configure Phase 2 proposals.
- Configure policies, and reference the proposals.
- Configure AutoKey IKE, and reference the policy and gateway.
- Configure the security policy.
- Configure the routing instance.

- Configure the VPN bind to tunnel interface.
- Configure the routing options.

Figure 25 on page 412 shows the topology used in this example.

Figure 25: Secure Tunnel Interface in a Virtual Router



Following tables show the configuration parameters.

Table 68: Interface, Routing Instance, Static Route, and Security Zone Information for SRX1

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	10.1.1.2/30
	ge-0/0/1.0	10.2.2.2/30
	st0.0 (tunnel interface)	10.3.3.2/30
Routing instance (Virtual Router)	VR1	ge-0/0/1.0 st0.0

Table 68: Interface, Routing Instance, Static Route, and Security Zone Information for SRX1 ***(Continued)***

Feature	Name	Configuration Parameters
Static routes	10.6.6.0/24	The next hop is st0.0.
Security zones	trust	• The ge-0/0/1 interface is bound to this zone.
	untrust	• The ge-0/0/0 interface is bound to this zone. • The st0.0 interface is bound to this zone.

Table 69: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	first_ikeprop	• Authentication method: pre-shared-keys
Policy	first_ikepol	• Mode: main • Proposal reference: first_ikeprop • IKE policy authentication method: pre-shared-keys
Gateway	first	• IKE policy reference: first_ikepol • External interface: ge-0/0/0.0 • Gateway address: 10.4.4.2

Table 70: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	first_ipsecprop	• protocol: esp • authentication-algorithm: hmac-md5-96 • encryption-algorithm: 3des-cbc
Policy	first_ipsecpol	• IPsec proposal reference: first_ipsecprop
VPN	first_vpn	• IKE gateway reference: first • IPsec policy reference: first_ipsecpol • Bind to interface: st0.0 • establish-tunnels immediately

Configuration

IN THIS SECTION

- Procedure | 414

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.2/30
set interfaces ge-0/0/1 unit 0 family inet address 10.2.2.2/30
set interfaces st0 unit 0 family inet address 10.3.3.2/30
```

```
set security zones security-zone trust interfaces ge-0/0/1
set security zones security-zone untrust interfaces ge-0/0/0
set security zones security-zone untrust interfaces st0.0
set security ike proposal first_ikeprop authentication-method pre-shared-keys
set security ike proposal first_ikeprop dh-group group2
set security ike proposal first_ikeprop authentication-algorithm md5
set security ike proposal first_ikeprop encryption-algorithm 3des-cbc
set security ike policy first_ikepol mode main
set security ike policy first_ikepol proposals first_ikeprop
set security ike policy first_ikepol pre-shared-key ascii-text "$ABC123"
set security ike gateway first ike-policy first_ikepol
set security ike gateway first address 10.4.4.2
set security ike gateway first external-interface ge-0/0/0.0
set security ipsec proposal first_ipsecprop protocol esp
set security ipsec proposal first_ipsecprop authentication-algorithm hmac-md5-96
set security ipsec proposal first_ipsecprop encryption-algorithm 3des-cbc
set security ipsec policy first_ipsecpol perfect-forward-secrecy keys group1
set security ipsec policy first_ipsecpol proposals first_ipsecprop
set security ipsec vpn first_vpn bind-interface st0.0
set security ipsec vpn first_vpn ike gateway first
set security ipsec vpn first_vpn ike ipsec-policy first_ipsecpol
set security ipsec vpn first_vpn establish-tunnels immediately
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
set security policies from-zone trust to-zone untrust policy p1 then permit
set security policies from-zone untrust to-zone trust policy p2 match source-address any
set security policies from-zone untrust to-zone trust policy p2 match destination-address any
set security policies from-zone untrust to-zone trust policy p2 match application any
set security policies from-zone untrust to-zone trust policy p2 then permit
set routing-instances VR1 instance-type virtual-router
set routing-instances VR1 interface ge-0/0/1.0
set routing-instances VR1 interface st0.0
set routing-instances VR1 routing-options static route 10.6.6.0/24 next-hop st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure an st0 in a VR:

1. Configure the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.2/30
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.2.2.2/30
user@host# set interfaces st0 unit 0 family inet address 10.3.3.2/30
```

2. Configure security zones.

```
[edit]
user@host# set security zones security-zone trust interfaces ge-0/0/1
user@host# set security zones security-zone untrust interfaces ge-0/0/0
user@host# set security zones security-zone untrust interfaces st0.0
```

3. Configure Phase 1 of the IPsec tunnel.

```
[edit security ike]
user@host# set proposal first_ikeprop authentication-method pre-shared-keys
user@host# set proposal first_ikeprop dh-group group2
user@host# set proposal first_ikeprop authentication-algorithm md5
user@host# set proposal first_ikeprop encryption-algorithm 3des-cbc
```

4. Configure the IKE policies, and reference the proposals.

```
[edit security ike]
user@host# set policy first_ikepol mode main
user@host# set policy first_ikepol proposals first_ikeprop
user@host# set policy first_ikepol pre-shared-key ascii-text "$ABC123"
```

5. Configure the IKE gateway, and reference the policy.

```
[edit security ike]
user@host# set gateway first ike-policy first_ikepol
user@host# set gateway first address 10.4.4.2
user@host# set gateway first external-interface ge-0/0/0.0
```

6. Configure Phase 2 of the IPsec tunnel.

```
[edit security ipsec]
user@host# set proposal first_ipsecprop protocol esp
user@host# set proposal first_ipsecprop authentication-algorithm hmac-md5-96
user@host# set proposal first_ipsecprop encryption-algorithm 3des-cbc
```

7. Configure the policies, and reference the proposals.

```
[edit security ipsec]
user@host# set policy first_ipsecpol perfect-forward-secrecy keys group1
user@host# set policy first_ipsecpol proposals first_ipsecprop
```

8. Configure AutoKey IKE, and reference the policy and gateway.

```
[edit security ipsec]
user@host# set vpn first_vpn ike gateway first
user@host# set vpn first_vpn ike ipsec-policy first_ipsecpol
user@host# set vpn first_vpn establish-tunnels immediately
```

9. Configure the VPN bind to tunnel interface.

```
[edit security ipsec]
user@host# set vpn first_vpn bind-interface st0.0
```

10. Configure the security policy.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy p1 match source-address any
user@host# set from-zone trust to-zone untrust policy p1 match destination-address any
user@host# set from-zone trust to-zone untrust policy p1 match application any
user@host# set from-zone trust to-zone untrust policy p1 then permit
user@host# set from-zone untrust to-zone trust policy p2 match source-address any
user@host# set from-zone untrust to-zone trust policy p2 match destination-address any
user@host# set from-zone untrust to-zone trust policy p2 match application any
user@host# set from-zone untrust to-zone trust policy p2 then permit
```

11. Configure the st0 in the routing instance.

```
[edit routing-instances]
user@host# set VR1 instance-type virtual-router
user@host# set VR1 interface ge-0/0/1.0
user@host# set VR1 interface st0.0
```

12. Configure the routing options.

```
[edit routing-instances VR1 routing-options]
user@host# set static route 10.6.6.0/24 next-hop st0.0
```

Results

From configuration mode, confirm your configuration by entering the `show security` and `show routing-instances` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security
    ike {
    proposal first_ikeprop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm 3des-cbc;
    }
    policy first_ikepol {
        mode main;
        proposals first_ikeprop;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway first {
        ike-policy first_ikepol;
        address 10.4.4.2;
        external-interface ge-0/0/0.0;
    }
}
    ipsec {
        proposal first_ipsecprop {
```

```
            protocol esp;
            authentication-algorithm hmac-md5-96;
            encryption-algorithm 3des-cbc;
        }
        policy first_ipsecpol {
            perfect-forward-secrecy {
                keys group1;
            }
            proposals first_ipsecprop;
        }
        vpn first_vpn {
            bind-interface st0.0;
            ike {
                gateway first;
                ipsec-policy first_ipsecpol;
            }
            establish-tunnels immediately;
        }
    }
policies {
    from-zone trust to-zone untrust {
        policy p1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
        from-zone untrust to-zone trust {
            policy p2 {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
```

```
        }
zones {
    security-zone trust {
        interfaces {
            ge-0/0/1.0;
        }
    }
    security-zone untrust {
        interfaces {
            ge-0/0/0.0;
            st0.0;
        }
    }
}

user@host# show routing-instances
        VR1 {
            instance-type virtual-router;
            interface ge-0/0/1.0;
            interface st0.0;
            routing-options {
            static {
            route 10.6.6.0/24 next-hop st0.0;
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying an st0 interface in the Virtual Router | **421**

To confirm that the configuration is working properly, perform this task:

Verifying an st0 interface in the Virtual Router

Purpose

Verify the st0 interface in the virtual router.

Action

From operational mode, enter the `show interfaces st0.0 detail` command. The number listed for routing table corresponds to the order that the routing tables in the `show route all` command.

SEE ALSO

Understanding Virtual Router Support for Route-Based VPNs | 409

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
10.4	Starting in Junos OS 10.4 release, the firewall supports placing the st0 interface in a routing instance.

RELATED DOCUMENTATION

Route-Based IPsec VPNs | 320

Dual Stack Tunnels over an External Interface

SUMMARY

Learn about dual-stack tunnels in IPsec VPNs.

IN THIS SECTION

Understanding VPN Tunnel Modes | 422

- Example: Configuring Dual-Stack Tunnels over an External Interface | 425

Dual-stack tunnels—parallel IPv4 and IPv6 tunnels over a single physical interface to a peer—are supported for route-based site-to-site VPNs. A physical interface configured with both IPv4 and IPv6 addresses can be used as an external interface for IPv4 and IPv6 gateways on the same peer or on different peers at the same time.

Understanding VPN Tunnel Modes

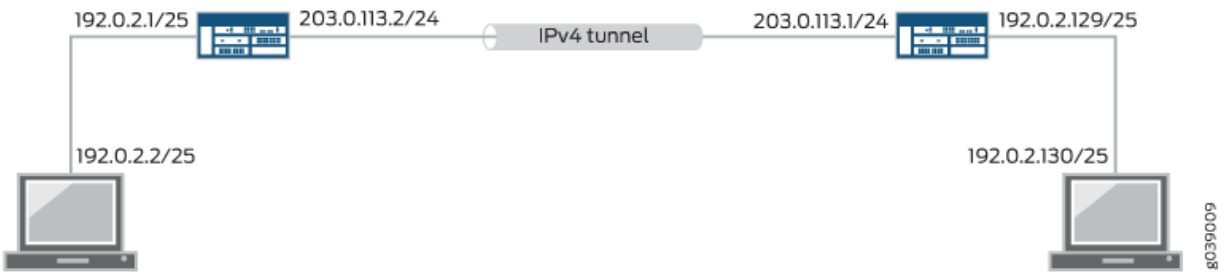
IN THIS SECTION

- Understanding Dual-Stack Tunnels over an External Interface | 424

In VPN tunnel mode, IPsec encapsulates the original IP datagram—including the original IP header—within a second IP datagram. The outer IP header contains the IP address of the gateway, while the inner header contains the ultimate source and destination IP addresses. The outer and inner IP headers can have a protocol field of IPv4 or IPv6. The firewalls support four tunnel modes for route-based site-to-site VPNs.

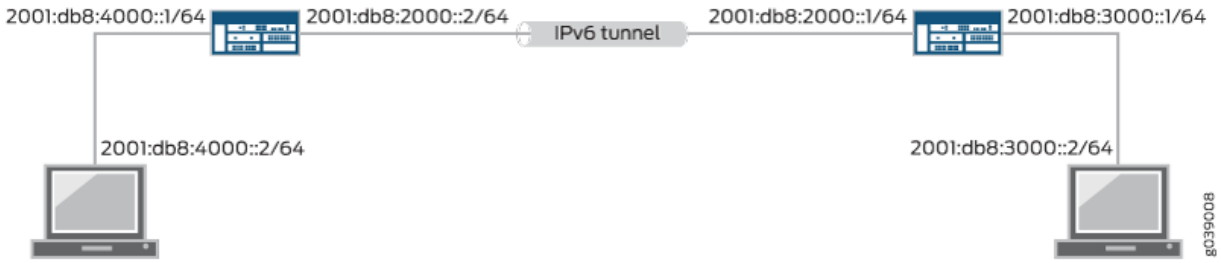
IPv4-in-IPv4 tunnels encapsulate IPv4 packets inside IPv4 packets, as shown in Figure 26 on page 422. The protocol fields for both the outer and the inner headers are IPv4.

Figure 26: IPv4-in-IPv4 Tunnel



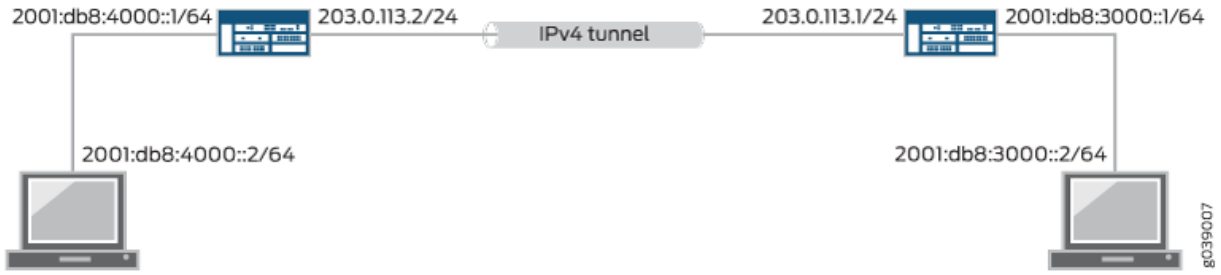
IPv6-in-IPv6 tunnels encapsulate IPv6 packets inside IPv6 packets, as shown in Figure 27 on page 423. The protocol fields for both the outer and inner headers are IPv6.

Figure 27: IPv6-in-IPv6 Tunnel



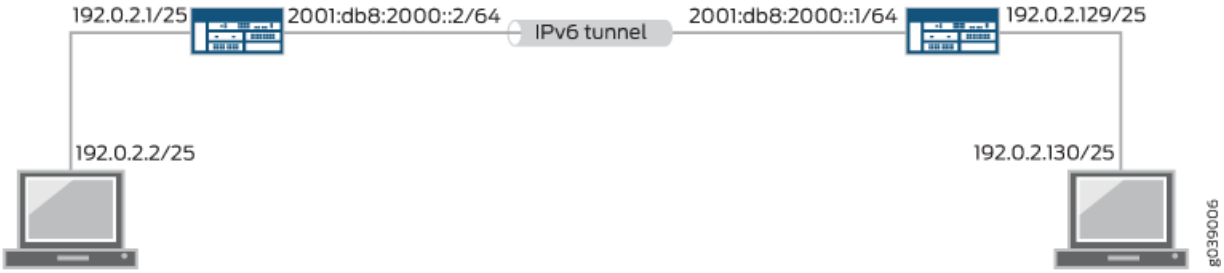
IPv6-in-IPv4 tunnels encapsulate IPv6 packets inside IPv4 packets, as shown in Figure 28 on page 423. The protocol field for the outer header is IPv4 and the protocol field for the inner header is IPv6.

Figure 28: IPv6-in-IPv4 Tunnel



IPv4-in-IPv6 tunnels encapsulate IPv4 packets inside IPv6 packets, as shown in Figure 29 on page 424. The protocol field for the outer header is IPv6 and the protocol field for the inner header is IPv4.

Figure 29: IPv4-in-IPv6 Tunnel



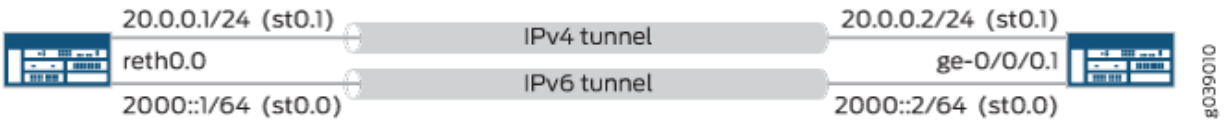
A single IPsec VPN tunnel can carry both IPv4 and IPv6 traffic. For example, an IPv4 tunnel can operate in both IPv4-in-IPv4 and IPv6-in-IPv4 tunnel modes at the same time. To allow both IPv4 and IPv6 traffic over a single IPsec VPN tunnel, the st0 interface bound to that tunnel must be configured with both `family inet` and `family inet6`.

A physical interface configured with both IPv4 and IPv6 addresses can be used as the external interface for parallel IPv4 and IPv6 tunnels to a peer in a route-based site-to-site VPN. This feature is known as *dual-stack tunnels* and requires separate st0 interfaces for each tunnel.

Understanding Dual-Stack Tunnels over an External Interface

Dual-stack tunnels—parallel IPv4 and IPv6 tunnels over a single physical interface to a peer—are supported for route-based site-to-site VPNs. A physical interface configured with both IPv4 and IPv6 addresses can be used as the external interface to IPv4 and IPv6 gateways on the same peer or on different peers at the same time. In Figure 30 on page 424, the physical interfaces reth0.0 and ge-0/0/0.1 support parallel IPv4 and IPv6 tunnels between two devices.

Figure 30: Dual-Stack Tunnels



In Figure 30 on page 424, separate secure tunnel (st0) interfaces must be configured for each IPsec VPN tunnel. Parallel IPv4 and IPv6 tunnels that are bound to the same st0 interface are not supported.

A single IPsec VPN tunnel can carry both IPv4 and IPv6 traffic. For example, an IPv4 tunnel can operate in both IPv4-in-IPv4 and IPv6-in-IPv4 tunnel modes at the same time. To allow both IPv4 and IPv6

traffic over a single IPsec VPN tunnel, the st0 interface bound to that tunnel must be configured with both `family inet` and `family inet6`.

If multiple addresses in the same address family are configured on the same external interface to a VPN peer, we recommend that you configure `local-address` at the [`edit security ike gateway` *`gateway-name`*] hierarchy level.

If `local-address` is configured, the specified IPv4 or IPv6 address is used as the local gateway address. If only one IPv4 and one IPv6 address is configured on a physical external interface, `local-address` configuration is not required.

The `local-address` value must be an IP address that is configured on an interface on the firewall. We recommend that `local-address` belong to the external interface of the IKE gateway. If `local-address` does not belong to the external interface of the IKE gateway, the interface must be in the same zone as the external interface of the IKE gateway and an intra-zone security policy must be configured to permit traffic.

The `local-address` value and the remote IKE gateway address must be in the same address family, either IPv4 or IPv6.

If `local-address` is not configured, the local gateway address is based on the remote gateway address. If the remote gateway address is an IPv4 address, the local gateway address is the primary IPv4 address of the external physical interface. If the remote gateway address is an IPv6 address, the local gateway address is the primary IPv6 address of the external physical interface.

SEE ALSO

VPN Feature Support for IPv6 Addresses | **153**

Understanding IPv6 IKE and IPsec Packet Processing | **159**

VPN Feature Support for IPv6 Addresses | **153**

Example: Configuring Dual-Stack Tunnels over an External Interface

IN THIS SECTION

- Requirements | **426**
- Overview | **426**
- Configuration | **430**

- Verification | 436

This example shows how to configure parallel IPv4 and IPv6 tunnels over a single external physical interface to a peer for route-based site-to-site VPNs.

Requirements

Before you begin, read "Understanding VPN Tunnel Modes" on page 422.

The configuration shown in this example is only supported with route-based site-to-site VPNs.

Overview

IN THIS SECTION

- Topology | 429

In this example, a redundant Ethernet interface on the local device supports parallel IPv4 and IPv6 tunnels to a peer device:

- The IPv4 tunnel carries IPv6 traffic; it operates in IPv6-in-IPv4 tunnel mode. The secure tunnel interface st0.0 bound to the IPv4 tunnel is configured with family inet6 only.
- The IPv6 tunnel carries both IPv4 and IPv6 traffic; it operates in both IPv4-in-IPv6 and IPv6-in-IPv6 tunnel modes. The secure tunnel interface st0.1 bound to the IPv6 tunnel is configured with both family inet and family inet6.

Table 71 on page 426 shows the Phase 1 options used in this example. The Phase 1 option configuration includes two IKE gateway configurations, one to the IPv6 peer and the other to the IPv4 peer.

Table 71: Phase 1 Options for Dual-Stack Tunnel Configuration

Option	Value
IKE proposal	ike_proposal
Authentication method	Preshared keys

Table 71: Phase 1 Options for Dual-Stack Tunnel Configuration *(Continued)*

Option	Value
Authentication algorithm	MD5
Encryption algorithm	3DES CBC
Lifetime	3600 seconds
IKE policy	ike_policy
Mode	Aggressive
IKE proposal	ike_proposal
Preshared key	ASCII text
IPv6 IKE gateway	ike_gw_v6
IKE policy	ike_policy
Gateway address	2000::2
External interface	reth1.0
IKE version	IKEv2
IPv4 IKE gateway	ike_gw_v4
IKE policy	ike_policy
Gateway address	20.0.0.2

Table 71: Phase 1 Options for Dual-Stack Tunnel Configuration *(Continued)*

Option	Value
External interface	reth1.0

Table 72 on page 428 shows the Phase 2 options used in this example. The Phase 2 option configuration includes two VPN configurations, one for the IPv6 tunnel and the other for the IPv4 tunnel.

Table 72: Phase 2 Options for Dual-Stack Tunnel Configuration

Option	Value
IPsec proposal	ipsec_proposal
Protocol	ESP
Authentication algorithm	HMAC SHA-1 96
Encryption algorithm	3DES CBC
IPsec policy	ipsec_policy
Proposal	ipsec_proposal
IPv6 VPN	test_s2s_v6
Bind interface	st0.1
IKE gateway	ike_gw_v6
IKE IPsec policy	ipsec_policy
Establish tunnels	Immediately
IPv4 VPN	test_s2s_v4

Table 72: Phase 2 Options for Dual-Stack Tunnel Configuration *(Continued)*

Option	Value
Bind interface	st0.0
IKE gateway	ike_gw_4
IKE IPsec policy	ipsec_policy

The following static routes are configured in the IPv6 routing table:

- Route IPv6 traffic to 3000::1/128 through st0.0.
- Route IPv6 traffic to 3000::2/128 through st0.1.

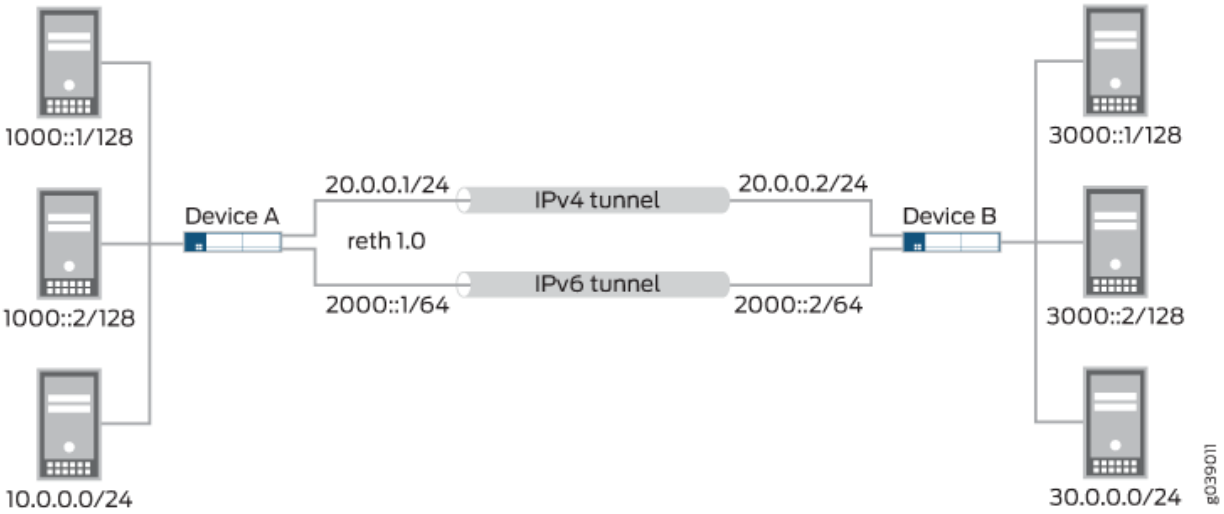
A static route is configured in the default (IPv4) routing table to route IPv4 traffic to 30.0.0.0/24 through st0.1.

Flow-based processing of IPv6 traffic must be enabled with the `mode flow-based` configuration option at the [`edit security forwarding-options family inet6`] hierarchy level.

Topology

In Figure 31 on page 430, the device A supports IPv4 and IPv6 tunnels to device B. IPv6 traffic to 3000::1/128 is routed through the IPv4 tunnel, while IPv6 traffic to 3000::2/128 and IPv4 traffic to 30.0.0.0/24 are routed through the IPv6 tunnel.

Figure 31: Dual-Stack Tunnel Example



Configuration

IN THIS SECTION

- Procedure | 430

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 gigether-options redundant-parent reth1
set interfaces ge-8/0/1 gigether-options redundant-parent reth1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 20.0.0.1/24
set interfaces reth1 unit 0 family inet6 address 2000::1/64
set interfaces st0 unit 0 family inet6
set interfaces st0 unit 1 family inet
```

```
set interfaces st0 unit 1 family inet6
set security ike proposal ike_proposal authentication-method pre-shared-keys
set security ike proposal ike_proposal authentication-algorithm md5
set security ike proposal ike_proposal encryption-algorithm 3des-cbc
set security ike proposal ike_proposal lifetime-seconds 3600
set security ike policy ike_policy mode aggressive
set security ike policy ike_policy proposals ike_proposal
set security ike policy ike_policy pre-shared-key ascii-text "$ABC123"
set security ike gateway ike_gw_v6 ike-policy ike_policy
set security ike gateway ike_gw_v6 address 2000::2
set security ike gateway ike_gw_v6 external-interface reth1.0
set security ike gateway ike_gw_v6 version v2-only
set security ike gateway ike_gw_v4 ike-policy ike_policy
set security ike gateway ike_gw_v4 address 20.0.0.2
set security ike gateway ike_gw_v4 external-interface reth1.0
set security ipsec proposal ipsec_proposal protocol esp
set security ipsec proposal ipsec_proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_proposal encryption-algorithm 3des-cbc
set security ipsec policy ipsec_policy proposals ipsec_proposal
set security ipsec vpn test_s2s_v6 bind-interface st0.1
set security ipsec vpn test_s2s_v6 ike gateway ike_gw_v6
set security ipsec vpn test_s2s_v6 ike ipsec-policy ipsec_policy
set security ipsec vpn test_s2s_v6 establish-tunnels immediately
set security ipsec vpn test_s2s_v4 bind-interface st0.0
set security ipsec vpn test_s2s_v4 ike gateway ike_gw_v4
set security ipsec vpn test_s2s_v4 ike ipsec-policy ipsec_policy
set routing-options rib inet6.0 static route 3000::1/128 next-hop st0.0
set routing-options rib inet6.0 static route 3000::2/128 next-hop st0.1
set routing-options static route 30.0.0.0/24 next-hop st0.1
set security forwarding-options family inet6 mode flow-based
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure dual-stack tunnels:

1. Configure the external interface.

```
[edit interfaces]
user@host# set ge-0/0/1 gigether-options redundant-parent reth1
user@host# set ge-8/0/1 gigether-options redundant-parent reth1
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 20.0.0.1/24
user@host# set reth1 unit 0 family inet6 address 2000::1/64
```

2. Configure the secure tunnel interfaces.

```
[edit interfaces]
user@host# set st0 unit 0 family inet6
user@host# set st0 unit 1 family inet
user@host# set st0 unit 1 family inet6
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike_proposal]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm md5
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 3600
[edit security ike policy ike_policy]
user@host# set mode aggressive
user@host# set proposals ike_proposal
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security ike gateway ike_gw_v6]
user@host# set ike-policy ike_policy
user@host# set address 2000::2
user@host# set external-interface reth1.0
user@host# set version v2-only
[edit security ike gateway ike_gw_v4]
user@host# set ike-policy ike_policy
user@host# set address 20.0.0.2
user@host# set external-interface reth1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
[edit security ipsec policy ipsec_policy]
user@host# set proposals ipsec_proposal
[edit security ipsec vpn test_s2s_v6 ]
user@host# set bind-interface st0.1
user@host# set ike gateway ike_gw_v6
user@host# set ike ipsec-policy ipsec_policy
user@host# set establish-tunnels immediately
[edit security ipsec vpn test_s2s_v4]
user@host# set bind-interface st0.0
user@host# set ike gateway ike_gw_v4
user@host# set ike ipsec-policy ipsec_policy
```

5. Configure static routes.

```
[edit routing-options rib inet6.0]
user@host# set static route 3000::1/128 next-hop st0.0
user@host# set static route 3000::2/128 next-hop st0.1
[edit routing-options]
user@host# set static route 30.0.0.0/24 next-hop st0.1
```

6. Enable IPv6 flow-based forwarding.

```
[edit security forwarding-options]
user@host# set family inet6 mode flow-based
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show routing-options`, and `show security forwarding-options` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show interfaces
    ge-0/0/1 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-8/0/1 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 20.0.0.1/24;
            }
            family inet6 {
                address 2000::1/64;
            }
        }
    }
    st0 {
        unit 0 {
            family inet;
            family inet6;
        }
        unit 1 {
            family inet6;
        }
    }
    [edit]
    user@host# show security ike
```

```
    proposal ike_proposal {
        authentication-method pre-shared-keys;
        authentication-algorithm md5;
        encryption-algorithm 3des-cbc;
        lifetime-seconds 3600;
    }
    policy ike_policy {
        mode aggressive;
        proposals ike_proposal;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway ike_gw_v6 {
        ike-policy ike_policy;
        address 2000::2;
        external-interface reth1.0;
        version v2-only;
    }
    gateway ike_gw_4 {
        ike-policy ike_policy;
        address 20.0.0.2;
        external-interface reth1.0;
    }
    [edit]
    user@host# show security ipsec
    proposal ipsec_proposal {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm 3des-cbc;
    }
    policy ipsec_policy {
        proposals ipsec_proposal;
    }
    vpn test_s2s_v6 {
        bind-interface st0.1;
        ike {
            gateway ike_gw_v6;
            ipsec-policy ipsec_policy;
        }
        establish-tunnels immediately;
    }
    vpn test_s2s_v4 {
        bind-interface st0.0;
        ike {
```

```
            gateway ike_gw_4;
            ipsec-policy ipsec_policy;
        }
    }
    [edit]
    user@host# show routing-options
    rib inet6.0 {
        static {
            route 3000::1/128 next-hop st0.0;
            route 3000::2/128 next-hop st0.1;
        }
    }
    static {
        route 30.0.0.0/24 next-hop st0.1;
    }
    [edit]
user@host# show security forwarding-options
    family {
        inet6 {
            mode flow-based;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Phase 1 Status | **437**
- Verifying IPsec Phase 2 Status | **437**
- Verifying Routes | **438**

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the `show security ike security-associations` command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
1081812113 UP  51d9e6df8a929624  7bc15bb40781a902  IKEv2          2000::2
1887118424 UP  d80b55b949b54f0a  b75ecc815529ae8f  Aggressive     20.0.0.2
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the peer devices.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command.

```
user@host> show security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131074 ESP:3des/sha1 8828bd36 3571/  unlim     -   root 500   20.0.0.2
  >131074 ESP:3des/sha1 c968afd8 3571/  unlim     -   root 500   20.0.0.2
  <131073 ESP:3des/sha1 8e9e695a 3551/  unlim     -   root 500   2000::2
  >131073 ESP:3des/sha1 b3a254d1 3551/  unlim     -   root 500   2000::2
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the peer devices.

Verifying Routes

Purpose

Verify active routes.

Action

From operational mode, enter the `show route` command.

```
user@host> show route
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.5.0.0/16        *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.10.0.0/16       *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.150.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.150.48.0/21     *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.155.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.157.64.0/19     *[Direct/0] 3d 01:43:23
                    > via fxp0.0
10.157.72.36/32    *[Local/0] 3d 01:43:23
                      Local via fxp0.0
10.204.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.206.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.209.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
20.0.0.0/24        *[Direct/0] 03:45:41
```

```
                    > via reth1.0
20.0.0.1/32         *[Local/0] 03:45:41
                      Local via reth1.0
30.0.0.0/24         *[Static/5] 00:07:49
                    > via st0.1
50.0.0.0/24         *[Direct/0] 03:45:42
                    > via reth0.0
50.0.0.1/32         *[Local/0] 03:45:42
                      Local via reth0.0
172.16.0.0/12       *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
192.168.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
192.168.102.0/23    *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
207.17.136.0/24     *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
207.17.136.192/32   *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0

inet6.0: 10 destinations, 14 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2000::/64           *[Direct/0] 03:45:41
                    > via reth1.0
2000::1/128         *[Local/0] 03:45:41
                      Local via reth1.0
3000::1/128         *[Static/5] 00:03:45
                    > via st0.0
3000::2/128         *[Static/5] 00:03:45
                    > via st0.1
5000::/64           *[Direct/0] 03:45:42
                    > via reth0.0
5000::1/128         *[Local/0] 03:45:42
                      Local via reth0.0
fe80::/64           *[Direct/0] 03:45:42
                    > via reth0.0
                    [Direct/0] 03:45:41
                    > via reth1.0
                    [Direct/0] 03:45:41
                    > via st0.0
                    [Direct/0] 03:45:13
                    > via st0.1
```

```
fe80::210:dbff:feff:1000/128
                   *[Local/0] 03:45:42
                      Local via reth0.0
fe80::210:dbff:feff:1001/128
                   *[Local/0] 03:45:41
                      Local via reth1.0
```

Meaning

The `show route` command lists active entries in the routing tables.

RELATED DOCUMENTATION

IPv6 IPsec VPNs | 153

IPsec VPN Overview | 79

IPsec VPN Tunnels with Chassis Clusters

IN THIS SECTION

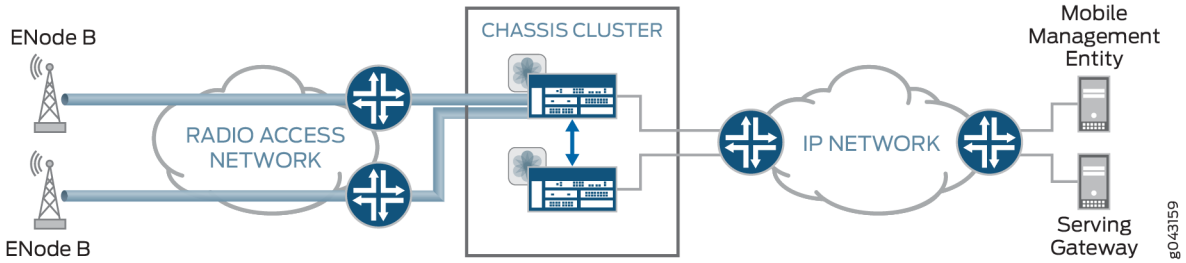
- Understanding Dual Active-Backup IPsec VPN Chassis Clusters | 440
- Example: Configuring Redundancy Groups for Loopback Interfaces | 442

Juniper Networks supports IPsec VPN tunnels in a chassis cluster setup. In an active/passive chassis cluster on the firewalls, all VPN tunnels terminate on the same node. In an active/active chassis cluster, VPN tunnels can terminate on either node.

Understanding Dual Active-Backup IPsec VPN Chassis Clusters

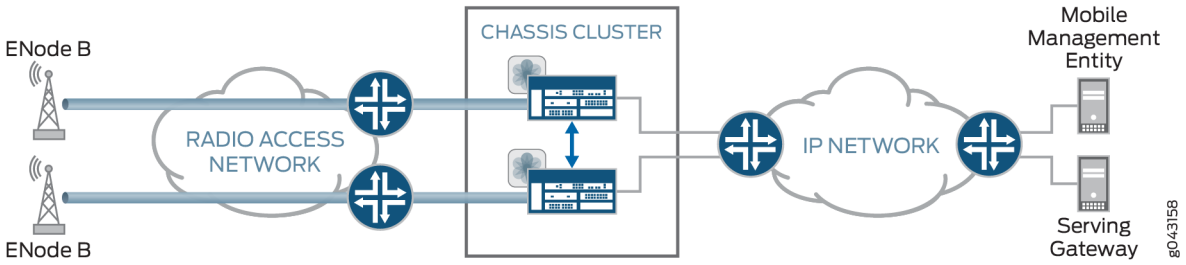
In an active/passive chassis cluster, all VPN tunnels terminate on the same node, as shown in Figure 32 on page 441.

Figure 32: Active/Passive Chassis Cluster with IPsec VPN Tunnels



In an active/active chassis cluster, VPN tunnels can terminate on either node. Both nodes in the chassis cluster can actively pass traffic through VPN tunnels on both nodes at the same time, as shown in Figure 33 on page 441. This deployment is known as *dual active-backup IPsec VPN chassis clusters*.

Figure 33: Dual Active-Backup IPsec VPN Chassis Clusters



The following features are supported with dual active-backup IPsec VPN chassis clusters:

- Route-based VPNs only. Policy-based VPNs are not supported.
- IKEv1 and IKEv2.
- Digital certificate or preshared key authentication.
- IKE and secure tunnel interfaces (st0) in virtual routers.
- Network Address Translation-Traversal (NAT-T).
- VPN monitoring.
- Dead peer detection.
- In-service software upgrade (ISSU).

- Insertion of Services Processing Cards (SPCs) on a chassis cluster device without disrupting the traffic on the existing VPN tunnels. See "VPN Support for Inserting Services Processing Cards" on page 86.
- Dynamic routing protocols.
- Secure tunnel interfaces (st0) configured in point-to-multipoint mode.
- AutoVPN with st0 interfaces in point-to-point mode with traffic selectors.
- IPv4-in-IPv4, IPv6-in-IPv4, IPv6-in-IPv6 and IPv4-in-IPv6 tunnel modes.
- Fragmented traffic.
- The loopback interface can be configured as the external interface for the VPN.

Dual active-backup IPsec VPN chassis clusters cannot be configured with Z-mode flows. Z-mode flows occur when traffic enters an interface on a chassis cluster node, passes through the fabric link, and exits through an interface on the other cluster node.

SEE ALSO

Chassis Cluster User Guide for Firewalls

Example: Configuring Redundancy Groups for Loopback Interfaces

IN THIS SECTION

- Requirements | **443**
- Overview | **443**
- Configuration | **445**
- Verification | **449**

This example shows how to configure a redundancy group (RG) for a loopback interface in order to prevent VPN failure. Redundancy groups are used to bundle interfaces into a group for failover purpose in a chassis cluster setup.

Requirements

This example uses the following hardware and software:

- A pair of firewalls that support chassis cluster
- An SSG140 device or equivalent
- Two switches
- Supported Junos OS Release for the firewalls

Before you begin:

Understand chassis cluster redundant Ethernet interfaces. See Chassis Cluster User Guide for Firewalls.

Overview

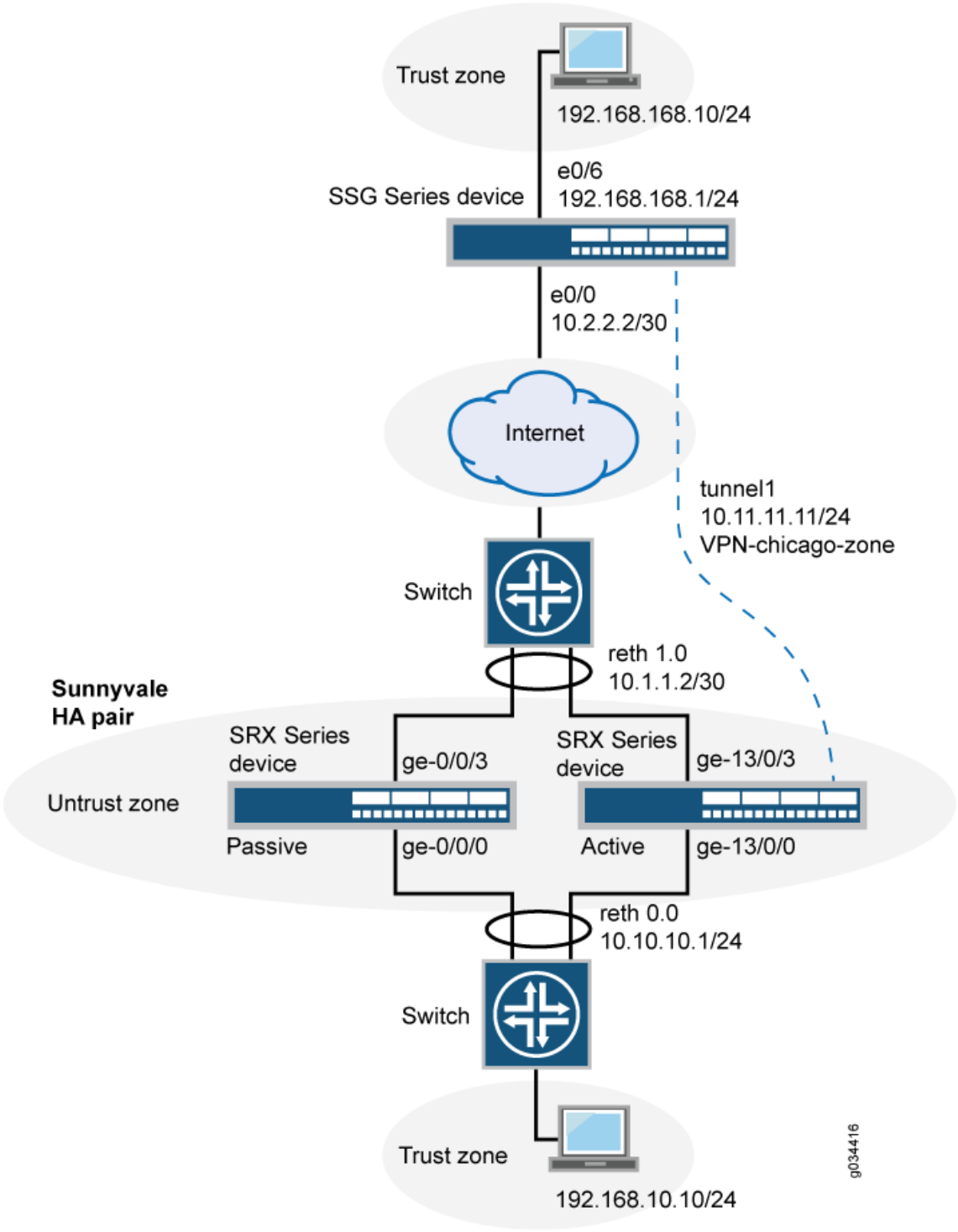
An Internet Key Exchange (IKE) gateway needs an external interface to communicate with a peer device. In a chassis cluster setup, the node on which the external interface is active selects a Services Processing Unit (SPU) to support the VPN tunnel. IKE and IPsec packets are processed on that SPU. Therefore, the active external interface decides the anchor SPU.

In a chassis cluster setup, the external interface is a redundant Ethernet interface. A redundant Ethernet interface can go down when its physical (child) interfaces are down. You can configure a loopback interface as an alternative physical interface to reach the peer gateway. Loopback interfaces can be configured on any redundancy group. This redundancy group configuration is only checked for VPN packets, because only VPN packets must find the anchor SPU through the active interface.

You must configure lo0.x in a custom virtual router, since lo0.0 is in the default virtual router and only one loopback interface is allowed in a virtual router.

Figure 34 on page 444 shows an example of a loopback chassis cluster VPN topology. In this topology, the firewall chassis cluster device is located in Sunnyvale, California. The chassis cluster device works as a single gateway in this setup. The SSG Series device (or a third-party device) is located in Chicago, Illinois. This device acts as a peer device to the firewall chassis cluster and it helps to build a VPN tunnel.

Figure 34: Loopback Interface for Chassis Cluster VPN



lo0.1 10.3.3.3/30 Untrust zone	A logical interface on loopback that may be active on either node in the cluster depending on the activeness of its RG.
st0.0 10.11.11.10/24 vpn-chicago-zone	A logical interface on the secure tunnel interface for IPSec VPN tunnel.

Configuration

IN THIS SECTION

- Procedure | 445

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces lo0 unit 1 family inet address 10.3.3.3/30
set routing-instances vr1 instance-type virtual-router
set routing-instances vr1 interface lo0.1
set routing-instances vr1 interface reth0.0
set routing-instances vr1 interface reth1.0
set routing-instances vr1 interface st0.0
set routing-instances vr1 routing-options static route 192.168.168.1/24 next-hop st0.0
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposal-set standard
set security ike policy ike-policy1 pre-shared-key ascii-text "$ABC123"
set security ike gateway t-ike-gate ike-policy ike-policy1
set security ike gateway t-ike-gate address 10.2.2.2
set security ike gateway t-ike-gate external-interface lo0.1
set security ipsec proposal p2-std-p1 authentication-algorithm hmac-sha1-96
set security ipsec proposal p2-std-p1 encryption-algorithm 3des-cbc
set security ipsec proposal p2-std-p1 lifetime-seconds 180
set security ipsec proposal p2-std-p2 authentication-algorithm hmac-sha1-96
set security ipsec proposal p2-std-p2 encryption-algorithm aes-128-cbc
set security ipsec proposal p2-std-p2 lifetime-seconds 180
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group2
set security ipsec policy vpn-policy1 proposals p2-std-p1
set security ipsec policy vpn-policy1 proposals p2-std-p2
set security ipsec vpn t-ike-vpn bind-interface st0.0
```

```
set security ipsec vpn t-ike-vpn ike gateway t-ike-gate
set security ipsec vpn t-ike-vpn ike proxy-identity local 10.10.10.1/24
set security ipsec vpn t-ike-vpn ike proxy-identity remote 192.168.168.1/24
set security ipsec vpn t-ike-vpn ike ipsec-policy vpn-policy1
```

Step-by-Step Procedure

To configure a redundancy group for a loopback interface:

1. Configure the loopback interface in one redundancy group.

```
[edit interfaces]
user@host# set lo0 redundant-pseudo-interface-options redundancy-group 1
```

2. Configure the IP address for the loopback interface.

```
[edit interfaces]
user@host# set lo0 unit 1 family inet address 10.3.3.3/30
```

3. Configure routing options.

```
[edit routing-instances]
user@host# set vr1 instance-type virtual-router
user@host# set vr1 interface lo0.1
user@host# set vr1 interface reth0.0
user@host# set vr1 interface reth1.0
user@host# set vr1 interface st0.0
user@host# set vr1 routing-options static route 192.168.168.1/24 next-hop st0.0
```

4. Configure the loopback interface as an external interface for the IKE gateway.

```
[edit security ike]
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposal-set standard
user@host# set policy ike-policy1 pre-shared-key ascii-text "$ABC123"
user@host# set gateway t-ike-gate ike-policy ike-policy1
user@host# set gateway t-ike-gate address 10.2.2.2
user@host# set gateway t-ike-gate external-interface lo0.1
```

5. Configure an IPsec proposal.

```
[edit security ipsec]
user@host# set proposal p2-std-p1 authentication-algorithm hmac-sha1-96
user@host# set proposal p2-std-p1 encryption-algorithm 3des-cbc
user@host# set proposal p2-std-p1 lifetime-seconds 180
user@host# set proposal p2-std-p2 authentication-algorithm hmac-sha1-96
user@host# set proposal p2-std-p2 encryption-algorithm aes-128-cbc
user@host# set proposal p2-std-p2 lifetime-seconds 180
user@host# set policy vpn-policy1 perfect-forward-secrecy keys group2
user@host# set policy vpn-policy1 proposals p2-std-p1
user@host# set policy vpn-policy1 proposals p2-std-p2
user@host# set vpn t-ike-vpn bind-interface st0.0
user@host# set vpn t-ike-vpn ike gateway t-ike-gate
user@host# set vpn t-ike-vpn ike proxy-identity local 10.10.10.1/24
user@host# set vpn t-ike-vpn ike proxy-identity remote 192.168.168.1/24
user@host# set vpn t-ike-vpn ike ipsec-policy vpn-policy1
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces lo0`, `show routing-instances`, `show security ike`, and `show security ipsec` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces lo0
        unit 1 {
            family inet {
                address 10.3.3.3/30;
            }
        }
        redundant-pseudo-interface-options {
            redundancy-group 1;
        }
```

```
[edit]
user@host# show routing-instances
    vr1 {
        instance-type virtual-router;
```

```
        interface lo0.1;
        interface reth0.0;
        interface reth1.0;
        interface st0.0;
        routing-options {
            static {
                route 192.168.168.1/24 next-hop st0.0;
            }
        }
    }
```

```
[edit]
user@host# show security ike
    policy ike-policy1 {
        mode main;
        proposal-set standard;
        pre-shared-key ascii-text "$ABC123";
    }
        gateway t-ike-gate {
            ike-policy ike-policy1;
            address 10.2.2.2;
            external-interface lo0.1;
        }
```

```
[edit]
user@host# show security ipsec
    proposal p2-std-p1 {
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm 3des-cbc;
        lifetime-seconds 180;
    }
        proposal p2-std-p2 {
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm aes-128-cbc;
            lifetime-seconds 180;
        }
        policy vpn-policy1 {
            perfect-forward-secrecy {
                keys group2;
            }
```

```
            proposals [ p2-std-p1 p2-std-p2 ];
        }
    policy vpn-policy2 {
        perfect-forward-secrecy {
            keys group2;
        }
        proposals [ p2-std-p1 p2-std-p2 ];
    }
        vpn t-ike-vpn {
            bind-interface st0.0;
            ike {
                gateway t-ike-gate;
                proxy-identity {
                    local 10.10.10.1/24;
                    remote 192.168.168.1/24;
                }
                ipsec-policy vpn-policy1;
            }
        }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying the Configuration | 449

Verifying the Configuration

Purpose

Verify that the configuration for redundancy groups for loopback interfaces is correct.

Action

From operational mode, enter the **show chassis cluster interfaces** command.

```
user@host> show chassis cluster interfaces
Control link status: Up
    Control interfaces:
     Index  Interface   Status
     0       em0              Up
     1       em1             Down
    Fabric link status: Up
    Fabric interfaces:
    Name    Child-interface    Status
     fab0    ge-0/0/7           Up   / Up
     fab0
     fab1    ge-13/0/7          Up   / Up
     fab1
    Redundant-ethernet Information:
    Name      Status    Redundancy-group
     reth0        Up          1
     reth1        Up          1
     reth2        Up          1
     reth3        Down      Not configured
     reth4        Down      Not configured
    Redundant-pseudo-interface Information:
    Name    Status      Redundancy-group
     lo0          Up          1
```

Meaning

The **show chassis cluster interfaces** command displays the chassis cluster interfaces information. If the status of the Redundant-pseudo-interface Information field shows the lo0 interface as Up and the status of the Redundant-ethernet Information field shows reth0, reth1, and reth2 fields as Up then your configuration is correct.

SEE ALSO

Understanding the Loopback Interface for a High Availability VPN | **1239**

RELATED DOCUMENTATION

- Chassis Cluster User Guide for Firewalls

Traffic Selectors in Route-Based VPNs

SUMMARY

Read this topic to learn about the traffic selectors in route-based IPsec VPNs and how to configure traffic selectors in your firewalls.

IN THIS SECTION

- Understand Traffic Selectors in Route-Based VPNs | **451**
- Example: Configure Traffic Selectors in a Route-Based VPN | **458**
- Platform-Specific ARI for Traffic Selectors Behavior | **478**

A traffic selector is an agreement between IKE peers to permit traffic through a VPN tunnel if the traffic matches a specified pair of local and remote addresses. Only the traffic that conforms to a traffic selector is permitted through the associated security association (SA).

Understand Traffic Selectors in Route-Based VPNs

IN THIS SECTION

- Traffic Selector Configuration | **452**
- Understand Auto Route Insertion | **453**
- Understand Traffic Selectors and Overlapping IP Addresses | **454**

A traffic selector is an agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified pair of local and remote addresses. With this feature, you can define a traffic selector within a specific route-based VPN, which can result in multiple Phase 2 IPsec security associations (SAs). Only traffic that conforms to a traffic selector is permitted through the associated SA.

Traffic Selector Configuration

To configure a traffic selector, use the `traffic-selector` configuration statement at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. The traffic selector is defined with the mandatory `local-ip` *`ip-address/netmask`* and `remote-ip` *`ip-address/netmask`* statements. The CLI operational command `show security ipsec security-association detail` displays traffic selector information for SAs. The `show security ipsec security-association traffic-selector` *`traffic-selector-name`* CLI command displays information for a specified traffic selector.

For a given traffic selector, a single address and netmask is specified for the local and remote addresses. Traffic selectors can be configured with IPv4 or IPv6 addresses. Address books cannot be used to specify local or remote addresses.

Multiple traffic selectors can be configured for the same VPN. A maximum of 200 traffic selectors can be configured for each VPN. Traffic selectors can be used with IPv4-in-IPv4, IPv4-in-IPv6, IPv6-in-IPv6, or IPv6-in-IPv4 tunnel modes.

Below features are not supported with traffic selectors:

- VPN monitoring
- Different address families configured for the local and remote IP addresses in a traffic selector
- A remote address of 0.0.0.0/0 (IPv4) or 0::0 (IPv6) for site-to-site VPNs
- Point-to-multipoint interfaces
- Any traffic over dynamic routing protocols

When there are multiple traffic selectors configured for a route-based VPN, clear traffic may enter a VPN tunnel without matching a traffic selector if the IKE gateway external interface is moved to another virtual router (VR). The software does not handle the multiple asynchronous interface events generated when an IKE gateway external interface is moved to another VR. As a workaround, first deactivate the IPsec VPN tunnel and commit the configuration without that tunnel before moving the IKE gateway external interface to another VR.

You can configure multiple sets of local IP prefix, remote IP prefix, source port range, destination port range, and protocol for traffic selection. This means, multiple sets of IP address ranges, port ranges, and protocols can be part of same traffic selector as defined in RFC 7296. When you configure multiple traffic selectors, each traffic selector leads to a separate negotiation that results in the multiple IPsec tunnels. But, if you configure multiple terms under one traffic selector, this configuration results in single IPsec SA negotiation with multiple IP prefixes, ports, and protocols. See Traffic Selector.

For traffic selector support in kmd and ike process, see Table 73 on page 453.

Table 73: Traffic Selector Support for kmd and iked Process

Component	IKEv1	IKEv2
kmd process	Traffic selector configuration supports remote IP and local IP.	Traffic selector configuration supports remote IP and local IP.
iked process	Traffic selector configuration supports only remote IP and local IP. The configuration doesn't support protocol, port or terms within the traffic selector.	Traffic selector configuration supports remote IP, local IP, port, and protocol including the terms configuration.

Understand Auto Route Insertion

Auto route insertion (ARI) automatically inserts a static route for the remote network and hosts protected by a remote tunnel endpoint. A route is created based on the remote IP address configured in the traffic-selector. In the case of traffic selectors, the configured remote address is inserted as a route in the routing instance associated with the st0 interface that is bound to the VPN.

Routing protocols and traffic selector configuration are mutually exclusive ways of steering traffic to a tunnel. ARI routes might conflict with routes that are populated through routing protocols. Therefore, you should not configure routing protocols on an st0 interface that is bound to a VPN on which traffic selectors are configured.

ARI is also known as reverse route insertion (RRI). ARI routes are inserted in the routing table as follows:

- If the `establish-tunnels immediately` option is configured at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level, ARI routes are added after Phase 1 and Phase 2 negotiations are complete. Because a route is not added until SAs are established, a failed negotiation does not result in traffic being routed to a st0 interface that is down. An alternate or backup tunnel is used instead.
- If the `establish-tunnels immediately` option is not configured at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level, ARI routes are added at configuration commit.
- An ARI route is not added if the configured or negotiated remote address in a traffic selector is 0.0.0.0/0 or 0::0.

The preference for the static ARI route is 5. This value is necessary to avoid conflict with similar routes that might be added by a routing protocol process.

The static ARI route cannot be leaked to other routing instances using the `rib-groups` configuration. Use the `import-policy` configuration to leak static ARI routes.

Understand Traffic Selectors and Overlapping IP Addresses

This section discusses overlapping IP addresses in traffic selector configurations.

Overlapping IP Addresses in Different VPNs Bound to the Same st0 Interface

This scenario is not supported with traffic selectors. Traffic selectors cannot be configured on different VPNs that are bound to the same point-to-multipoint st0 interface, as shown in the following example:

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
}
vpn vpn-2 {
    bind-interface st0.1;
}
```

Overlapping IP Addresses in the Same VPN Bound to the Same st0 Interface

When overlapping IP addresses are configured for multiple traffic selectors in the same VPN, the first configured traffic selector that matches the packet determines the tunnel used for packet encryption.

In the following example, four traffic selectors (ts-1, ts-2, ts-3, and ts-4) are configured for the VPN (vpn-1), which is bound to the point-to-point st0.1 interface:

```
[edit]
user@host# show security ipsec vpn vpn-1
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.5.0/24;
        remote-ip 10.1.5.0/24;
    }
    traffic-selector ts-2 {
        local-ip 192.168.0.0/16;
        remote-ip 10.1.0.0/16;
    }
    traffic-selector ts-3 {
        local-ip 172.16.0.0/16;
        remote-ip 10.2.0.0/16;
```

```
    }
    traffic-selector ts-4 {
        local-ip 172.16.5.0/24;
        remote-ip 10.2.5.0/24;
    }
}
```

A packet with a source address 192.168.5.5 and a destination address 10.1.5.10 matches traffic selectors ts-1 and ts-2. However, traffic selector ts-1 is the first configured match and the tunnel associated with ts-1 is used for packet encryption.

A packet with a source address 172.16.5.5 and a destination address 10.2.5.10 matches the traffic selectors ts-3 and ts-4. However, traffic selector ts-3 is the first configured match and the tunnel associated with traffic selector ts-3 is used for packet encryption.

Overlapping IP Addresses in Different VPNs Bound to Different st0 Interfaces

When overlapping IP addresses are configured for multiple traffic selectors in different VPNs that are bound to different point-to-point st0 interfaces, an st0 interface is first selected by the longest prefix match for a given packet. Within the VPN that is bound to the selected st0 interface, the traffic selector is then selected based on the first configured match for the packet.

In the following example, a traffic selector is configured in each of two VPNs. The traffic selectors are configured with the same local subnetwork but different remote subnetworks.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 192.168.1.0/24;
        remote-ip 10.2.2.0/24;
    }
}
```

Different remote subnetworks are configured in each traffic selector, therefore two different routes are added to the routing table. Route lookup uses the st0 interface bound to the appropriate VPN.

In the following example, a traffic selector is configured in each of two VPNs. The traffic selectors are configured with different remote subnetworks. The same local subnetwork is configured for each traffic selector, but different netmask values are specified.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.0.0/8;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 192.168.0.0/16;
        remote-ip 10.2.2.0/24;
    }
}
```

A different remote subnetwork is configured in each traffic selector, therefore two different routes are added to the routing table. Route lookup uses the st0 interface bound to the appropriate VPN.

In the following example, traffic selectors are configured in each of two VPNs. The traffic selectors are configured with different local and remote subnetworks.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
```

```
        local-ip 172.16.1.0/24;
        remote-ip 10.2.2.0/24;
    }
}
```

In this case, the traffic selectors do not overlap. The remote subnetworks configured in the traffic selectors are different, therefore two different routes are added to the routing table. Route lookup uses the st0 interface bound to the appropriate VPN.

In the following example, a traffic selector is configured in each of two VPNs. The traffic selectors are configured with the same local subnetwork. The same remote subnetwork is configured for each traffic selector, but different netmask values are specified.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.0.0/16;
    }
}
```

Note that the `remote-ip` configured for ts-1 is 10.1.1.0/24 while the `remote-ip` configured for ts-2 is 10.1.0.0/16. For a packet destined to 10.1.1.1, route lookup selects the st0.1 interface as it has the longer prefix match. The packet is encrypted based on the tunnel corresponding to the st0.1 interface.

In some cases, valid packets can be dropped due to traffic selector traffic enforcement. In the following example, traffic selectors are configured in each of two VPNs. The traffic selectors are configured with different local subnetworks. The same remote subnetwork is configured for each traffic selector, but different netmask values are specified.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
```

```
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 172.16.1.0/16;
        remote-ip 10.1.0.0/16;
    }
}
```

Two routes to 10.1.1.0 (10.1.1.0/24 via interface st0.1 and 10.1.0.0/16 via interface st0.2) are added to the routing table. A packet sent from source 172.16.1.1 to destination 10.1.1.1 matches the routing table entry for 10.1.1.0/24 via interface st0.1. However, the packet does not match the traffic specified by traffic selector ts-1 and is dropped.

If multiple traffic selectors are configured with the same remote subnetwork and netmask, equal cost routes are added to the routing table. This case is not supported with traffic selectors as the route chosen cannot be predicted.

SEE ALSO

Understanding VPN Tunnel Modes | 422

Example: Configure Traffic Selectors in a Route-Based VPN

IN THIS SECTION

- Requirements | 459
- Overview | 459
- Configuration | 460
- Verification | 475

This example shows how to configure traffic selectors for a route-based VPN.

Requirements

Before you begin,

- Read "Understanding Traffic Selectors in Route-Based VPNs" on page 451.
- Install the IKE package.

```
request system software add optional://junos-ike.tgz
```

To know about the platform support for `junos-ike` package, see Support for junos-ike Package.

Overview

IN THIS SECTION

- Topology | 460

This example configures traffic selectors to allow traffic to flow between subnetworks on SRX_A and subnetworks on SRX_B.

Table 74 on page 459 shows the traffic selectors for this example. Traffic selectors are configured under Phase 2 options.

Table 74: Traffic Selector Configurations

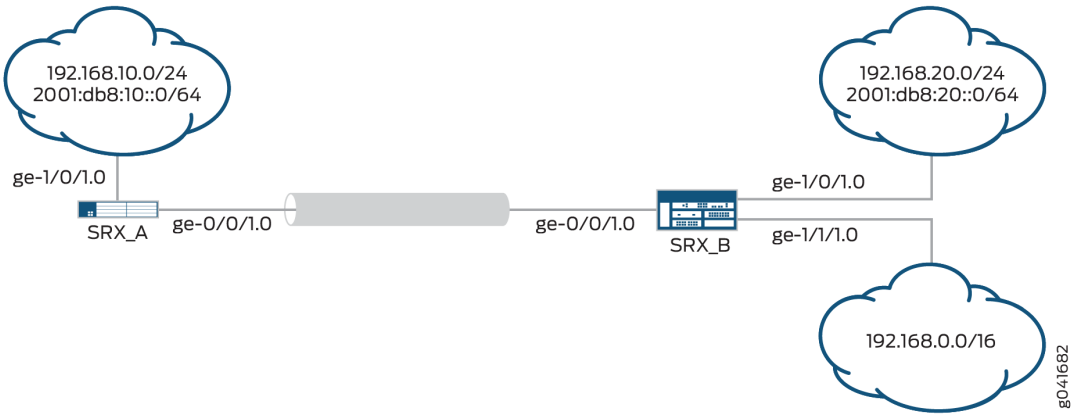
SRX_A			SRX_B		
Traffic Selector Name	Local IP	Remote IP	Traffic Selector Name	Local IP	Remote IP
TS1-ipv6	2001:db8:10::0/64	2001:db8:20::0/64	TS1-ipv6	2001:db8:20::0/64	2001:db8:10::0/64
TS2-ipv4	192.168.10.0/24	192.168.0.0/16	TS2-ipv4	192.168.0.0/16	192.168.10.0/24

Flow-based processing of IPv6 traffic must be enabled with the `mode flow-based` configuration option at the [`edit security forwarding-options family inet6`] hierarchy level.

Topology

In Figure 35 on page 460, an IPv6 VPN tunnel carries both IPv4 and IPv6 traffic between the SRX_A and SRX_B devices. That is, the tunnel operates in both IPv4-in-IPv6 and IPv6-in-IPv6 tunnel modes.

Figure 35: Traffic Selector Configuration Example



Configuration

IN THIS SECTION

- Configuring SRX_A | **461**
- Configuring SRX_B | **468**

Configuring SRX_A

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces st0 unit 1 family inet
set interfaces st0 unit 1 family inet6
set interfaces ge-1/0/1 unit 0 family inet address 192.168.10.1/24
set interfaces ge-1/0/1 unit 0 family inet6 address 2001:db8:10::0/64
set security ike proposal PSK-DH14-AES256-SHA256 authentication- method pre-shared-keys
set security ike proposal PSK-DH14-AES256-SHA256 dh-group group14
set security ike proposal PSK-DH14-AES256-SHA256 authentication- algorithm sha-256
set security ike proposal PSK-DH14-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ike policy site-2-site mode main
set security ike policy site-2-site proposals PSK-DH14-AES256-SHA256
set security ike policy site-2-site pre-shared-key ascii-text "$ABC123"
set security ike gateway SRX_A-to-SRX_B ike-policy site-2-site
set security ike gateway SRX_A-to-SRX_B address 192.168.20.2
set security ike gateway SRX_A-to-SRX_B external-interface ge-0/0/1.0
set security ike gateway SRX_A-to-SRX_B local-address 192.168.10.1
set security ipsec proposal ESP-AES256-SHA256 protocol esp
set security ipsec proposal ESP-AES256-SHA256 authentication- algorithm hmac-sha-256-128
set security ipsec proposal ESP-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ipsec policy site-2-site perfect-forward-secrecy keys group14
set security ipsec policy site-2-site proposals ESP-AES256-SHA256
set security ipsec vpn SRX_A-to-SRX_B bind-interface st0.1
set security ipsec vpn SRX_A-to-SRX_B ike ipsec-policy site-2-site
set security ipsec vpn SRX_A-to-SRX_B ike gateway SRX_A-to-SRX_B
set security ipsec vpn SRX_A-to-SRX_B traffic-selector TS1-ipv6 term term1 local-ip
2001:db8:10::0/64 remote-ip 2001:db8:20::0/64
set security ipsec vpn SRX_A-to-SRX_B traffic-selector TS2-ipv4 term term2 local-ip
192.168.10.0/24 remote-ip 192.168.0.0/16
set security forwarding-options family inet6 mode flow-based
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-1/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/1.0
```

```
set security zones security-zone VPN interfaces st0.1
set security policies from-zone VPN to-zone trust policy 1 match source-address any
set security policies from-zone VPN to-zone trust policy 1 match destination-address any
set security policies from-zone VPN to-zone trust policy 1 match application any
set security policies from-zone VPN to-zone trust policy 1 then permit
set security policies from-zone trust to-zone VPN policy 1 match source-address any
set security policies from-zone trust to-zone VPN policy 1 match destination-address any
set security policies from-zone trust to-zone VPN policy 1 match application any
set security policies from-zone trust to-zone VPN policy 1 then permit
set security policies default-policy deny -all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure traffic selectors:

1. Configure the external interface.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::1/64
```

2. Configure the secure tunnel interface.

```
[edit interfaces]
user@host# set st0 unit 1 family inet
user@host# set st0 unit 1 family inet6
```

3. Configure the internal interface.

```
[edit interfaces]
user@host# set ge-1/0/1 unit 0 family inet address 192.168.10.1/24
user@host# set ge-1/0/1 unit 0 family inet6 address 2001:db8:10::0/64
```

4. Configure Phase 1 options.

```
[edit security ike proposal PSK-DH14-AES256-SHA256]
user@host# set authentication-method pre-shared-keys
```

```
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy site-2-site]
user@host# set mode main
user@host# set proposals PSK-DH14-AES256-SHA256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security ike gateway SRX_A-to-SRX_B]
user@host# set ike-policy site-2-site
user@host# set address 192.168.20.2
user@host# set external-interface ge-0/0/1.0
user@host# set local-address 192.168.10.1
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal ESP-AES256-SHA256]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy site-2-site]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ESP-AES256-SHA256
[edit security ipsec vpn SRX_A-to-SRX_B]
user@host# set bind-interface st0.1
user@host# set ike gateway SRX_A-to-SRX_B
user@host# set ike ipsec-policy site-2-site
user@host# set traffic-selector TS1-ipv6 term term1 local-ip 2001:db8:10::0/64 remote-ip
2001:db8:20::0/64
user@host# set traffic-selector TS2-ipv4 term term2 local-ip 192.168.10.0/24 remote-ip
192.168.0.0/16
```

6. Enable IPv6 flow-based forwarding.

```
[edit security forwarding-options]
user@host# set family inet6 mode flow-based
```

7. Configure security zones and the security policy.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic  system-services all
```

```
user@host# set host-inbound-traffic  protocols all
user@host# set interfaces ge-1/0/1.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic  system-services ike
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone VPN]
user@host# set interfaces st0.1
[edit security policies from-zone VPN to-zone trust ]
user@host# set policy 1 match source-address any
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies from-zone trust to-zone VPN ]
user@host# set policy 1 match source-address any
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies]
user@host# set default-policy deny-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security forwarding-options`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:2000::1/64;
            }
        }
    }
    ge-1/0/1 {
        unit 0 {
            family inet {
                address 192.168.10.1/24;
            }
```

```
            family inet6 {
                address 10::1/64;
            }
        }
    }
    st0 {
        unit 1 {
            family inet;
            family inet6;
        }
    }
    [edit]
    user@host# show security ike
    proposal PSK-DH14-AES256-SHA256 {
        authentication-method pre-shared-keys;
        dh-group group14;
        authentication-algorithm sha-256;
        encryption-algorithm aes-256-cbc;
    }
    policy site-2-site {
        mode main;
        proposals PSK-DH14-AES256-SHA256;
            pre-shared-key ascii-text
        "$ABC123"; ## SECRET-DATA
    }
    gateway SRX_A-to-SRX_B {
        ike-policy site-2-site;
        address 192.168.20.2;
        external-interface ge-0/0/1.0;
        local-address 192.168.10.1;
    }
    [edit]
    user@host# show security ipsec
    proposal ESP-AES256-SHA256 {
        protocol esp;
        authentication-algorithm hmac-sha-256-128;
        encryption-algorithm aes-256-cbc;
    }
    policy site-2-site {
        perfect-forward-secrecy keys group14;
        proposals ESP-AES256-SHA256;
    }
    vpn SRX_A-to-SRX_B {
```

```
        bind-interface st0.1;
        ike {
            ipsec-policy site-2-site;
            gateway SRX_A-to-SRX_B;
        }
        traffic-selector TS1-ipv6 {
            local-ip 2001:db8:10::0/64;
            remote-ip 2001:db8:20::0/64;
        }
        traffic-selector TS2-ipv4 {
            local-ip 192.168.10.0/24;
            remote-ip 192.168.0.0/16;
        }
    }
    [edit]
    user@host# show security forwarding-options
    family {
        inet6 {
            mode flow-based;
        }
    }
    [edit]
    user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-1/0/1.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ike;
            }
        }
        interfaces {
```

```
            ge-0/0/1.0;
        }
    }
    security-zone VPN {
        interfaces {
            st0.1;
        }
    }
    [edit]
user@host# show security policies
    from-zone VPN to-zone trust {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone trust to-zone VPN {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring SRX_B

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::2/64
set interfaces st0 unit 1 family inet
set interfaces st0 unit 1 family inet6
set interfaces ge-1/0/1 unit 0 family inet address 192.168.20.1/24
set interfaces ge-1/0/1 unit 0 family inet6 address 2001:db8:20::0/64
set interfaces ge-1/1/1 unit 0 family inet address 192.168.0.1/24
set security ike proposal PSK-DH14-AES256-SHA256 authentication-method pre-shared-keys
set security ike proposal PSK-DH14-AES256-SHA256 dh-group group14
set security ike proposal PSK-DH14-AES256-SHA256 authentication-algorithm sha-256
set security ike proposal PSK-DH14-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ike policy site-2-site mode main
set security ike policy site-2-site proposals PSK-DH14-AES256-SHA256
set security ike policy site-2-site pre-shared-key ascii-text "$ABC123"
set security ike gateway SRX_B-to-SRX_A ike-policy site-2-site
set security ike gateway SRX_B-to-SRX_A address 192.168.10.1
set security ike gateway SRX_B-to-SRX_A external-interface ge-0/0/1.0
set security ike gateway SRX_B-to-SRX_A local-address 192.168.20.2
set security ipsec proposal ESP-AES256-SHA256 protocol esp
set security ipsec proposal ESP-AES256-SHA256 authentication-algorithm hmac-sha-256-128
set security ipsec proposal ESP-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ipsec policy site-2-site perfect-forward-secrecy keys group14
set security ipsec policy site-2-site proposals ESP-AES256-SHA256
set security ipsec vpn SRX_B-to-SRX-A bind-interface st0.1
set security ipsec vpn SRX_B-to-SRX-A ike ipsec-policy site-2-site
set security ipsec vpn SRX_B-to-SRX-A ike gateway SRX_B-to-SRX_A
set security ipsec vpn SRX_B-to-SRX-A traffic-selector TS1-ipv6 local-ip 2001:db8:20::0/64
remote-ip 2001:db8:10::0/64
set security ipsec vpn SRX_B-to-SRX-A traffic-selector TS2-ipv4 local-ip 192.168.0.0/16 remote-
ip 192.168.10.0/24
set security forwarding-options family inet6 mode flow-based
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-1/0/1.0
set security zones security-zone trust interfaces ge-1/1/1.0
```

```
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone VPN interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set security policies from-zone VPN to-zone trust policy 1 match source-address any
set security policies from-zone VPN to-zone trust policy 1 match destination-address any
set security policies from-zone VPN to-zone trust policy 1 match application any
set security policies from-zone VPN to-zone trust policy 1 then permit
set security policies from-zone trust to-zone VPN policy 1 match source-address any
set security policies from-zone trust to-zone VPN policy 1 match destination-address any
set security policies from-zone trust to-zone VPN policy 1 match application any
set security policies from-zone trust to-zone VPN policy 1 then permit
set security policies default-policy deny -all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure traffic selectors:

1. Configure the external interface.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::2/64
```

2. Configure the secure tunnel interface.

```
[edit interfaces]
user@host# set st0 unit 1 family inet
user@host# set st0 unit 1 family inet6
```

3. Configure the internal interfaces.

```
[edit interfaces]
user@host# set ge-1/0/1 unit 0 family inet address 192.168.20.1/24
user@host# set ge-1/0/1 unit 0 family inet6 address 2001:db8:20::0/64
user@host# set ge-1/1/1 unit 0 family inet address 192.168.0.1/24
```

4. Configure Phase 1 options.

```
[edit security ike proposal PSK-DH14-AES256-SHA256]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy site-2-site]
user@host# set mode main
user@host# set proposals PSK-DH14-AES256-SHA256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security ike gateway SRX_B-to-SRX_A]
user@host# set ike-policy site-2-site
user@host# set address 192.168.10.1
user@host# set external-interface ge-0/0/1.0
user@host# set local-address 192.168.20.2
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal ESP-AES256-SHA256]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy site-2-site]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ESP-AES256-SHA256
[edit security ipsec vpn SRX_B-to-SRX-A]
user@host# set bind-interface st0.1
user@host# set ike gateway SRX_B-to-SRX_A
user@host# set ike ipsec-policy site-2-site
user@host# set traffic-selector TS1-ipv6 local-ip 2001:db8:20::0/64 remote-ip
2001:db8:10::0/64
user@host# set traffic-selector TS2-ipv4 local-ip 192.168.0.0/16 remote-ip 192.168.10.0/24
```

6. Enable IPv6 flow-based forwarding.

```
[edit security forwarding-options]
user@host# set family inet6 mode flow-based
```

7. Configure security zones and the security policy.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic  system-services all
user@host# set host-inbound-traffic  protocols all
user@host# set interfaces ge-1/0/1.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic  system-services ike
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone VPN]
user@host# set interfaces st0.1
[edit security policies from-zone VPN to-zone trust ]
user@host# set policy 1 match source-address any
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies from-zone trust to-zone VPN ]
user@host# set policy 1 match source-address any
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies]
user@host# set default-policy deny-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security forwarding-options`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:2000::2/64;
            }
        }
    }
```

```
    ge-1/0/1 {
        unit 0 {
            family inet {
                address 192.168.20.1/24;
            }
            family inet6 {
                address 2001:db8:20::0/64;
            }
        }
    }
    ge-1/1/1 {
        unit 0 {
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    st0 {
        unit 1 {
            family inet;
            family inet6;
        }
    }
    [edit]
    user@host# show security ike
    proposal PSK-DH14-AES256-SHA256 {
        authentication-method pre-shared-keys;
        dh-group group14;
        authentication-algorithm sha-256;
        encryption-algorithm aes-256-cbc;
    }
    policy site-2-site {
        mode main;
        proposals PSK-DH14-AES256-SHA256;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway SRX_B-to-SRX_A {
        ike-policy site-2-site;
        address 192.168.10.1;
        external-interface ge-0/0/1.0;
        local-address 192.168.20.2;
    }
    [edit]
```

```
user@host# show security ipsec
proposal ESP-AES256-SHA256 {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-256-cbc;
}
policy site-2-site {
    perfect-forward-secrecy keys group14;
    proposals ESP-AES256-SHA256;
}
vpn SRX_B-to-SRX-A {
    bind-interface st0.1;
    ike {
        ipsec-policy site-2-site;
        gateway SRX_B-to-SRX_A;
    }
    traffic-selector TS1-ipv6 {
        local-ip 2001:db8:20::0/64;
        remote-ip 2001:db8:10::0/64;
    }
    traffic-selector TS2-ipv4 {
        local-ip 192.168.0.0/16;
        remote-ip 192.168.10.0/24;
    }
}
[edit]
user@host# show security forwarding-options
family {
    inet6 {
        mode flow-based;
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
```

```
        interfaces {
            ge-1/0/1.0;
            ge-1/1/1.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ike;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
    security-zone VPN {
        interfaces {
            st0.1;
        }
    }
    [edit]
user@host# show security policies
    from-zone VPN to-zone trust {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone trust to-zone VPN {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
```

```
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IPsec Phase 2 Status | **475**
- Verifying Traffic Selectors | **477**
- Verifying Routes | **478**

Confirm that the configuration is working properly.

The sample outputs shown are on SRX-A.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command.

```
user@host> show security ipsec security-associations
  Total active tunnels: 3
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <268173313 ESP:3des/ sha-256 3d75aeff 2984/ unlim -  root 500   2001:db8:2000::2
  >268173313 ESP:3des/ sha-256 a468fece 2984/ unlim -  root 500   2001:db8:2000::2
  <268173316 ESP:3des/ sha-256 417f3cea 3594/ unlim -  root 500   2001:db8:2000::2
  >268173316 ESP:3des/ sha-256 a4344027 3594/ unlim -  root 500   2001:db8:2000::2
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@host> show security ipsec security-associations detail
  ID: 268173313 Virtual-system: root, VPN Name: SRX_A-to-SRX_B
  Local Gateway: 192.168.10.1, Remote Gateway: 2192.168.20.2
  Traffic Selector Name: TS1-ipv6
  Local Identity: ipv6(2001:db8:10::-2001:db8:10::ffff:ffff:ffff:ffff)
  Remote Identity: ipv6(2001:db8:20::-2001:db8:20::ffff:ffff:ffff:ffff)
  Version: IKEv1
    DF-bit: clear
    Bind-interface: st0.1

  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: c608b29
  Tunnel Down Reason: SA not initiated
    Direction: inbound, SPI: 3d75aeff, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2976 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2354 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: a468fece, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2976 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2354 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

  ID: 268173316 Virtual-system: root, VPN Name: SRX_A-to-SRX_B
  Local Gateway: 192.168.10.1, Remote Gateway: 192.168.20.2
  Traffic Selector Name: TS2-ipv4
  Local Identity: ipv4(192.168.10.0-192.168.10.255)
  Remote Identity: ipv4(192.168.20.0-192.168.20.255)
  Version: IKEv1
    DF-bit: clear
    Bind-interface: st0.1

  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: c608b29
```

```
  Tunnel Down Reason: SA not initiated
    Direction: inbound, SPI: 417f3cea, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3586 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2948 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: a4344027, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3586 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2948 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the peer devices.

Verifying Traffic Selectors

Purpose

Verify negotiated traffic selectors on the secure tunnel interface.

Action

From operational mode, enter the `show security ipsec traffic-selector st0.1` command.

```
user@host> show security ipsec traffic-selector st0.1
Source IP                                        Destination
IP                                                        Interface   Tunnel-id         IKE-ID
2001:db8:10::-2001:db8:10::ffff:ffff:ffff:ffff
2001:db8:20::-2001:db8:20::ffff:ffff:ffff:ffff      st0.1       268173313      2001:db8:2000::1
192.168.10.0-192.168.10.255
```

```
192.168.0.0-192.168.255.255                          st0.1       268173316
2001:db8:2000::1
192.168.10.0-192.168.10.255
192.168.20.0-192.168.20.255                          st0.1       268173317
2001:db8:2000::1
```

Verifying Routes

Purpose

Verify active routes

Action

From operational mode, enter the `show route` command.

```
user@host> show route
inet.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.0/16        *[ARI-TS/5] 00:00:32
                    > via st0.1
2001:db8:20::0/64      *[ARI-TS/5] 00:00:34
                    > via st0.1
```

Meaning

The `show route` command lists active entries in the routing tables. Routes to the remote IP address configured in each traffic selector should be present with the correct st0 interface.

SEE ALSO

Understanding VPN Tunnel Modes | 422

Platform-Specific ARI for Traffic Selectors Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platforms.

Table 75: Platform-Specific Behavior

Platform	Difference
MX Series	• On MX Series routers that support auto route insertion (ARI) for traffic selectors with the iked process, the device supports installation of default routes when the `st0` interface is in a non-default routing instance. The feature supports migration from MS-MPC to SPC3 as MX-SPC3 injects these routes as ARI for traffic selector routes. The configuration facilitates route installation in specified routing instances.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
21.1R1	Starting with Junos OS Release 21.1R1, you can configure multiple sets of local IP prefixes, remote IP prefixes, source port ranges, destination port ranges, and protocols for traffic selection.
15.1X49-D140	Starting with Junos OS Release 15.1X49-D140, when you configure the traffic-selector with a remote address of 0::0 (IPv6), the following **"error: configuration check-out failed"** message is displayed when performing the commit and the configuration checkout fails.
15.1X49-D100	Starting with Junos OS Release 15.1X49-D100, traffic selectors can be configured with IKEv2 site-to-site VPNs.
12.1X46-D10	Starting with Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, traffic selectors can be configured with IKEv1 site-to-site VPNs.

RELATED DOCUMENTATION

Route-Based IPsec VPNs | 320

Migrate to Route-Based IPsec VPNs

SUMMARY

Read this topic when migrating from policy-based VPNs to route-based VPNs using a shared st0 interface.

IN THIS SECTION

- Comparison of Policy-Based VPNs and Route-Based VPNs | **480**
- Shared Point-to-Point st0 Interface | **483**
- Migrate Policy-Based VPNs to Route-Based VPNs | **485**

This topic describes how to migrate existing policy-based IPsec VPNs to route-based VPNs using a shared point-to-point st0 interface. It compares policy-based and route-based VPNs, explains the role of the shared st0 interface in the migration, and provides essential configuration guidance to support the transition.

Comparison of Policy-Based VPNs and Route-Based VPNs

SUMMARY

This table compares policy-based and route-based VPNs to help you understand why you should migrate to route-based VPNs.

Table 76 on page 481 summarizes the differences between policy-based VPNs and route-based VPNs.

Table 76: Comparison Between Policy-Based VPNs and Route-Based VPNs

Policy-Based VPNs	Route-Based VPNs
In policy-based VPNs, a tunnel is treated as an object that, together with source, destination, application, and action, constitutes a tunnel policy that permits VPN traffic.	In route-based VPNs, a policy does not specifically reference a VPN tunnel.
A tunnel policy specifically references a VPN tunnel by name.	A route determines which traffic is sent through the tunnel based on a destination IP address.
The number of policy-based VPN tunnels that you can create is limited by the number of tunnels that the device supports.	The number of route-based VPN tunnels that you create is limited by the number of st0 interfaces (for point-to-point VPNs) or the number of tunnels that the device supports, whichever is lower.
With a policy-based VPN, although you can create numerous tunnel policies referencing the same VPN tunnel, each tunnel policy pair creates an individual IPsec SA with the remote peer. Each SA counts as an individual VPN tunnel.	Because the route, not the policy, determines which traffic goes through the tunnel, multiple policies can be supported with a single SA or VPN.
In a policy-based VPN, the action must be permit and must include a tunnel.	In a route-based VPN, the regulation of traffic is not coupled to the means of its delivery.
The exchange of dynamic routing information is not supported in policy-based VPNs.	Route-based VPNs support the exchange of dynamic routing information through VPN tunnels. You can enable an instance of a dynamic routing protocol, such as OSPF, on an st0 interface that is bound to a VPN tunnel.
Use a policy-based VPN when traffic-selection requirements cannot be defined using routes.	Route-based VPNs uses routes to specify the traffic sent to a tunnel; a policy does not specifically reference a VPN tunnel.
Policy-based VPNs cannot be used for hub-and-spoke topologies.	Route-based configurations are used for hub-and-spoke topologies.

Table 76: Comparison Between Policy-Based VPNs and Route-Based VPNs *(Continued)*

Policy-Based VPNs	Route-Based VPNs
When a tunnel does not connect large networks running dynamic routing protocols and you do not need to conserve tunnels or define various policies to filter traffic through the tunnel, use policy-based VPN tunnel.	With route-based VPNs, a policy does not specifically reference a VPN tunnel.
Policy-based VPN tunnels are required for remote-access (dial-up) VPN configurations.	Route-based VPNs do not support remote-access (dial-up) VPN configurations.
Policy-based VPNs might be required if the third party requires separate SAs for each remote subnet.	Route-based VPNs might not work correctly with some third-party vendors.
Policy-based VPNs cannot be used if NAT is required for tunneled traffic.	Route-based VPNs support NAT for st0 interfaces.
With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy.	When the security device does a route lookup to find the interface through which it must send traffic to reach an address, it finds a route through a secure tunnel (st0) interface. With a route-based VPN tunnel, you can consider a tunnel as a means for delivering traffic, and can consider the policy as a method for either permitting or denying the delivery of that traffic.

Proxy ID is supported for both route-based and policy-based VPNs. Route-based tunnels also offer the usage of multiple traffic selectors also known as multi-proxy ID. A traffic selector is an agreement between IKE peers to permit traffic through a tunnel, if the traffic matches a specified pair of local and remote IP address prefix, source port range, destination port range, and protocol. You define a traffic selector within a specific route-based VPN, which can result in multiple Phase 2 IPsec SAs. Only traffic that conforms to a traffic selector is permitted through an SA. The traffic selector is commonly required when remote gateway devices are non-Juniper Networks devices.

Shared Point-to-Point st0 Interface

SUMMARY

In this topic, you'll learn about sharing the point-to-point st0 logical interface between multiple VPN objects.

IN THIS SECTION

- Benefits | 483
- Limitations | 483
- Sample Configuration | 484

Junos OS supports sharing of point-to-point st0 logical interface when you run IPsec VPN service using the iked process to provide a migration path from the kmd process. You can configure multiple VPNs objects to share a point-to-point st0 interface if you meet the following prerequisites:

- You've configured explicit traffic selectors.
- You've not used wildcard network mask in your configuration.

Read further to understand the benefits and limitations of shared point-to-point st0 interface.

Benefits

- Eliminates the logical interface (IFL) limit on st0 interface.
- Negotiates multiple security associations (SAs) for a single IKE gateway when you have multiple subnets.
- Helps migrating policy-based VPNs to route-based VPNs.
- Eliminates the manual management of static routes to the tunnel using auto route insertion (ARI) with traffic selectors.
- Eliminates the need for next hop tunnel binding (NHTB) as shared st0 interface is not just limited to point-to-multipoint mode.

Limitations

IPsec VPNs cannot share point-to-point st0 interface if:

- Proxy ID is configured.
- Explicit traffic selectors are not configured.

- One VPN object has proxy ID configured and the other VPN object has default traffic selector configured.
- One VPN object has explicit traffic selector configured and the other VPN object has default traffic selector configured.
- One VPN object has proxy ID configured and the other VPN object has explicit traffic selector configured.

Sample Configuration

You can bind the same st0 interface in multiple IPsec VPN objects. You can configure two different IKE gateways with two different IPsec VPN objects binding to the same st0 interface.

```
[edit security ipsec]
user@host# show
vpn vpn1 {
    bind-interface st0.0;
    ike {
            gateway gw1;
            ipsec-policy ipsec_pol;
    }
    traffic-selector ts1 {
        local-ip 192.168.2.0/24;
        remote-ip 10.0.2.0/24;
    }
}
vpn vpn2 {
    bind-interface st0.0;
    ike {
             gateway gw2;
             ipsec-policy ipsec_pol;
    }
    traffic-selector ts1 {
        local-ip 192.168.3.0/24;
        remote-ip 10.0.3.0/24;
    }
}
```

Migrate Policy-Based VPNs to Route-Based VPNs

SUMMARY

Read this topic if you plan to migrate your configuration from policy-based VPNs to route-based VPNs using the shared point-to-point st0 interface.

IN THIS SECTION

- Limitations | 485
- Sample Configuration | 486

Although the SRX Series Firewalls support policy-based VPNs on firewalls that run the IPsec VPN using the kmd process, there are associated limitations. While the policy can control the traffic entering the VPN tunnel in terms of the protocol and the port number of an application, IKEv1 doesn't support protocol or port negotiation in security association (SA) negotiation. So the firewall cannot perform granular control of traffic with policy-based VPNs. We recommend that you migrate your policy-based VPNs to route-based VPNs.

To migrate from policy-based VPNs to route-based VPNs, perform the following steps:

- Deactivate the IPsec VPN objects that are running in your Junos OS device using the kmd process.
- Install the `junos-ike` package to run IPsec VPN service using the iked process. See install junos-ike package.
- Configure the prerequisites related to the shared point-to-point st0 interface.
- Activate the previously deactivated IPsec VPN objects with the shared point-to-point st0 interface.

We recommend that you carry out migration using your migration best practices to minimise the downtime.

Limitations

- You cannot switch back to the kmd-based IPsec VPN service once you migrate to the iked process with shared point-to-point st0 interface.
- The policy-based VPNs implicitly enforce the sequence order in which the policies are configured when a policy lookup if performed on the data traffic. But the route-based VPNs do not enforce the sequence order in which VPNs are configured, even with traffic selectors, because the sequence is governed by the metric per traffic selector per VPN configuration.

Sample Configuration

Before the migration, let's consider you've the following configuration for the policy-based IPsec VPN that uses the kmd process. Note that the support for policy-based VPNs is available with IKEv1 only. In this configuration, if the security policy matches the criteria, the device directs the traffic to the VPN tunnel.

```
[edit security policies]
user@host# show
from-zone zone1 to-zone zone2 {
    policy policy1 {
        match {
            source-address 192.168.2.0/24;
            destination-address 10.0.2.0/24;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn vpn1;
                }
            }
        }
    }
}
from-zone zone3 to-zone zone4 {
    policy policy2 {
        match {
            source-address 192.168.3.0/24;
            destination-address 10.0.3.0/24;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn vpn2;
                }
            }
        }
```

```
    }
}
```

```
[edit security ipsec]
user@host# show
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 2400;
}
policy ipsec_pol {
    proposals ipsec_prop;
}
vpn vpn1 {
    ike {
            gateway gw1;
            ipsec-policy ipsec_pol;
    }
    establish-tunnels immediately;
}
vpn vpn2 {
    ike {
             gateway gw2;
             ipsec-policy ipsec_pol;
    }
    establish-tunnels immediately;
}
```

After the migration, you'll notice the following configuration. Note that the support for multiple traffic selectors, port, and protocol is not available with IKEv1. You must bind the VPN objects to the same st0 interface for the IPsec VPN service that uses the iked process. You can configure two different IKE gateways with two different IPsec VPN objects binding to the same st0 interface with explicit traffic selectors configuration.

```
[edit security ipsec]
user@host# show
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-256-cbc;
```

```
    lifetime-seconds 2400;
}
policy ipsec_pol {
    proposals ipsec_prop;
}
vpn vpn1 {
    bind-interface st0.0;
    ike {
            gateway gw1;
            ipsec-policy ipsec_pol;
    }
    traffic-selector ts1 {
        local-ip 192.168.2.0/24;
        remote-ip 10.0.2.0/24;
    }
    establish-tunnels immediately;
}
vpn vpn2 {
    bind-interface st0.0;
    ike {
             gateway gw2;
             ipsec-policy ipsec_pol;
    }
    traffic-selector ts1 {
        local-ip 192.168.3.0/24;
        remote-ip 10.0.3.0/24;
    }
    establish-tunnels immediately;
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.4R1	Support for the shared point-to-point st0 interface with the iked process added in Junos OS Release 24.4R1.
24.4R1	We've introduced support for migrating policy-based VPNs to route-based VPNs using the shared point-to-point st0 interface with the iked process in Junos OS Release 24.4R1.

5

CHAPTER

Class-of-Service Based VPN

IN THIS CHAPTER

- CoS-Based IPsec VPNs | **490**

CoS-Based IPsec VPNs

SUMMARY

Read this topic to understand CoS-based IPsec VPNs and how you can configure the feature in Junos OS devices.

IN THIS SECTION

- Understand CoS-Based IPsec VPNs with Multiple IPsec SAs | **490**
- Understand Traffic Selectors and CoS-Based IPsec VPNs | **493**
- Example: Configure CoS-Based IPsec VPNs | **496**
- Understand CoS Support on st0 Interfaces | **525**
- Platform-Specific CoS-Based IPsec VPN Behavior | **527**

We support Junos class of service (CoS) feature that can provide multiple classes of service for VPNs. On the device, you can configure multiple forwarding classes for transmitting packets, define which packets are placed into each output queue, schedule the transmission service level for each queue, and manage congestion. IKE negotiates an IPsec tunnel for every Forwarding Class (FC) and each FC is mapped to a set of Differentiated Services Code Point (DSCP) values.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific CoS-Based IPsec VPN Behavior" on page 527 section for notes related to your platform.

Understand CoS-Based IPsec VPNs with Multiple IPsec SAs

IN THIS SECTION

- Overview | **491**
- Benefits | **491**
- Map FCs to IPsec SAs | **491**

- IPsec SA Negotiation | **492**
- Rekey | **492**
- Add or Delete FCs from a VPN | **493**
- Dead Peer Detection (DPD) | **493**
- Commands | **493**
- Supported VPN Features | **493**

In this topic, you'll learn about concepts that are related to class of service (CoS) based IPsec VPNs.

Overview

The CoS forwarding classes (FCs) that are configured on the Junos OS device can be mapped to IPsec security associations (SAs). Packets for each FC are mapped to a different IPsec SA, thus providing for CoS treatment on the local device and on intermediate routers. This feature is proprietary to Juniper Networks and works with supported Junos OS devices and Junos OS releases. The VPN peer device must be a Junos OS device that supports this feature or any other product that supports the same functionality in the same way as the Junos OS device.

Benefits

- Helps you ensure different data streams, with each tunnel using a separate set of security associations.
- Helps you to facilitate the IPsec VPN deployments where differentiated traffic is required, such as voice-over-IP.

Map FCs to IPsec SAs

You can configure up to 8 forwarding classes (FC) for a VPN with the `multi-sa forwarding-classes` configuration statement at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. The number of IPsec SAs negotiated with a peer gateway is based on the number of FCs configured for the VPN. The mapping of FCs to IPsec SAs applies to all traffic selectors that are configured for the VPN.

All IPsec SAs created for the FCs of a specific VPN are represented by the same tunnel ID. Tunnel-related events consider the state and statistics of all IPsec SAs. All IPsec SAs related to a tunnel are anchored to the same SPU or the same thread ID on the Junos OS device.

The order of FC configuration need not be same in both the peers. So Junos OS doesn't guarantee the same IPsec SA pair for the same FC on both ends of the VPN tunnel.

IPsec SA Negotiation

When you configure multiple FCs for a VPN, a unique IPsec SA is negotiated with the peer for each FC. In addition, a default IPsec SA is negotiated to send packets that do not match a configured FC. The default IPsec is negotiated even if the VPN peer device is not configured for FCs or does not support FC to IPsec SA mapping. The default IPsec SA is the first IPsec SA to be negotiated and the last SA to be torn down.

Depending on the number of FCs configured, when IPsec SAs are in the process of negotiating, packets might arrive with an FC for which an IPsec SA has yet to be negotiated. Until an IPsec SA for a given FC is negotiated, the traffic is sent to the default IPsec SA. A packet with an FC that does not match any of the IPsec SAs is sent on the default IPsec SA.

Mapping of FCs to IPsec SAs is done on the local VPN gateway. The local and peer gateways might have FCs configured in a different order. Each peer gateway maps FCs in the order in which IPsec SA negotiations are completed. Thus, the local and peer gateways might have different FC to IPsec SA mappings. A gateway stops negotiating new IPsec SAs once the configured number of FCs is reached. A peer gateway might initiate more IPsec SAs than the number of FCs configured on the local gateway. In this case, the local gateway accepts the additional IPsec SA requests—up to 18 IPsec SAs. The local gateway uses the other IPsec SAs only for decrypting incoming IPsec traffic. If a packet is received with an FC that does not match any configured FC, the packet is sent on the default FC IPsec SA.

If a delete notification is received for the default IPsec SA from the peer device, only the default IPsec SA is deleted and the default IPsec SA is negotiated newly. During this time, traffic which might go on default IPsec SA is be dropped. The VPN tunnel is brought down only if the default IPsec SA is the last SA.

If the `establish-tunnels immediately` option is configured and committed for the VPN, the Junos OS device negotiates IPsec SA without waiting for traffic to arrive. If negotiations do not complete for an IPsec SA for a configured FC, negotiations are retried every 60 seconds.

If the `establish-tunnels on-traffic` option is configured for the VPN, the Junos OS device negotiates IPsec SAs when the first data packet arrives; the FC for the first packet does not matter. With either option, the default IPsec SA is negotiated first, then each IPsec SA is negotiated one by one in the order in which the FCs are configured on the device.

Rekey

When using multiple SAs with Differentiated Services Code Point (DSCP) traffic steering with traffic selectors, the following behavior occurs during rekey—When the traffic selectors perform rekeying, if one or more of the traffic selectors are unable to rekey for any reason, the specific SA is brought down when the lifetime expires. In this case, traffic that use to match the specific SA is sent through the default traffic selector instead.

Add or Delete FCs from a VPN

When FCs are added or deleted from a VPN, the IKE and IPsec SAs for the VPN are brought up or down and restarts the negotiations. The `clear security ipsec security-associations` command clears all IPsec SAs.

Dead Peer Detection (DPD)

When DPD is configured with this feature, the `optimized` mode sends probes only when there is outgoing traffic and no incoming traffic on any of the IPsec SA. While the `probe-idle` mode sends probes only when there is no outgoing and no incoming traffic on any of the IPsec SAs. VPN monitoring is not supported with DPD feature.

Commands

The `show security ipsec sa details index` *`tunnel-id`* command displays all IPsec SA details including the FC name.

The `show security ipsec stats index` *`tunnel-id`* command displays statistics for each FC.

Supported VPN Features

The following VPN features are supported with CoS-based IPsec VPNs:

- Route-based site-to-site VPNs.
- Traffic selectors.
- AutoVPN.
- Auto Discovery VPNs (ADVPNs).
- IKEv2. IKEv1 is not supported.
- Dead peer detection (DPD). VPN monitoring is not supported.
- PMI is not supported.

Understand Traffic Selectors and CoS-Based IPsec VPNs

A traffic selector is an agreement between the IKE peers to permit traffic through a VPN tunnel if the traffic matches a specified pair of local and remote addresses. Only traffic that conforms to a traffic selector is permitted through the associated security association (SA).

The CoS-based IPsec VPN feature supports the following scenarios

- One or multiple traffic selectors in a route-based site-to-site VPN with the same FCs.
- Multiple traffic selectors, with different FCs for each traffic selector. This scenario requires separate VPN configurations.

This topic describes the VPN configurations and the IPsec SA that are negotiated for each scenario.

In the following scenarios, three FCs are configured on the Junos OS device:

```
forwarding-classes {
    queue 7 voip-data;
    queue 6 web-data;
    queue 5 control-data;
                    }
```

In the first scenario, VPN vpn1 is configured with a single traffic selector ts1 and the three FCs. In this configuration, four IPsec SAs are negotiated for traffic selector ts1—one for the default IPsec SA and three for the IPsec SAs that are mapped to FCs.

```
ipsec {
    vpn vpn1 {
       ts1 {
           local-ip 192.168.3.0/24;
           remote-ip 192.168.4.0/24;
              }multi-sa {
        forwarding-class web-data;
           forwarding-class voip-data
           forwarding-class control-data;
          }
       }
   }
```

In the second scenario, VPN vpn1 is configured with two traffic selectors ts1 and ts2 and the three FCs. In this configuration, four IPsec SAs are negotiated for traffic selector ts1 and four IPsec SAs are negotiated for traffic selector ts2. For each traffic selector, there is one IPsec SA negotiated for the default IPsec SA and three IPsec SAs negotiated for the IPsec SAs that are mapped to FCs.

```
ipsec {
    vpn vpn1 {
       ts1 {
```

```
            local-ip 192.168.3.0/24;
            remote-ip 192.168.4.0/24;
               }
        ts2 {
            local-ip 192.168.6.0/24;
            remote-ip 192.168.7.0/24;
          }

    multi-sa {
        forwarding-class web-data;
           forwarding-class voip-data
           forwarding-class control-data;
         }
       }
   }
```

In the third scenario, traffic selectors ts1 and ts2 support different sets of FCs. The traffic selectors need to be configured for different VPNs. In this configuration, four IPsec SAs are negotiated for traffic selector ts1 in VPN vpn1—one for the default IPsec SA and three for the IPsec SAs that are mapped to FCs.

```
ipsec {
    vpn vpn1 {
        bind-interface st0.0;
        ts1 {
            local-ip 192.168.3.0/24;
            remote-ip 192.168.4.0/24;
            }
        multi-sa {
         forwarding-class web-data;
         forwarding-class voip-data;
            forwarding-class control-data;
                 }
     vpn vpn2 {
         bind-interface st0.0;
         ts2 {
             local-ip 192.168.6.0/24;
             remote-ip 192.168.7.0/24;
             }
        multi-sa {
          forwarding-class web-data;
            forwarding-class voip-data;
```

```
        }
    }
```

SEE ALSO

Understand CoS-Based IPsec VPNs with Multiple IPsec SAs | 490

Example: Configure CoS-Based IPsec VPNs | 496

Example: Configure CoS-Based IPsec VPNs

IN THIS SECTION

- Requirements | 496
- Overview | 497
- Configuration | 500
- Verification | 521

This example shows how to configure a CoS-based IPsec VPNs with multiple IPsec SAs to allow packets mapping for each forwarding class to a different IPsec SA, thus providing for CoS treatment on the local device and on intermediate routers.

This feature is proprietary to Juniper Networks and only works with supported Junos OS devices and Junos OS releases. The VPN peer device must be an Junos OS device that supports this feature.

Requirements

This example uses the following hardware:

- A firewall with supported Junos OS

Before you begin:

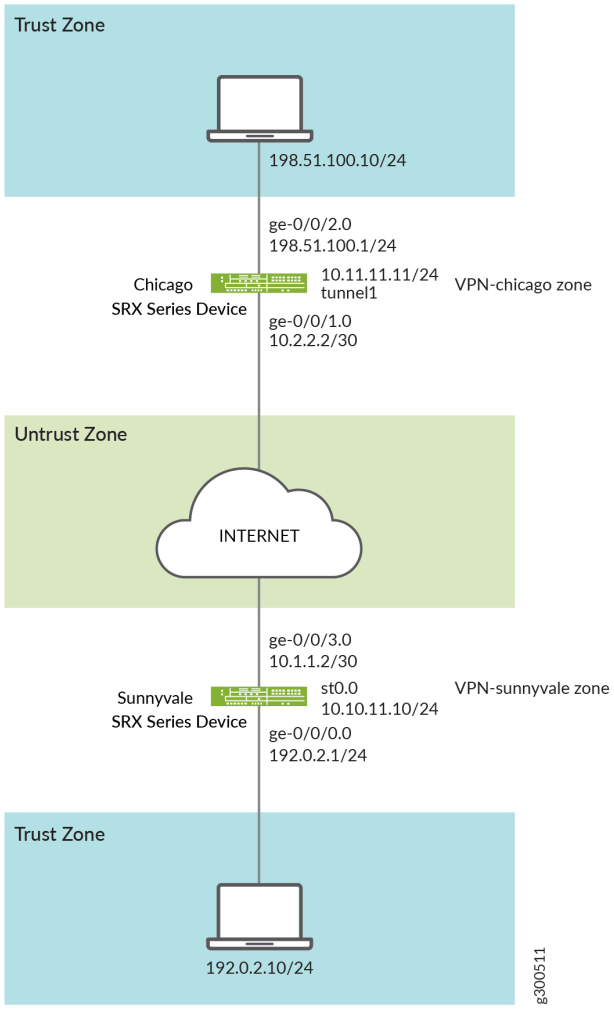
- Understand how Class of service (CoS) forwarding classes (FCs) configured on the firewall can be mapped to IPsec security associations (SAs).
- Understand Traffic Selectors and CoS-Based IPsec VPNs.

Overview

In this example, you configure an IPsec route-based VPN for a branch office in Chicago, because you do not need to conserve tunnel resources or configure many security policies to filter traffic through the tunnel. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale.

Figure 36 on page 497 shows an example of an IPsec route-based VPN topology. In this topology, one firewall is located in Sunnyvale, and one firewall is located in Chicago.

Figure 36: IPsec Route-Based VPN Topology



In this example, you configure interfaces, an IPv4 default route and security zones. Then you configure IKE, IPsec, a security policy, and CoS parameters. See Table 77 on page 498 through Table 80 on page 500.

Table 77: Interface, Static Route, and Security Zone Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	192.0.2.1/24
	ge-0/0/3.0	10.1.1.2/30
	st0.0 (tunnel interface)	10.10.11.10/24
Static routes	0.0.0.0/0 (default route)	The next hop is st0.0.
Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.
	untrust	• All system services are allowed. • The ge-0/0/3.0 interface is bound to this zone.
	vpn	The st0.0 interface is bound to this zone.

Table 78: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ike-proposal	• Authentication method: rsa-signatures • Diffie-Hellman group: group14 • Authentication algorithm: sha-256 • Encryption algorithm: aes-256-cbc

Table 78: IKE Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Policy	ike-policy	• Mode: main • Proposal reference: ike-proposal • IKE policy authentication method: rsa-signatures
Gateway	gw-sunnyvale	• IKE policy reference: ike-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2

Table 79: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha-256 • Encryption algorithm: aes-256-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop
VPN	ipsec_vpn1	• IKE gateway reference: gw-chicago • IPsec policy reference: ipsec_pol

Table 80: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

The security policy permits traffic from the trust zone to the vpn zone.	vpn	• Match criteria:
		  • source-address sunnyvale
		  • destination-address chicago
		  • application any
		• Action: permit

The security policy permits traffic from the vpn zone to the trust zone.	vpn	• Match criteria:
		  • source-address chicago
		  • destination-address sunnyvale
		  • application any
		• Action: permit

Configuration

IN THIS SECTION

- Configuring Basic Network and Security Zone Information | **501**
- Configuring CoS | **505**
- Configuring IKE | **512**
- Configuring IPsec | **515**
- Configuring Security Policies | **519**

Configuring Basic Network and Security Zone Information

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
set interfaces st0 unit 0 family inet address 10.10.11.10/24
set routing-options static route 0.0.0.0/0 next-hop st0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone vpn-chicago interfaces st0.0
set security zones security-zone vpn-chicago host-inbound-traffic protocols all
set security zones security-zone vpn-chicago host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone untrust host-inbound-traffic protocols all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure interface, static route, and security zone information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
user@host# set interfaces st0 unit 0 family inet address 10.10.11.10/24
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop st0.0
```

3. Configure the untrust security zone.

```
[edit ]
user@host# edit security zones security-zone untrust
```

4. Specify allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic protocols all
```

5. Assign an interface to the security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/3.0
```

6. Specify allowed system services for the security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services ike
```

7. Configure the trust security zone.

```
[edit]
user@host# edit security zones security-zone trust
```

8. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/0.0
```

9. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
```

10. Configure the vpn security zone.

```
[edit]
user@host# edit security zones security-zone vpn
```

11. Assign an interface to the security zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set interfaces st0.0
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.1.2/30;
        }
```

```
    }
}
st0 {
    unit 0 {
        family inet {
        address 10.10.11.10/24;
        }
    }
}
```

```
[edit]
user@host# show routing-options
static {
    route 0.0.0.0/0 next-hop st0.0;
}
```

```
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
```

```
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn-chicago {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring CoS

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set class-of-service classifiers dscp ba-classifier import default
set class-of-service classifiers dscp ba-classifier forwarding-class best-effort loss-priority
high code-points 000000
set class-of-service classifiers dscp ba-classifier forwarding-class ef-class loss-priority high
code-points 000001
set class-of-service classifiers dscp ba-classifier forwarding-class af-class loss-priority high
code-points 001010
set class-of-service classifiers dscp ba-classifier forwarding-class network-control loss-
priority high code-points 000011
set class-of-service classifiers dscp ba-classifier forwarding-class res-class loss-priority
high code-points 000100
set class-of-service classifiers dscp ba-classifier forwarding-class web-data loss-priority high
code-points 000101
set class-of-service classifiers dscp ba-classifier forwarding-class control-data loss-priority
```

```
high code-points 000111
set class-of-service classifiers dscp ba-classifier forwarding-class voip-data loss-priority
high code-points 000110
set class-of-service forwarding-classes queue 7 voip-data
set class-of-service forwarding-classes queue 6 control-data
set class-of-service forwarding-classes queue 5 web-data
set class-of-service forwarding-classes queue 4 res-class
set class-of-service forwarding-classes queue 2 af-class
set class-of-service forwarding-classes queue 1 ef-class
set class-of-service forwarding-classes queue 0 best-effort
set class-of-service forwarding-classes queue 3 network-control
set class-of-service interfaces ge-0/0/3 unit 0 classifiers dscp ba-classifier
set class-of-service interfaces ge-0/0/3 unit 0 scheduler-map sched_1
set class-of-service scheduler-maps sched_1 forwarding-class voip-data scheduler Q7
set class-of-service scheduler-maps sched_1 forwarding-class control-data scheduler Q6
set class-of-service scheduler-maps sched_1 forwarding-class web-data scheduler Q5
set class-of-service scheduler-maps sched_1 forwarding-class res-class scheduler Q4
set class-of-service scheduler-maps sched_1 forwarding-class af-class scheduler Q2
set class-of-service scheduler-maps sched_1 forwarding-class ef-class scheduler Q1
set class-of-service scheduler-maps sched_1 forwarding-class best-effort scheduler Q0
set class-of-service scheduler-maps sched_1 forwarding-class network-control scheduler Q3
set class-of-service schedulers Q7 transmit-rate percent 5
set class-of-service schedulers Q7 priority strict-high
set class-of-service schedulers Q6 transmit-rate percent 25
set class-of-service schedulers Q6 priority high
set class-of-service schedulers Q5 transmit-rate remainder
set class-of-service schedulers Q5 priority high
set class-of-service schedulers Q4 transmit-rate percent 25
set class-of-service schedulers Q4 priority medium-high
set class-of-service schedulers Q3 transmit-rate remainder
set class-of-service schedulers Q3 priority medium-high
set class-of-service schedulers Q2 transmit-rate percent 10
set class-of-service schedulers Q2 priority medium-low
set class-of-service schedulers Q1 transmit-rate percent 10
set class-of-service schedulers Q1 priority medium-low
set class-of-service schedulers Q0 transmit-rate remainder
set class-of-service schedulers Q0 priority low
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure CoS:

1. Configure behavior aggregate classifiers for DiffServ CoS.

```
[edit class-of-service]
user@host# edit classifiers dscp ba-classifier
user@host# set import default
```

2. Configure a best-effort forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class best-effort loss-priority high code-points 000000
```

3. Define the DSCP value to be assigned to the forwarding class.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class ef-class loss-priority high code-points 000001
user@host# set forwarding-class af-class loss-priority high code-points 001010
user@host# set forwarding-class network-control loss-priority high code-points 000011
user@host# set forwarding-class res-class loss-priority high code-points 000100
user@host# set forwarding-class web-data loss-priority high code-points 000101
user@host# set forwarding-class control-data loss-priority high code-points 000111
user@host# set forwarding-class voip-data loss-priority high code-points 000110
```

4. Define eight forwarding classes (queue names) for the eight queues.

```
[edit class-of-service forwarding-classes]
user@host# set queue 7 voip-data
user@host# set queue 6 control-data
user@host# set queue 5 web-data
user@host# set queue 4 res-class
user@host# set queue 2 af-class
user@host# set queue 1 ef-class
user@host# set queue 0 best-effort
user@host# set queue 3 network-control
```

5. Configure classifiers on the ingress (ge) interfaces.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/3 unit 0 classifiers dscp ba-classifier
```

6. Apply the scheduler map to the ge interface.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/3 unit 0 scheduler-map sched_1
```

7. Configure the scheduler map to associate schedulers with defined forwarding classes.

```
[edit class-of-service]
user@host# set scheduler-maps sched_1 forwarding-class voip-data scheduler Q7
user@host# set scheduler-maps sched_1 forwarding-class control-data scheduler Q6
user@host# set scheduler-maps sched_1 forwarding-class web-data scheduler Q5
user@host# set scheduler-maps sched_1 forwarding-class res-class scheduler Q4
user@host# set scheduler-maps sched_1 forwarding-class af-class scheduler Q2
user@host# set scheduler-maps sched_1 forwarding-class ef-class scheduler Q1
user@host# set scheduler-maps sched_1 forwarding-class best-effort scheduler Q0
user@host# set scheduler-maps sched_1 forwarding-class network-control scheduler Q3
```

8. Define the schedulers with priority and transmit rates.

```
[edit set class-of-service]
user@host# set schedulers Q7 transmit-rate percent 5
user@host# set schedulers Q7 priority strict-high
user@host# set schedulers Q6 transmit-rate percent 25
user@host# set schedulers Q6 priority high
user@host# set schedulers Q5 transmit-rate remainder
user@host# set schedulers Q5 priority high
user@host# set schedulers Q4 transmit-rate percent 25
user@host# set schedulers Q4 priority medium-high
user@host# set schedulers Q3 transmit-rate remainder
user@host# set schedulers Q3 priority medium-high
user@host# set schedulers Q2 transmit-rate percent 10
user@host# set schedulers Q2 priority medium-low
user@host# set schedulers Q1 transmit-rate percent 10
user@host# set schedulers Q1 priority medium-low
```

```
user@host# set schedulers Q0 transmit-rate remainder
user@host# set schedulers Q0 priority low
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
classifiers {
    dscp ba-classifier {
        import default;
        forwarding-class best-effort {
            loss-priority high code-points 000000;
        }
        forwarding-class ef-class {
            loss-priority high code-points 000001;
        }
        forwarding-class af-class {
            loss-priority high code-points 001010;
        }
        forwarding-class network-control {
            loss-priority high code-points 000011;
        }
        forwarding-class res-class {
            loss-priority high code-points 000100;
        }
        forwarding-class web-data {
            loss-priority high code-points 000101;
        }
        forwarding-class control-data {
            loss-priority high code-points 000111;
        }
        forwarding-class voip-data {
            loss-priority high code-points 000110;
        }
    }
}
forwarding-classes {
```

```
    queue 7 voip-data;
    queue 6 control-data;
    queue 5 web-data;
    queue 4 res-class;
    queue 2 af-class;
    queue 1 ef-class;
    queue 0 best-effort;
    queue 3 network-control;
}
interfaces {
    ge-0/0/3 {
        unit 0 {
            classifiers {
                dscp ba-classifier;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            scheduler-map sched_1;
        }
    }
}
scheduler-maps {
    sched_1 {
        forwarding-class voip-data scheduler Q7;
        forwarding-class control-data scheduler Q6;
        forwarding-class web-data scheduler Q5;
        forwarding-class res-class scheduler Q4;
        forwarding-class af-class scheduler Q2;
        forwarding-class ef-class scheduler Q1;
        forwarding-class best-effort scheduler Q0;
        forwarding-class network-control scheduler Q3;
    }
}
schedulers {
    Q7 {
        transmit-rate percent 5;
        priority strict-high;
    }
    Q6 {
        transmit-rate percent 25;
        priority high;
```

```
    }
    Q5 {
        transmit-rate {
            remainder;
        }
        priority high;
    }
    Q4 {
        transmit-rate percent 25;
        priority medium-high;
    }
    Q3 {
        transmit-rate {
            remainder;
        }
        priority medium-high;
    }
    Q2 {
        transmit-rate percent 10;
        priority medium-low;
    }
    Q1 {
        transmit-rate percent 10;
        priority medium-low;
    }
    Q0 {
        transmit-rate {
            remainder;
        }
        priority low;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike-proposal authentication-method pre-shared-keys
set security ike proposal ike-proposal dh-group group14
set security ike proposal ike-proposal authentication-algorithm sha-256
set security ike proposal ike-proposal encryption-algorithm aes-256-cbc
set security ike policy ike-policy mode main
set security ike policy ike-policy proposals ike-proposal
set security ike policy ike-policy pre-shared-key ascii-text $ABC123
set security ike gateway gw-sunnyvale external-interface ge-0/0/3.0
set security ike gateway gw-sunnyvale ike policy ike-policy
set security ike gateway gw-sunnyvale address 10.2.2.2
set security ike gateway gw-sunnyvale version v2-only
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure IKE:

1. Create the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-proposal]
user@host# set dh-group group14
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-algorithm sha-256
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-proposal]
user@host# set encryption-algorithm aes-256-cbc
```

6. Create an IKE policy.

```
[edit security ike]
user@host# set policy ike-policy
```

7. Set the IKE policy mode.

```
[edit security ike policy ike-policy]
user@host# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-policy]
user@host# set proposals ike-proposal
```

9. Define the IKE policy authentication method.

```
[edit security ike policy ike-policy]
user@host# set pre-shared-key ascii-text $ABC123
```

10. Create an IKE gateway and define its external interface.

```
[edit security ike]
user@host# set gateway gw-sunnyvale external-interface ge-0/0/3.0
```

11. Define the IKE policy reference.

```
[edit security ike gateway gw-sunnyvale]
user@host# set ike policy ike-policy
```

12. Define the IKE gateway address.

```
[edit security ike gateway gw-sunnyvale]
user@host# set address 10.2.2.2
```

13. Define the IKE gateway version.

```
[edit security ike gateway gw-sunnyvale]
user@host# set version v2-only
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy ike-policy {
    mode main;
    proposals ike-proposal;
    pre-shared-key ascii-text "$ABC123";
```

```
}
gateway gw-sunnyvale {
    ike policy ike-policy;
    address 10.2.2.2;
    external-interface ge-0/0/3.0;
    version v2-only;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec traceoptions flag all
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha-256
set security ipsec proposal ipsec_prop encryption-algorithm aes256-cbc
set security ipsec proposal ipsec_prop lifetime-seconds 3600
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn ipsec_vpn1 bind-interface st0.0
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class ef-class
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class af-class
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class res-class
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class web-data
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class control-data
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class voip-data
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class network-control
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class best-effort
set security ipsec vpn ipsec_vpn1 ike gateway gw_sunnyvale
set security ipsec vpn ipsec_vpn1 ike ipsec-policy ipsec_pol
set security ipsec vpn ipsec_vpn1 establish-tunnels immediately
set security ipsec vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 local-ip 203.0.113.2/25
set security ipsec vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 remote-ip 192.0.2.30/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure IPsec:

1. Enable IPsec trace options.

```
[edit]
user@host# set security ipsec traceoptions flag all
```

2. Create an IPsec proposal.

```
[edit]
user@host# set security ipsec proposal ipsec_prop
```

3. Specify the IPsec proposal protocol.

```
[edit security ipsec proposal ipsec_prop]
user@host# set protocol esp
```

4. Specify the IPsec proposal authentication algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set authentication-algorithm hmac-sha-256
```

5. Specify the IPsec proposal encryption algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set encryption-algorithm aes256-cbc
```

6. Specify the lifetime (in seconds) of an IPsec security association (SA).

```
[set security ipsec proposal ipsec_prop]
user@host# set lifetime-seconds 3600
```

7. Create the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec_pol
```

8. Specify the IPsec proposal reference.

```
[edit security ipsec policy ipsec_pol]
user@host# set proposals ipsec_prop
```

9. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 bind-interface st0.0
```

10. Configure the forwarding class to the multiple IPsec SA.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class ef-class
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class af-class
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class res-class
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class web-data
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class control-data
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class voip-data
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class network-control
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class best-effort
```

11. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 ike gateway gw_sunnyvale
```

12. Specify the IPsec policies.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 ike ipsec-policy ipsec_pol
```

13. Specify that the tunnel be brought up immediately to negotiate IPsec SA when the first data packet arrives to be sent.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 establish-tunnels immediately
```

14. Configure local IP addresses for a traffic selector.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 local-ip 203.0.113.2/25
```

15. Configure remote IP addresses for a traffic selector.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 remote-ip 192.0.2.30/24
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
traceoptions {
    flag all;
}
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha-256;
    encryption-algorithm aes256-cbc;
}
proposal ipsec_prop {
    lifetime-seconds 3600;
}
policy ipsec_pol {
    proposals ipsec_prop;
}
vpn ipsec_vpn1 {
```

```
    bind-interface st0.0;
    multi-sa {
        forwarding-class ef-class;
        forwarding-class af-class;
        forwarding-class res-class;
        forwarding-class web-data;
        forwarding-class control-data;
        forwarding-class voip-data;
        forwarding-class network-control;
        forwarding-class best-effort;
    }
    ike {
        gateway gw_sunnyvale;
        ipsec-policy ipsec_pol;
    }
    traffic-selector ipsec_vpn1_TS1 {
        local-ip 203.0.113.2/25;
        remote-ip 192.0.2.30/24;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone vpn policy vpn match source-address sunnyvale
set security policies from-zone trust to-zone vpn policy vpn match destination-address chicago
set security policies from-zone trust to-zone vpn policy vpn match application any
set security policies from-zone trust to-zone vpn policy vpn then permit
set security policies from-zone vpn to-zone trust policy vpn match source-address chicago
set security policies from-zone vpn to-zone trust policy vpn match destination-address sunnyvale
set security policies from-zone vpn to-zone trust policy vpn match application any
set security policies from-zone vpn to-zone trust policy vpn then permit
```

Enable security policies trace options for troubleshooting the policy-related issues.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the vpn zone.

```
[edit security policies from-zone trust to-zone vpn]
user@host# set policy vpn match source-address sunnyvale
user@host# set policy vpn match destination-address chicago
user@host# set policy vpn match application any
user@host# set policy vpn then permit
```

2. Create the security policy to permit traffic from the vpn zone to the trust zone.

```
[edit security policies from-zone vpn to-zone trust]
user@host# set policy vpn match source-address chicago
user@host# set policy vpn match destination-address sunnyvale
user@host# set policy vpn match application any
user@host# set policy vpn then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone vpn {
    policy vpn {
        match {
            source-address sunnyvale;
            destination-address chicago;
            application any;
        }
```

```
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn {
        match {
            source-address chicago;
            destination-address sunnyvale;
            application any;
        }
        then {
            permit;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IPsec Security Associations | 521

Confirm that the configuration is working properly.

Verifying IPsec Security Associations

Purpose

Verify the IPsec status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`131073`* `detail` and `show security ipsec statistics index` *`131073`* commands.

For brevity, the show command outputs does not display all the values of the configuration. Only a subset of the configuration is displayed. Rest of the configuration on the system has been replaced with ellipses (...).

```
user@host> show security ipsec security-associations
Total active tunnels: 2     Total Ipsec sas: 18
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:aes256/sha256 2d8e710b 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 5f3a3239 1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 5d227e19 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 5490da   1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 211fb8bc 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 dde29cd0 1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 49b64080 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 314afea0 1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 fec6f6ea 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 428a3a0d 1949/ unlim   -   root 500   5.0.0.1
...
```

```
user@host> show security ipsec security-associations index 131073 detail

ID: 131073 Virtual-system: root, VPN Name: IPSEC_VPN1
  Local Gateway: 4.0.0.1, Remote Gateway: 5.0.0.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 500, Nego#: 18, Fail#: 0, Def-Del#: 0 Flag: 0x600a39
  Multi-sa, Configured SAs# 9, Negotiated SAs#: 9
  Tunnel events:
    Mon Apr 23 2018 22:20:54 -0700: IPSec SA negotiation successfully completed (1 times)
    Mon Apr 23 2018 22:20:54 -0700: IKE SA negotiation successfully completed (2 times)
    Mon Apr 23 2018 22:20:18 -0700: User cleared IKE SA from CLI, corresponding IPSec SAs
cleared (1 times)
    Mon Apr 23 2018 22:19:55 -0700: IPSec SA negotiation successfully completed (2 times)
    Mon Apr 23 2018 22:19:23 -0700: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Mon Apr 23 2018 22:19:23 -0700: Bind-interface's zone received. Information updated (1 times)
    Mon Apr 23 2018 22:19:23 -0700: External interface's zone received. Information updated (1
times)
  Direction: inbound, SPI: 2d8e710b, AUX-SPI: 0
```

```
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1563 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: default
  Direction: outbound, SPI: 5f3a3239, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1563 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: default
  Direction: inbound, SPI: 5d227e19, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1551 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: best-effort
  Direction: outbound, SPI: 5490da, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1551 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
...
```

```
user@host> show security ipsec statistics index 131073

ESP Statistics:
  Encrypted bytes:              952
  Decrypted bytes:              588
```

```
  Encrypted packets:              7
  Decrypted packets:              7
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
FC Name     Encrypted Pkts  Decrypted Pkts  Encrypted bytes  Decrypted bytes
  best-effort 7                 7                 952               588
  custom_q1   0                 0                 0                 0
  custom_q2   0                 0                 0                 0
  network-control 0             0                 0                 0
  custom_q4   0                 0                 0                 0
  custom_q5   0                 0                 0                 0
  custom_q6   0                 0                 0                 0
  custom_q7   0                 0                 0                 0
  default     0                 0                 0                 0
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 131073. Use this value with the show security ipsec security-associations index command to get more information about this particular SA.
- There is one IPsec SA pair using port 500.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 1949/ unlim value indicates that the Phase lifetime expires in 1949 seconds, and that no lifesize has been specified, which indicates that it is unlimited.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.

The `show security ike security-associations index 131073 detail` command lists additional information about the SA with an index number of 131073:

- The local identity and remote identity make up the proxy ID for the SA. A proxy ID mismatch is one of the most common causes for a Phase failure. If no IPsec SA is listed, confirm that Phase proposals, including the proxy ID settings, are correct for both peers.

- Displays all the child SA details including forwarding class name.

The `show security ipsec statistics index 131073` command lists statistics for each forwarding class name.

- An error value of zero in the output indicates a normal condition.
- We recommend running this command multiple times to observe any packet loss issues across a VPN. Output from this command also displays the statistics for encrypted and decrypted packet counters, error counters, and so on.
- You must enable security flow trace options to investigate which ESP packets are experiencing errors and why.

SEE ALSO

Understand CoS-Based IPsec VPNs with Multiple IPsec SAs | 490

Understand Traffic Selectors and CoS-Based IPsec VPNs | 493

Example: Configure CoS-Based IPsec VPNs | 496

Introduction to IKE in Junos OS | 31

Understand CoS Support on st0 Interfaces

IN THIS SECTION

- Limitations of CoS Support on VPN st0 Interfaces | 526

You can configure class of service (CoS) features such as classifier, policer, queuing, scheduling, shaping, rewriting markers, and virtual channels on the secure tunnel interface (st0) for point-to-point VPNs.

The st0 tunnel interface is an internal interface that can be used by route-based VPNs to route cleartext traffics to an IPsec VPN tunnel. The following CoS features are supported on the firewall's st0 interface:

- Classifiers
- Policers
- Queuing, scheduling, and shaping
- Rewrite markers

- Virtual channels

Limitations of CoS Support on VPN st0 Interfaces

The following limitations apply to CoS support on VPN st0 interfaces:

- The maximum number for software queues is 2048. If the number of st0 interfaces exceeds 2048, not enough software queues can be created for all the st0 interfaces.
- Only route-based VPNs can apply CoS features on st0 interfaces. Table 81 on page 526 describes the st0 CoS feature support for different types of VPNs.

  Table 81: CoS Feature Support for VPN

Classifier Features	Site-to-Site VPN (P2P)	AutoVPN (P2P)	Site-to-Site/ AutoVPN /AD-VPN (P2MP)
Classifiers, policers, and rewriting markers	Supported	Supported	Supported
Queueing, scheduling, and shaping based on st0 logical interfaces	Supported	Not supported	Not supported
Queueing, scheduling, and shaping based on virtual channels	Supported	Supported	Supported

- When defining a CoS shaping rate on a st0 tunnel interface, consider the following restrictions:
  - The shaping rate on the tunnel interface must be less than that of the physical egress interface.
  - The shaping rate only measures the packet size that includes the inner Layer 3 cleartext packet with an ESP/AH header and an outer IP header encapsulation. The outer Layer 2 encapsulation added by the physical interface is not factored into the shaping rate measurement.
  - The CoS behavior works as expected when the physical interface carries the shaped GRE or IP-IP tunnel traffic only. If the physical interface carries other traffic, thereby lowering the available bandwidth for tunnel interface traffic, the CoS features do not work as expected.

SEE ALSO

Class of Service User Guide (Security Devices)

Platform-Specific CoS-Based IPsec VPN Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platforms.

Table 82: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX300, SRX320, SRX340, SRX345, and SRX550HM devices that support CoS-based IPsec VPNs, one st0 logical interface can bind to multiple VPN tunnels. However, since the eight queues for the st0 logical interface cannot reroute the traffic to different tunnels, these devices do not support pre-tunneling. You can use the virtual channel feature as a workaround on these devices. • On SRX550M, SRX5400, SRX5600, and SRX5800 devices that support CoS-based IPsec VPNs, the bandwidth limit and burst-size limit values in a policer configuration are per-SPU, rather than per-system limitation. The policer behavior matches that of the physical interface.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.4R1	Starting with Junos OS Release 24.4R1, support for CoS-based IPsec VPNs with the iked process is added.
17.4R1	Starting with Junos OS Release 17.4R1, support for listed CoS features is added for the st0 interface for SRX4600 devices.
15.1X49-D70	Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, support for queuing, scheduling, shaping, and virtual channels is added to the st0 interface for SRX5400, SRX5600, and SRX5800 devices. Support for all the listed CoS features is added for the st0 interface for SRX1500, SRX4100, and SRX4200 devices.

15.1X49-D60	Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, class of service (CoS) features such as classifier, policer, queuing, scheduling, shaping, rewriting markers, and virtual channels can now be configured on the secure tunnel interface (st0) for point-to-point VPNs.

RELATED DOCUMENTATION

- Class of Service User Guide (Security Devices)

6

CHAPTER

NAT-T

IN THIS CHAPTER

- Route-Based VPNs with NAT-T | **530**

Route-Based VPNs with NAT-T

SUMMARY

Read this topic to understand IPsec VPNs with NAT-T.

IN THIS SECTION

- Understanding NAT-T | **530**
- Example: Configuring a Route-Based VPN with the Responder behind a NAT Device | **531**
- Example: Configuring NAT-T with Dynamic Endpoint VPN | **568**
- Platform-Specific NAT-T with IPsec VPN Behavior | **590**

Network Address Translation-Traversal (NAT-T) is a method used for managing IP address translation-related issues encountered when the data protected by IPsec passes through a device configured with NAT for address translation.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific NAT-T with IPsec VPN Behavior" on page 590 section for notes related to your platform.

Understanding NAT-T

Network Address Translation-Traversal (NAT-T) is a method for getting around IP address translation issues encountered when data protected by IPsec passes through a NAT device for address translation. Any changes to the IP addressing, which is the function of NAT, causes IKE to discard packets. After detecting one or more NAT devices along the datapath during Phase 1 exchanges, NAT-T adds a layer of User Datagram Protocol (UDP) encapsulation to IPsec packets so they are not discarded after address translation. NAT-T encapsulates both IKE and ESP traffic within UDP with port 4500 used as both the source and destination port. Because NAT devices age out stale UDP translations, keepalive messages are required between the peers.

NAT-T is enabled by default therefore you must use the `no-nat-traversal` statement at the `[edit security ike gateway` *`gateway-name`* hierarchy level for disabling the NAT-T.

There are two broad categories of NAT:

- Static NAT, where there is a one-to-one relationship between the private and public addresses. Static NAT works in both inbound and outbound directions.
- Dynamic NAT, where there is a many-to-one or many-to-many relationship between the private and public addresses. Dynamic NAT works in the outbound direction only.

The location of a NAT device can be such that:

- Only the IKEv1 or IKEv2 initiator is behind a NAT device. Multiple initiators can be behind separate NAT devices. Initiators can also connect to the responder through multiple NAT devices.
- Only the IKEv1 or IKEv2 responder is behind a NAT device.
- Both the IKEv1 or IKEv2 initiator and the responder are behind a NAT device.

Dynamic endpoint VPN covers the situation where the initiator's IKE external address is not fixed and is therefore not known by the responder. This can occur when the initiator's address is dynamically assigned by an ISP or when the initiator's connection crosses a dynamic NAT device that allocates addresses from a dynamic address pool.

Configuration examples for NAT-T are provided for the topology in which only the responder is behind a NAT device and the topology in which both the initiator and responder are behind a NAT device. Site-to-site IKE gateway configuration for NAT-T is supported on both the initiator and responder. A remote IKE ID is used to validate a peer's local IKE ID during Phase 1 of IKE tunnel negotiation. Both the initiator and responder require a `local-identity` and a `remote-identity` setting.

SEE ALSO

IPsec Overview | 12

Example: Configuring a Route-Based VPN with the Responder behind a NAT Device

IN THIS SECTION

- Requirements | 532
- Overview | 532
- Configuration | 538
- Verification | 557

This example shows how to configure a route-based VPN with a responder behind a NAT device between a branch office and the corporate office.

Requirements

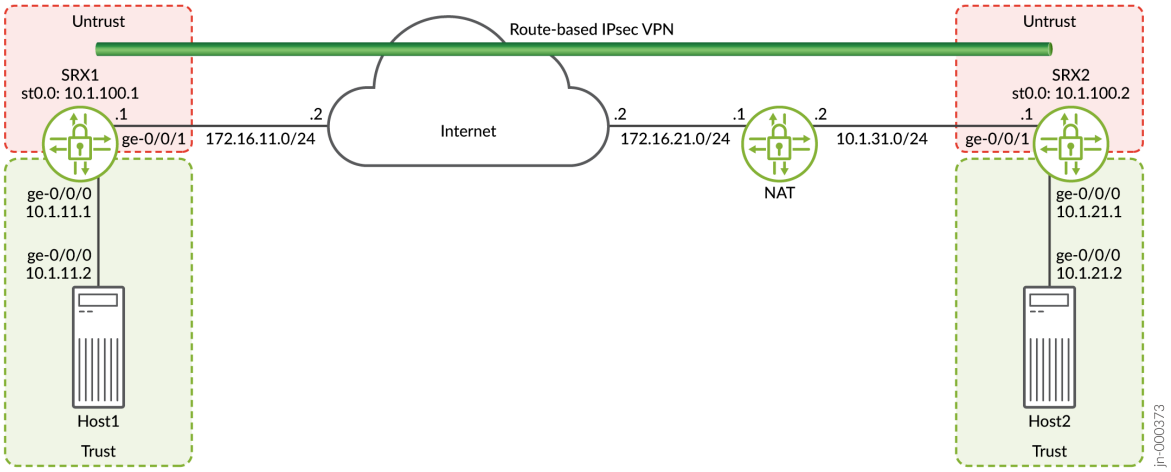
Before you begin, read "IPsec Overview" on page 12.

Overview

In this example, you configure a route-based VPN. Host1 will use the VPN to connect to their corporate headquarters on SRX2.

Figure 37 on page 532 shows an example of a topology for route-based VPN with only the responder behind a NAT device.

Figure 37: Route-Based VPN Topology with Only the Responder behind a NAT Device



In this example, you configure interfaces, IPsec, and security policies for both an initiator in SRX1 and a responder in SRX2. Then you configure IKE Phase 1 and IPsec Phase 2 parameters.

SRX1 sends packets with the destination address of 172.16.21.1 to establish the VPN. The NAT device translates the destination address to 10.1.31.1.

See Table 83 on page 533 through Table 85 on page 534 for specific configuration parameters used for the initiator in the examples.

Table 83: Interface, Routing Options, and Security Parameters for SRX1

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/1	172.16.11.1/24
	ge-0/0/0	10.1.11.1/24
	st0.0 (tunnel interface)	10.1.100.1/24
Static routes	10.1.21.0/24	The next hop is st0.0.
	172.16.21.1/32	The next hop is 172.16.11.2.
Security zones	untrust	• The system services of IKE and ping. • The ge-0/0/1.0 and the st0.0 interfaces are bound to this zone.
	trust	• Allow all system services. • Allow all protocols. • The ge-0/0/0.0 interface is bound to this zone.
Security policies	to-SRX2	Permit traffic from 10.1.11.0/24 in the trust zone to 10.1.21.0/24 in the untrust zone.
	from-SRX2	Permit traffic from 10.1.21.0/24 in the untrust zone to 10.1.11.0/24 in the trust zone.

Table 84: IKE Phase 1 Configuration Parameters for SRX1

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: 3des-cbc
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw1	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 172.16.21.1 • Local peer (initiator): branch_natt1@example.net • Remote peer (responder): responder_natt1@example.net

Table 85: IPsec Phase 2 Configuration Parameters for SRX1

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • Perfect forward secrecy (PFS) keys: group2

Table 85: IPsec Phase 2 Configuration Parameters for SRX1 *(Continued)*

Feature	Name	Configuration Parameters
VPN	vpn1	• IKE gateway reference: gw1 • IPsec policy reference: ipsec_pol • Bind to interface: st0.0 • Establish tunnels immediately

See Table 86 on page 535 through Table 88 on page 537 for specific configuration parameters used for the responder in the examples.

Table 86: Interface, Routing Options, and Security Parameters for SRX2

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/1	10.1.31.1/24
	ge-0/0/0	10.1.21.1/24
	st0.0 (tunnel interface)	10.1.100.2/24
Static routes	172.16.11.1/32	The next hop is 10.1.31.2.
	10.1.11.0/24	The next hop is st0.0.
Security zones	untrust	• Allow IKE and ping system services. • The ge-0/0/1.0 and the st0.0 interfaces are bound to this zone.

Table 86: Interface, Routing Options, and Security Parameters for SRX2 *(Continued)*

Feature	Name	Configuration Parameters
	trust	• Allow all system services. Allow all protocols. • The ge-0/0/0.0 interface is bound to this zone.
Security policies	to-SRX1	Permit traffic from 10.1.21.0/24 in the trust zone to 10.1.11.0/24 in the untrust zone.
	from-SRX1	Permit traffic from 10.1.11.0/24 in the untrust zone to 10.1.21.0/24 in the trust zone.

Table 87: IKE Phase 1 Configuration Parameters for SRX2

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: 3des-cbc
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text

Table 87: IKE Phase 1 Configuration Parameters for SRX2 *(Continued)*

Feature	Name	Configuration Parameters
Gateway	gw1	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 172.16.11.1 • Local peer (responder): responder_natt1@example.net • Remote peer (initiator): branch_natt1@example.net

Table 88: IPsec Phase 2 Configuration Parameters for SRX2

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • PFS keys: group2
VPN	vpn1	• IKE gateway reference: gw1 • IPsec policy reference: ipsec_pol • Bind to interface: st0.0 • Establish tunnels immediately

Configuration

IN THIS SECTION

- Configuring Interface, Routing Options, and Security Parameters for SRX1 | **538**
- Configuring IKE for SRX1 | **543**
- Configuring IPsec for SRX1 | **545**
- Configuring Interfaces, Routing Options, and Security Parameters for SRX2 | **547**
- Configuring IKE for SRX2 | **553**
- Configuring IPsec for SRX2 | **555**
- Configuration for the NAT Device | **557**

Configuring Interface, Routing Options, and Security Parameters for SRX1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security address-book book1 address Host1 10.1.11.0/24
set security address-book book1 attach zone trust
set security address-book book2 address Host2 10.1.21.0/24
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy to-SRX2 match source-address Host1
set security policies from-zone trust to-zone untrust policy to-SRX2 match destination-address
Host2
set security policies from-zone trust to-zone untrust policy to-SRX2 match application any
set security policies from-zone trust to-zone untrust policy to-SRX2 then permit
set security policies from-zone untrust to-zone trust policy from-SRX2 match source-address Host2
set security policies from-zone untrust to-zone trust policy from-SRX2 match destination-address
Host1
set security policies from-zone untrust to-zone trust policy from-SRX2 match application any
set security policies from-zone untrust to-zone trust policy from-SRX2 then permit
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces st0.0
```

```
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet address 10.1.11.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.11.1/24
set interfaces st0 unit 0 family inet address 10.1.100.1/24
set routing-options static route 10.1.21.0/24 next-hop st0.0
set routing-options static route 172.16.21.1/32 next-hop 172.16.11.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interfaces, static routes, and security parameters:

1. Configure the interfaces connected to the Internet, Host1, and the interface used for the VPN.

```
[edit]
user@SRX1# set interfaces ge-0/0/0 unit 0 family inet address 10.1.11.1/24
user@SRX1# set interfaces ge-0/0/1 unit 0 family inet address 172.16.11.1/24
user@SRX1# set interfaces st0 unit 0 family inet address 10.1.100.1/24
```

2. Configure static routes for the traffic that will use the VPN and for SRX1 to reach the NAT device.

```
[edit]
user@SRX1# set routing-options static route 10.1.21.0/24 next-hop st0.0
user@SRX1# set routing-options static route 172.16.21.1/32 next-hop 172.16.11.2
```

3. Configure the untrust security zone.

```
[edit]
user@SRX1# set security zones security-zone untrust host-inbound-traffic system-services ike
user@SRX1# set security zones security-zone untrust host-inbound-traffic system-services ping
user@SRX1# set security zones security-zone untrust interfaces st0.0
user@SRX1# set security zones security-zone untrust interfaces ge-0/0/1.0
```

4. Configure the trust security zone.

```
[edit]
user@SRX1# set security zones security-zone trust host-inbound-traffic system-services all
user@SRX1# set security zones security-zone trust host-inbound-traffic protocols all
user@SRX1# set security zones security-zone trust interfaces ge-0/0/0.0
```

5. Configure address books for the networks used in the security policies.

```
[edit]
user@SRX1# set security address-book book1 address Host1 10.1.11.0/24
user@SRX1# set security address-book book1 attach zone trust
user@SRX1# set security address-book book2 address Host2 10.1.21.0/24
user@SRX1# set security address-book book2 attach zone untrust
```

6. Create security policies to allow traffic between the hosts.

```
[edit]
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 match source-
address Host1
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 match
destination-address Host2
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 match
application any
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 then permit
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 match
source-address Host2
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 match
destination-address Host1
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 match
application any
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.11.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.16.11.1/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 10.1.100.1/24;
        }
    }
}
```

```
[edit]
user@SRX1# show routing-options
static {
    route 10.1.21.0/24 next-hop st0.0;
    route 172.16.21.1/32 next-hop 172.16.11.2;
}
```

```
[edit]
user@SRX1# show security
```

```
address-book {
    book1 {
        address Host1 10.1.11.0/24;
        attach {
            zone trust;
        }
    }
    book2 {
        address Host2 10.1.21.0/24;
        attach {
            zone untrust;
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy to-SRX2 {
            match {
                source-address Host1;
                destination-address Host2;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
        policy from-SRX2 {
            match {
                source-address Host2;
                destination-address Host1;
                application any;
            }
            then {
                permit;
            }
        }
    }
}
zones {
    security-zone untrust {
        host-inbound-traffic {
```

```
            system-services {
                ike;
                ping;
            }
        }
        interfaces {
            st0.0;
            ge-0/0/1.0;
        }
    }
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE for SRX1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_prop authentication-method pre-shared-keys
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode main
set security ike policy ike_pol proposals ike_prop
set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
```

```
set security ike gateway gw1 ike-policy ike_pol
set security ike gateway gw1 address 172.16.21.1
set security ike gateway gw1 local-identity user-at-hostname "srx1@example.com"
set security ike gateway gw1 remote-identity user-at-hostname "srx2@example.com"
set security ike gateway gw1 external-interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create an IKE Phase 1 proposal.

```
[edit]
user@SRX1# set security ike proposal ike_prop authentication-method pre-shared-keys
user@SRX1# set security ike proposal ike_prop dh-group group2
user@SRX1# set security ike proposal ike_prop authentication-algorithm sha1
user@SRX1# set security ike proposal ike_prop encryption-algorithm 3des-cbc
```

2. Create an IKE Phase 1 policy.

```
[edit]
user@SRX1# set security ike policy ike_pol mode main
user@SRX1# set security ike policy ike_pol proposals ike_prop
user@SRX1# set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
```

3. Configure the IKE Phase 1 gateway parameters. The gateway address should be the IP for the NAT device.

```
[edit security ike gateway gw1]
user@SRX1# set security ike gateway gw1 ike-policy ike_pol
user@SRX1# set security ike gateway gw1 address 172.16.21.1
user@SRX1# set security ike gateway gw1 local-identity user-at-hostname "srx1@example.com"
user@SRX1# set security ike gateway gw1 remote-identity user-at-hostname "srx2@example.com"
user@SRX1# set security ike gateway gw1 external-interface ge-0/0/1.0
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX1# show security ike
proposal ike_prop {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_pol {
    mode main;
    proposals ike_prop;
    pre-shared-key ascii-text “$9$xPn7-VwsgaJUHqp01IcSs2g”; ## SECRET-DATA
}
gateway gw1 {
    ike-policy ike_pol;
    address 172.16.21.1;
    local-identity user-at-hostname "srx1@example.com";
    remote-identity user-at-hostname "srx2@example.com";
    external-interface ge-0/0/1.0;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for SRX1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
```

```
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn vpn1 bind-interface st0.0
set security ipsec vpn vpn1 ike gateway gw1
set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
set security ipsec vpn vpn1 establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@SRX1# set security ipsec proposal ipsec_prop protocol esp
user@SRX1# set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
user@SRX1# set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
```

2. Create the IPsec Phase 2 policy.

```
[edit]
user@SRX1# set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
user@SRX1# set security ipsec policy ipsec_pol proposals ipsec_prop
```

3. Configure the IPsec VPN parameters.

```
[edit]
user@SRX1# set security ipsec vpn vpn1 bind-interface st0.0
user@SRX1# set security ipsec vpn vpn1 ike gateway gw1
user@SRX1# set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
user@SRX1# set security ipsec vpn vpn1 establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX1# show security ipsec
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
}
policy ipsec_pol {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn vpn1 {
    bind-interface st0.0;
    ike {
        gateway gw1;
        ipsec-policy ipsec_pol;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Interfaces, Routing Options, and Security Parameters for SRX2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security address-book book1 address Host2 10.1.21.0/24
set security address-book book1 attach zone trust
set security address-book book2 address Host1 10.1.11.0/24
```

```
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy to-SRX1 match source-address Host2
set security policies from-zone trust to-zone untrust policy to-SRX1 match destination-address
Host1
set security policies from-zone trust to-zone untrust policy to-SRX1 match application any
set security policies from-zone trust to-zone untrust policy to-SRX1 then permit
set security policies from-zone untrust to-zone trust policy from-SRX1 match source-address Host1
set security policies from-zone untrust to-zone trust policy from-SRX1 match destination-address
Host2
set security policies from-zone untrust to-zone trust policy from-SRX1 match application any
set security policies from-zone untrust to-zone trust policy from-SRX1 then permit
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet address 10.1.21.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.31.1/24
set interfaces st0 unit 0 family inet address 10.1.100.2/24
set routing-options static route 172.16.11.1/32 next-hop 10.1.31.2
set routing-options static route 10.1.11.0/24 next-hop st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interfaces, static routes, and security parameters:

1. Configure the interfaces connected to the Internet, Host2, and the interface used for the VPN.

```
[edit]
user@SRX2# set interfaces ge-0/0/0 unit 0 family inet address 10.1.21.1/24
user@SRX2# set interfaces ge-0/0/1 unit 0 family inet address 10.1.31.1/24
user@SRX2# set interfaces st0 unit 0 family inet address 10.1.100.2/24
```

2. Configure static routes for the traffic that will use the VPN and for SRX2 to reach SRX1.

```
[edit]
user@SRX2# set routing-options static route 172.16.11.1/32 next-hop 10.1.31.2
user@SRX2# set routing-options static route 10.1.11.0/24 next-hop st0.0
```

3. Configure the untrust security zone.

```
[edit]
user@SRX2# set security zones security-zone untrust host-inbound-traffic system-services ike
user@SRX2# set security zones security-zone untrust host-inbound-traffic system-services ping
user@SRX2# set security zones security-zone untrust interfaces ge-0/0/1.0
user@SRX2# set security zones security-zone untrust interfaces st0.0
```

4. Configure the trust security zone.

```
[edit]
user@SRX2# set security zones security-zone trust host-inbound-traffic system-services all
user@SRX2# set security zones security-zone trust host-inbound-traffic protocols all
user@SRX2# set security zones security-zone trust interfaces ge-0/0/0.0
```

5. Configure address books for the networks used in the security policies.

```
[edit]
user@SRX2# set security address-book book1 address Host2 10.1.21.0/24
user@SRX2# set security address-book book1 attach zone trust
user@SRX2# set security address-book book2 address Host1 10.1.11.0/24
user@SRX2# set security address-book book2 attach zone untrust
```

6. Create security policies to allow traffic between the hosts.

```
[edit]
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 match source-
address Host2
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 match
destination-address Host1
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 match
application any
```

```
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 then permit
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 match
source-address Host1
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 match
destination-address Host2
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 match
application any
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.21.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.31.1/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 10.1.100.2/24;
        }
```

```
    }
}
```

```
[edit]
user@SRX2# show routing-options
static {
    route 172.16.11.1/32 next-hop 10.1.31.2;
    route 10.1.11.0/24 next-hop st0.0;
}
```

```
[edit]
user@SRX2# show security
address-book {
    book1 {
        address Host2 10.1.21.0/24;
        attach {
            zone trust;
        }
    }
    book2 {
        address Host1 10.1.11.0/24;
        attach {
            zone untrust;
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy to-SRX1 {
            match {
                source-address Host2;
                destination-address Host1;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
```

```
        policy from-SRX1 {
            match {
                source-address Host1;
                destination-address Host2;
                application any;
            }
            then {
                permit;
            }
        }
    }
}
zones {
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ike;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
            st0.0;
        }
    }
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE for SRX2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_prop authentication-method pre-shared-keys
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode main
set security ike policy ike_pol proposals ike_prop
set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
set security ike gateway gw1 ike-policy ike_pol
set security ike gateway gw1 address 172.16.11.1
set security ike gateway gw1 local-identity user-at-hostname "srx2@example.com"
set security ike gateway gw1 remote-identity user-at-hostname "srx1@example.com"
set security ike gateway gw1 external-interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create an IKE Phase 1 proposal.

```
[edit]
user@SRX2# set security ike proposal ike_prop authentication-method pre-shared-keys
user@SRX2# set security ike proposal ike_prop dh-group group2
user@SRX2# set security ike proposal ike_prop authentication-algorithm sha1
user@SRX2# set security ike proposal ike_prop encryption-algorithm 3des-cbc
```

2. Create an IKE Phase 1 policy.

```
[edit]
user@SRX2# set security ike policy ike_pol mode main
```

```
user@SRX2# set security ike policy ike_pol proposals ike_prop
user@SRX2# set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
```

3. Configure the IKE Phase 1 gateway parameters. The gateway address should be the IP for SRX1.

```
[edit]
user@SRX2# set security ike gateway gw1 ike-policy ike_pol
user@SRX2# set security ike gateway gw1 address 172.16.11.1
user@SRX2# set security ike gateway gw1 local-identity user-at-hostname "srx2@example.com"
user@SRX2# set security ike gateway gw1 remote-identity user-at-hostname "srx1@example.com"
user@SRX2# set security ike gateway gw1 external-interface ge-0/0/1.0
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX2# show security ike
proposal ike_prop {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_pol {
    mode main;
    proposals ike_prop;
    pre-shared-key ascii-text "$9$mP5QF3/At0IE-VsYoa36/"; ## SECRET-DATA
}
gateway gw1 {
    ike-policy ike_pol;
    address 172.16.11.1;
    local-identity user-at-hostname "srx2@example.com";
    remote-identity user-at-hostname "srx1@example.com";
    external-interface ge-0/0/1.0;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for SRX2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn vpn1 bind-interface st0.0
set security ipsec vpn vpn1 ike gateway gw1
set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
set security ipsec vpn vpn1 establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@SRX2# set security ipsec proposal ipsec_prop protocol esp
user@SRX2# set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
user@SRX2# set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
```

2. Create the IPsec Phase 2 policy.

```
[edit]
user@SRX2# set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
user@SRX2# set security ipsec policy ipsec_pol proposals ipsec_prop
```

3. Configure the IPsec VPN parameters.

```
[edit]
user@SRX2# set security ipsec vpn vpn1 bind-interface st0.0
user@SRX2# set security ipsec vpn vpn1 ike gateway gw1
user@SRX2# set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
user@SRX2# set security ipsec vpn vpn1 establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX2# show security ipsec
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
}
policy ipsec_pol {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn vpn1 {
    bind-interface st0.0;
    ike {
        gateway gw1;
        ipsec-policy ipsec_pol;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuration for the NAT Device

CLI Quick Configuration

Static NAT is used in the example. Static NAT is bidirectional which means that traffic from 10.1.31.1 to 172.16.11.1 will also use the same NAT configuration.

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security nat static rule-set rule1 from zone untrust
set security nat static rule-set rule1 rule ipsec match source-address 172.16.11.1/32
set security nat static rule-set rule1 rule ipsec match destination-address 172.16.21.1/32
set security nat static rule-set rule1 rule ipsec then static-nat prefix 10.1.31.1/32
set security policies from-zone trust to-zone untrust policy allow-out match source-address any
set security policies from-zone trust to-zone untrust policy allow-out match destination-address
any
set security policies from-zone trust to-zone untrust policy allow-out match application any
set security policies from-zone trust to-zone untrust policy allow-out then permit
set security policies from-zone untrust to-zone trust policy allow-out-in match source-address
any
set security policies from-zone untrust to-zone trust policy allow-out-in match destination-
address any
set security policies from-zone untrust to-zone trust policy allow-out-in match application any
set security policies from-zone untrust to-zone trust policy allow-out-in then permit
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet address 172.16.21.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.31.2/24
set routing-options static route 172.16.11.0/24 next-hop 172.16.21.2
```

Verification

IN THIS SECTION

Verifying the IKE Phase 1 Status on SRX1 | **558**

- Verifying IPsec Security Associations on SRX1 | **560**
- Verifying the IKE Phase 1 Status on SRX2 | **562**
- Verifying IPsec Security Associations on SRX2 | **564**
- Verifying Host-to-Host Reachability | **567**

To confirm that the configuration is working properly, perform these tasks:

Verifying the IKE Phase 1 Status on SRX1

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command. For a more detailed output, use the **show security ike security-associations detail** command.

```
user@SRX1> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
302301  UP     84e8fc61d0750278  ea9a07ef032805b6  Main           172.16.21.1
```

```
user@SRX1> show security ike security-associations detail
IKE peer 172.16.21.1, Index 302301, Gateway Name: gw1
  Role: Initiator, State: UP
  Initiator cookie: 84e8fc61d0750278, Responder cookie: ea9a07ef032805b6
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 172.16.11.1:4500, Remote: 172.16.21.1:4500
  Lifetime: Expires in 19657 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: srx2@example.com
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
```

```
  Encryption            : 3des-cbc
  Pseudo random function: hmac-sha1
  Diffie-Hellman group  : DH-group-2
 Traffic statistics:
  Input  bytes  :                1780
  Output bytes  :                2352
  Input  packets:                   7
  Output packets:                  14
  Input  fragmentated packets:      0
  Output fragmentated packets:      0
 IPSec security associations: 4 created, 0 deleted
 Phase 2 negotiations in progress: 1

   Negotiation type: Quick mode, Role: Initiator, Message ID: 0
   Local: 172.16.11.1:4500, Remote: 172.16.21.1:4500
   Local identity: srx1@example.com
   Remote identity: srx2@example.com
   Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote address—Verify that the remote IP address is correct and that port 4500 is being used for peer-to-peer communication. Remember that NAT-T encapsulates both IKE and ESP traffic within UDP with port 4500.
- Role initiator state
  - Up—The Phase 1 SA is established.
  - Down—There was a problem establishing the Phase 1 SA.
  - Both peers in the IPsec SA pair are using port 4500.
  - Peer IKE ID—Verify the remote address is correct.
  - Local identity and remote identity—Verify these are correct.

- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations on SRX1

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec security-associations** command. For a more detailed output, use the **show security ipsec security-associations detail** command.

```
user@SRX1> show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
```

```
  <131073 ESP:3des/sha1   fc5dbac4 2160/ unlim  -   root 4500  172.16.21.1
  >131073 ESP:3des/sha1   45fed9d8 2160/ unlim  -   root 4500  172.16.21.1
```

```
user@SRX1> show security ipsec security-associations detail
ID: 131073 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 172.16.11.1, Remote Gateway: 172.16.21.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 4500, Nego#: 7, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Fri Jul 22 2022 11:07:40 -0700: IPSec SA rekey successfully completed (3 times)
    Fri Jul 22 2022 08:38:41 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:41 -0700: User cleared IPSec SA from CLI (1 times)
    Fri Jul 22 2022 08:38:41 -0700: IKE SA negotiation successfully completed (3 times)
    Fri Jul 22 2022 08:38:26 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:26 -0700: User cleared IPSec SA from CLI (1 times)
    Fri Jul 22 2022 08:38:25 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:24 -0700: User cleared IPSec SA from CLI (1 times)
    Fri Jul 22 2022 08:37:37 -0700: IPSec SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: fc5dbac4, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2153 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1532 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 45fed9d8, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2153 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1532 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an address of 172.16.21.1.
- Both peers in the IPsec SA pair are using port 4500.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 2160/ unlim value indicates that the Phase 2 lifetime expires in 2160 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

Verifying the IKE Phase 1 Status on SRX2

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command. For a more detailed output, use the **show security ike security-associations detail** command.

```
user@SRX2> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5567091 UP     84e8fc61d0750278  ea9a07ef032805b6  Main           172.16.11.1
```

```
user@SRX2> show security ike security-associations detail
IKE peer 172.16.11.1, Index 5567091, Gateway Name: gw1
  Role: Responder, State: UP
  Initiator cookie: 84e8fc61d0750278, Responder cookie: ea9a07ef032805b6
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.31.1:4500, Remote: 172.16.11.1:4500
  Lifetime: Expires in 18028 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
```

```
Remote Access Client Info: Unknown Client
Peer ike-id: srx1@example.com
AAA assigned IP: 0.0.0.0
Algorithms:
 Authentication        : hmac-sha1-96
 Encryption            : 3des-cbc
 Pseudo random function: hmac-sha1
 Diffie-Hellman group  : DH-group-2
Traffic statistics:
 Input  bytes  :                 2352
 Output bytes  :                 1780
 Input  packets:                   14
 Output packets:                    7
 Input  fragmentated packets:       0
 Output fragmentated packets:       0
IPSec security associations: 4 created, 3 deleted
Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Responder, Message ID: 0
  Local: 10.1.31.1:4500, Remote: 172.16.11.1:4500
  Local identity: srx2@example.com
  Remote identity: srx1@example.com
  Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations detail` command to get more information about the SA.
- Remote address—Verify that the remote IP address is correct and that port 4500 is being used for peer-to-peer communication.
- Role responder state
  - Up—The Phase 1 SA has been established.
  - Down—There was a problem establishing the Phase 1 SA.

  - Peer IKE ID—Verify the address is correct.
  - Local identity and remote identity—Verify these addresses are correct.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations on SRX2

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec security-associations** command. For a more detailed output, use the **show security ipsec security-associations detail** command.

```
user@SRX2> show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:3des/sha1   45fed9d8 1526/ unlim  -   root 4500  172.16.11.1
  >131073 ESP:3des/sha1   fc5dbac4 1526/ unlim  -   root 4500  172.16.11.1
```

```
user@SRX2> show security ipsec security-associations detail
ID: 131073 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 10.1.31.1, Remote Gateway: 172.16.11.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 4500, Nego#: 25, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Fri Jul 22 2022 11:07:40 -0700: IPSec SA negotiation successfully completed (4 times)
    Fri Jul 22 2022 08:38:41 -0700: Initial-Contact received from peer. Stale IKE/IPSec SAs
cleared (1 times)
    Fri Jul 22 2022 08:38:41 -0700: IKE SA negotiation successfully completed (5 times)
    Fri Jul 22 2022 08:38:26 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:26 -0700: IPSec SA delete payload received from peer, corresponding
IPSec SAs cleared (1 times)
    Fri Jul 22 2022 08:38:25 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:25 -0700: Initial-Contact received from peer. Stale IKE/IPSec SAs
cleared (1 times)
    Fri Jul 22 2022 08:37:37 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:37:37 -0700: IPSec SA delete payload received from peer, corresponding
IPSec SAs cleared (1 times)
    Thu Jul 21 2022 17:57:09 -0700: Peer's IKE-ID validation failed during negotiation (1 times)
    Thu Jul 21 2022 17:49:30 -0700: IKE SA negotiation successfully completed (4 times)
  Direction: inbound, SPI: 45fed9d8, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1461 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 885 seconds
```

```
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: fc5dbac4, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1461 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 885 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an ip address of 172.16.11.1.
- Both peers in the IPsec SA pair are using port 4500.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 1562/ unlim value indicates that the Phase 2 lifetime expires in 1562 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index` *index_id* `detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

Verifying Host-to-Host Reachability

Purpose

Verify Host1 can reach Host2.

Action

From Host1 ping Host2. To verify the traffic is using the VPN, use the command `show security ipsec statistics` on SRX1. Clear the statistics by using the command `clear security ipsec statistics` before running the ping command.

```
user@Host1> ping 10.1.21.2 count 10 rapid
PING 10.1.21.2 (10.1.21.2): 56 data bytes
!!!!!!!!!!
--- 10.1.21.2 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.437/4.270/7.637/1.158 ms
```

```
user@SRX1> show security ipsec statistics
ESP Statistics:
  Encrypted bytes:             1360
  Decrypted bytes:              840
  Encrypted packets:             10
  Decrypted packets:             10
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Meaning

The outputs show Host1 can ping Host2 and that the traffic is using the VPN.

SEE ALSO

IPsec Overview | 12

Example: Configuring NAT-T with Dynamic Endpoint VPN

IN THIS SECTION

- Requirements | 568
- Overview | 568
- Configuration | 570
- Verification | 586

This example shows how to configure a route-based VPN where the IKEv2 initiator is a dynamic endpoint behind a NAT device.

Requirements

This example uses the following hardware and software components:

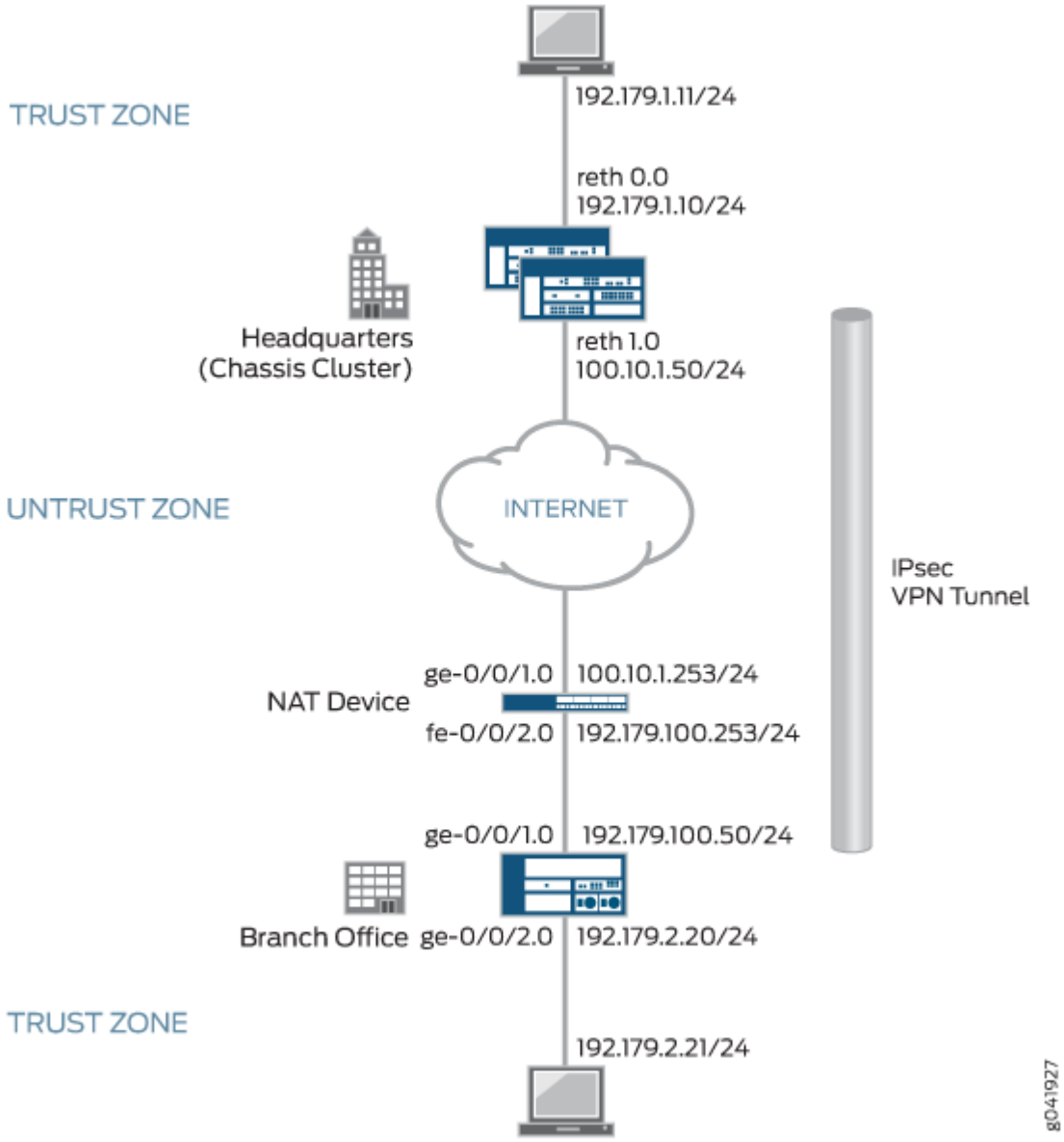
- Two firewalls configured in a chassis cluster
- One firewall providing NAT
- One firewall providing branch office network access
- Junos OS Release that supports IKEv2 NAT-T

Overview

In this example, an IPsec VPN is configured between the branch office (IKEv2 initiator) and headquarters (IKEv2 responder) to secure network traffic between the two locations. The branch office is located behind the NAT device. The branch office address is assigned dynamically and is unknown to the responder. The initiator is configured with the remote identity of the responder for tunnel negotiation. This configuration establishes a dynamic endpoint VPN between the peers across the NAT device.

Figure 38 on page 569 shows an example of a topology with NAT-Traversal (NAT-T) and dynamic endpoint VPN.

Figure 38: NAT-T with Dynamic Endpoint VPN



In this example, the initiator's IP address, 192.179.100.50, which has been dynamically assigned to the device, is hidden by the NAT device and translated to 100.10.1.253.

The following configuration options apply in this example:

- The local identity configured on the initiator must match the remote gateway identity configured on the responder.
- Phase 1 and Phase 2 options must match between the initiator and responder.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Configuration

IN THIS SECTION

- Configuring the Branch Office Device (IKEv2 Initiator) | **570**
- Configuring the NAT Device | **576**
- Configuring the Headquarters Device (IKEv2 Responder) | **579**

Configuring the Branch Office Device (IKEv2 Initiator)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 192.179.100.50/24
set interfaces ge-0/0/2 unit 0 family inet address 192.179.2.20/24
set interfaces st0 unit 0 family inet address 172.168.100.1/16
set routing-options static route 192.179.1.0/24 next-hop st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/2.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway HQ_GW ike-policy IKE_POL
set security ike gateway HQ_GW address 100.10.1.50
```

```
set security ike gateway HQ_GW local-identity hostname branch.example.net
set security ike gateway HQ_GW external-interface ge-0/0/1.0
set security ike gateway HQ_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn HQ_VPN bind-interface st0.0
set security ipsec vpn HQ_VPN ike gateway HQ_GW
set security ipsec vpn HQ_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn HQ_VPN establish-tunnels immediately
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the branch office device:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 192.179.100.50/24
user@host# set ge-0/0/2 unit 0 family inet address 192.179.2.20/24
user@host# set st0 unit 0 family inet address 172.168.100.1/16
```

2. Configure routing options.

```
[edit routing-options]
user@host# set static route 192.179.1.0/24 next-hop st0.0
```

3. Configure zones.

```
[edit security zones security-zones trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/2.0
[edit security zones security-zones untrust]
```

```
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host#set interfaces st0.0
```

4. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set pre-shared-key ascii-text "$ABC123"
 [edit security ike gateway HQ_GW]
user@host# set ike-policy IKE_POL
user@host# set address 100.10.1.50
user@host# set local-identity hostname branch.example.net
user@host# set external-interface ge-0/0/1.0
user@host# set version v2-only
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set proposals IPSEC_PROP
user@host# set perfect-forward-secrecy keys group5
[edit security ipsec vpn HQ_VPN]
user@host# set bind-interface st0.0
user@host# set ike gateway HQ_GW
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

6. Configure the security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, `show security ike`, `show security ipsec`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.179.100.50/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.179.2.20/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 172.168.100.1/16;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 192.179.1.0/24 next-hop st0.0;
}
```

```
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method pre-shared-keys;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    pre-shared-key ascii-text "$ABC123”
}
gateway HQ_GW{
    ike-policy IKE_POL;
```

```
    address 100.10.1.50;
    local-identity hostname branch.example.net;
    external-interface ge-0/0/1.0;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn HQ_VPN {
    bind-interface st0.0;
    ike {
        gateway HQ_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the NAT Device

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 100.10.1.253/24
set interfaces fe-0/0/2 unit 0 family inet address 192.179.100.253/24
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/2.0
set security nat source rule-set DYNAMIC from zone trust
set security nat source rule-set DYNAMIC to zone untrust
set security nat source rule-set DYNAMIC rule R2R3 match source-address 0.0.0.0/0
set security nat source rule-set DYNAMIC rule R2R3 then source-nat interface
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the intermediate router providing NAT:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 100.10.1.253/24
user@host# set fe-0/0/2 unit 0 family inet address 192.179.100.253/24
```

2. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
```

```
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/2.0
```

3. Configure NAT.

```
[edit security nat source rule-set DYNAMIC]
user@host# set from zone trust
user@host# set to zone untrust
user@host# set rule R2R3 match source-address 0.0.0.0/0
user@host# set rule R2R3 then source-nat interface
```

4. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat source`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 100.10.1.253/24;
        }
    }
}
fe-0/0/2 {
    unit 0 {
        family inet {
            address 192.179.100.253/24;
        }
```

```
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/2.0;
    }
}
[edit]
user@host# show security nat source
rule-set DYNAMIC {
    from zone untrust;
    to zone trust;
    rule R2R3 {
        match {
            source-address 0.0.0.0/0;
        }
        then {
            source-nat {
                interface;
            }
```

```
        }
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Headquarters Device (IKEv2 Responder)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set chassis cluster reth-count 5
set chassis cluster redundancy-group 1 node 0 priority 220
set chassis cluster redundancy-group 1 node 1 priority 149
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/2 weight 255
set interfaces ge-0/0/1 gigether-options redundant-parent reth0
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-8/0/1 gigether-options redundant-parent reth0
set interfaces ge-8/0/2 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.179.1.10/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 100.10.1.50/24
set interfaces st0 unit 0 family inet address 172.168.100.2/16
set routing-options static route 192.179.2.0/24 next-hop st0.0
set routing-options static route 192.179.100.0/24 next-hop 100.10.1.253
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone trust host-inbound-traffic system-services all
```

```
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces reth0.0
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway Branch_GW ike-policy IKE_POL
set security ike gateway Branch_GW dynamic hostname branch.example.net
set security ike gateway Branch_GW dead-peer-detection optimized
set security ike gateway Branch_GW external-interface reth1.0
set security ike gateway Branch_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn Branch_VPN bind-interface st0.0
set security ipsec vpn Branch_VPN ike gateway Branch_GW
set security ipsec vpn Branch_VPN ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

1. Configure two nodes as the chassis cluster.

```
[edit chassis cluster]
user@host# set reth-count 5
user@host# set redundancy-group 1 node 0 priority 220
user@host# set redundancy-group 1 node 1 priority 149
user@host# set redundancy-group 1 interface-monitor ge-0/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-0/0/2 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/2 weight 255
```

2. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 gigether-options redundant-parent reth0
user@host# set ge-0/0/2 gigether-options redundant-parent reth1
user@host# set ge-8/0/1 gigether-options redundant-parent reth0
user@host# set ge-8/0/2 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 192.179.1.10/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 100.10.1.50/24
user@host# set st0 unit 0 family inet address 172.168.100.2/16
```

3. Configure routing options.

```
[edit routing-options]
user@host# set static route 192.179.2.0/24 next-hop st0.0
user@host# set static route 192.179.100.0/24 next-hop 100.10.1.253
```

4. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
user@host# set interfaces st0.0
user@host# set interfaces reth1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth0.0
```

5. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
```

```
user@host# set proposals IKE_PROP
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security ike gateway Branch_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic hostname branch.example.net
user@host# set dead-peer-detection optimized
user@host# set external-interface reth1.0
user@host# set version v2-only
```

6. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn Branch_VPN]
user@host# set bind-interface st0.0
user@host# set ike gateway Branch_GW
user@host# set ike ipsec-policy IPSEC_POL
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show chassis cluster`, `show interfaces`, `show routing-options`, `show security zones`, `show security ike`, `show security ipsec`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis cluster
reth-count 5;
redundancy-group 1 {
```

```
    node 0 priority 220;
    node 1 priority 149;
    interface-monitor {
        ge-0/0/1 weight 255;
        ge-8/0/1 weight 255;
        ge-0/0/2 weight 255;
        ge-8/0/2 weight 255;
    }
}
[edit]
user@host# show interfaces
ge-0/0/1 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-0/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-8/0/1 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-8/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.179.1.10/24;
        }
    }
}
reth1 {
    redundant-ether-options {
```

```
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 100.10.1.50/24;
        }
    }
}
st0 {
    unit 0{
        family inet {
            address 172.168.100.2/16;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 192.179.2.0/24 next-hop st0.0;
    route 192.179.100.0/24 next-hop 100.10.1.253;
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
```

```
        }
    }
    interfaces {
        st0.0;
        reth1.0;
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method pre-shared-keys;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    pre-shared-key ascii-text “$ABC123”
}
gateway Branch_GW {
    ike-policy IKE_POL;

    dynamic hostname branch.example.net;
    dead-peer-detection optimized;
    external-interface reth1.0;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn Branch_VPN {
    bind-interface st0.0;
    ike {
```

```
        gateway Branch_GW;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

Verification

IN THIS SECTION

- Verifying the IKE Phase 1 Status for the Responder | 586
- Verifying IPsec Security Associations for the Responder | 588

Confirm that the configuration is working properly.

Verifying the IKE Phase 1 Status for the Responder

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode on node 0, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations detail** command.

```
user@host# show security ike security-associations
node0:
Index      State  Initiator cookie   Responder cookie  Mode    Remote Address
1367024684 UP     f82c54347e2f3fb1   020e28e1e4cae003  IKEv2    100.10.1.253
```

```
user@host# show security ike security-associations detail
node0:
IKE peer 100.10.1.253, Index 1367024684, Gateway Name: Branch_GW
  Location: FPC 5, PIC 0, KMD-Instance 2
  Role: Responder, State: UP
  Initiator cookie: f82c54347e2f3fb1, Responder cookie: 020e28e1e4cae003
  Exchange type: IKEv2, Authentication method: Pre-shared-keys
  Local: 100.10.1.50:4500, Remote: 100.10.1.253:2541
  Lifetime: Expires in 3593 seconds
  Peer ike-id: branch.example.net
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                  683
   Output bytes  :                  400
   Input  packets:                    2
   Output packets:                    1
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index` *`index_id`* `detail` command to get more information about the SA.
- Remote address—Verify that the local IP address is correct and that port 4500 is being used for peer-to-peer communication.
- Role responder state
  - Up—The Phase 1 SA has been established.

  - Down—There was a problem establishing the Phase 1 SA.
  - Peer IKE ID—Verify the address is correct.
  - Local identity and remote identity—Verify these addresses are correct.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that sends IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations for the Responder

Purpose

Verify the IPsec status.

Action

From operational mode on node 0, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations detail** command.

```
user@host# show security ipsec security-associations
node0
  Total active tunnels: 1
  ID        Algorithm            SPI      Life:sec/kb  Mon lsys Port Gateway
  <77856771 ESP:aes-cbc-256/sha1 4ad5af40 7186/unlim   - root   2541 100.10.1.253
  >77856771 ESP:aes-cbc-256/sha1 5bb0a5ee 7186/unlim   - root   2541 100.10.1.253
```

```
user@host# show security ipsec security-associations detail
node0
  ID: 77856771 Virtual-system: root, VPN Name: Branch_VPN
  Local Gateway: 100.10.1.50, Remote Gateway: 100.10.1.253
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
    DF-bit: clear
    Bind-interface: st0.0

  Port: 2541, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 608a29
  Tunnel Down Reason: SA not initiated
    Location: FPC 5, PIC 0, KMD-Instance 2
    Direction: inbound, SPI: 4ad5af40, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 7182 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 6587 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an IP address of 100.10.1.253.

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The lifetime value indicates that the Phase 2 lifetime expires in 7186 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index index_id detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, match for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

SEE ALSO

> IPsec Overview | **12**
>
> *Security Policies Overview*

Platform-Specific NAT-T with IPsec VPN Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platforms.

Table 89: Platform-Specific Behavior

Platform	Difference

SRX Series	• On SRX5400, SRX5600, and SRX5800 devices that support NAT-T with IPsec VPNs, note the following tunnel scaling limitations:
	  • For a given private IP address, the NAT device must translate both 500 and 4500 private ports to the same public IP address.
	  • The total number of tunnels from a given public translated IP cannot exceed 1000 tunnels.
	• On SRX5600 and SRX5800 devices that support NAT-T with dynamic endpoint VPN:
	  • IKE negotiations with NAT-T do not work when the IKE peer is behind the NAT device that changes the source IP address of the IKE packet during the negotiation. For example, if you configure the NAT device with dynamic IP, the NAT device changes the source IP because the IKE protocol switches the UDP port from 500 to 5400.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.2R1	When the NAT-T remote port changes, the peer device might create a new session during incoming Dead Peer Detection (DPD), causing session mismatches and interrupting traffic. To prevent this, you can enable optimized dead peer detection. Use the command set security ike gateway gateway-name dead-peer-detection optimized to update the existing NAT-T session with the new port number, allowing traffic to resume.

RELATED DOCUMENTATION

Traffic Selectors in Route-Based VPNs | 451

7

CHAPTER

Group VPN

IN THIS CHAPTER

- Group VPNv1 | **593**
- Group VPNv2 | **655**
- Group VPNv2 Server Clusters | **715**

Group VPNv1

SUMMARY

Read this topic to learn about Group VPNv1 in Junos OS.

IN THIS SECTION

- Group VPNv1 Overview | **594**
- Group VPNv1 Configuration Overview | **603**
- Understanding IKE Phase 1 Configuration for Group VPNv1 | **604**
- Understanding IPsec SA Configuration for Group VPNv1 | **604**
- Understanding Dynamic Policies for Group VPNv1 | **605**
- Understanding Antireplay for Group VPNv1 | **606**
- Example: Configuring Group VPNv1 Server and Members | **607**
- Example: Configuring Group VPNv1 Server-Member Communication for Unicast Rekey Messages | **636**
- Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages | **638**
- Example: Configuring Group VPNv1 with Server-Member Colocation | **641**

Group VPN is a set of features that are necessary to secure IP multicast group traffic or unicast traffic over a private WAN that originates on or flows through a device.

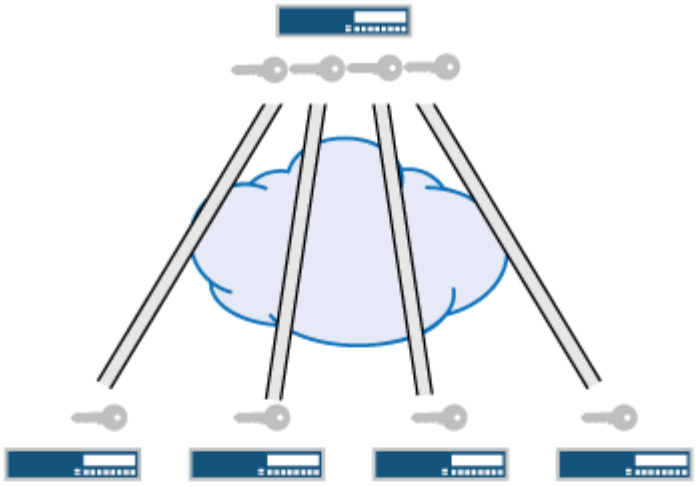
Group VPNv1 Overview

IN THIS SECTION

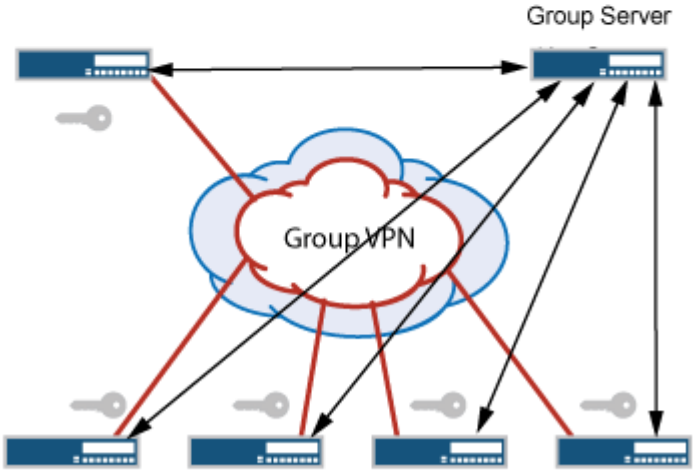
- Understanding the GDOI Protocol for Group VPNv1 | **596**
- Understanding Group VPNv1 Limitations | **596**
- Understanding Group VPNv1 Servers and Members | **597**
- Understanding Group VPNv1 Server-Member Communication | **598**
- Understanding Group VPNv1 Group Key Operations | **599**
- Understanding Group VPNv1 Heartbeat Messages | **602**
- Understanding Group VPNv1 Server-Member Colocation Mode | **602**

An IPsec security association (SA) is a unidirectional agreement between virtual private network (VPN) participants that defines the rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications. With current VPN implementations, the SA is a point-to-point tunnel between two security devices. Group VPNv1 extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 39 on page 595).

Figure 39: Standard IPsec VPN and Group VPNv1



Standard IPsec VPN



Group VPN

Server distributes IPsec SA. All members that belong to the group share the same IPsec SA.

With Group VPNv1, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header. Secure multicast packets are replicated in the same way as cleartext multicast packets in the core network.

Group VPNv1 has some propriety limitations regarding RFC 6407, *The Group Domain of Interpretation (GDOI)*. To use Group VPN without proprietary limitations, upgrade to Group VPNv2.

Understanding the GDOI Protocol for Group VPNv1

Group VPNv1 is based on RFC 3547, *The Group Domain of Interpretation* (GDOI). This RFC describes the protocol between group members and a group server to establish SAs among group members. GDOI messages create, maintain, or delete SAs for a group of devices. The GDOI protocol runs on port 848.

The Internet Security Association and Key Management Protocol (ISAKMP) defines two negotiation phases to establish SAs for an AutoKey IKE IPsec tunnel. Phase 1 allows two devices to establish an ISAKMP SA. Phase 2 establishes SAs for other security protocols, such as GDOI.

With group VPN, Phase 1 ISAKMP SA negotiation is performed between a group server and a group member. The server and member must use the same ISAKMP policy. In Phase 2, GDOI exchanges between the server and member establish the SAs that are shared with other group members. A group member does not need to negotiate IPsec with other group members. GDOI exchanges in Phase 2 must be protected by ISAKMP Phase 1 SAs.

There are two types of GDOI exchanges:

- The `groupkey-pull` exchange allows a member to request SAs and keys shared by the group from the server.
- The `groupkey-push` exchange is a single rekey message that allows the server to send group SAs and keys to members before existing group SAs expire. Rekey messages are unsolicited messages sent from the server to members.

Understanding Group VPNv1 Limitations

The following are not supported in this release for group VPNv1:

- Non-default routing instances
- *Chassis cluster*
- Server clusters
- Route-based group VPN
- Public Internet-based deployment
- SNMP
- Deny policy from Cisco GET VPN server
- J-Web interface for configuration and monitoring

When you configure Group VPNv1 members for use with Group VPNv2 servers, note the following limitations:

- Group VPNv2 supports the IETF draft specification *IP Delivery Delay Detection Protocol* for a time-based antireplay mechanism. Therefore, IP delivery delay detection protocol-based antireplay is not supported on Group VPNv1 members and must be disabled on the Group VPNv2 server with the `deactivate security group-vpn server group group-name anti-replay-time-window` command.
- The Group VPNv2 server does not support colocation, where the group server and group member functions exist in the same device.
- The Group VPNv2 server does not support heartbeat transmittals. Heartbeat must be disabled on the Group VPNv1 member with the `deactivate security group-vpn member ipsec vpn vpn-name heartbeat-threshold` command. We recommend using Group VPNv2 server clusters to avoid traffic impact due to reboots or other interruptions on the Group VPNv2 server.
- Groupkey-push messages sent from the Group VPNv2 server are based on RFC 6407, *The Group Domain of Interpretation (GDOI)* and are not supported on Group VPNv1 members. Therefore, groupkey-push messages must be disabled on the Group VPNv2 server with the `deactivate security group-vpn server group group-name server-member-communication` command.

  Rekeys are supported with groupkey-pull messages. If there are scaling issues where Group VPNv1 members cannot complete the groupkey-pull operation before the TEK hard lifetime expires, we recommend increasing the TEK lifetime to allow sufficient time for members to complete the groupkey-pull operation. Juniper's scaling numbers are qualified with a 2 hour TEK lifetime.
- If the Group VPNv2 server is rebooted or upgraded, or the SAs for the group are cleared, new members cannot be added to the network until the next rekey occurs for existing members. New members cannot send traffic to existing members that have old keys. As a workaround, clear the SAs on the existing Group VPNv1 members with the `clear security group-vpn member ipsec security-associations` command.
- Because multicast data traffic is not supported by Group VPNv2 members, multicast data traffic cannot be used when Group VPNv1 and Group VPNv2 members coexist in the network for the same group.

Understanding Group VPNv1 Servers and Members

The center of a group VPN is the group server. The group server performs the following tasks:

- Controls group membership
- Generates encryption keys
- Manages group SAs and keys and distributes them to group members

Group members encrypt traffic based on the group SAs and keys provided by the group server.

A group server can service multiple groups. A single security device can be a member of multiple groups.

Each group is represented by a group identifier, which is a number between 1 and 65,535. The group server and group members are linked together by the group identifier. There can be only one group identifier per group, and multiple groups cannot use the same group identifier.

The following is a high-level view of group VPN server and member actions:

1. The group server listens on UDP port 848 for members to register. A member device must provide correct IKE Phase 1 authentication to join the group. Preshared key authentication on a per-member basis is supported.
2. Upon successful authentication and registration, the member device retrieves group SAs and keys from the server with a GDOI `groupkey-pull` exchange.
3. The server adds the member to the membership for the group.
4. Group members exchange packets encrypted with group SA keys.

The server periodically sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or when the group SA has changed.

Understanding Group VPNv1 Server-Member Communication

Server-member communication allows the server to send GDOI `groupkey-push` messages to members. If server-member communication is not configured for the group, members can send GDOI `groupkey-pull` messages to register and reregister with the server, but the server is not able to send rekey messages to members.

Server-member communication is configured for the group by using the `server-member-communication` *configuration statement* at the [`edit security group-vpn server`] hierarchy. The following options can be defined:

- Encryption algorithm used for communications between the server and member. You can specify 3des-cbc, aes-128-cbc, aes-192-cbc, aes-256-cbc, or des-cbc. There is no default algorithm.
- Authentication algorithm (md5 or sha1) used to authenticate the member to the server. There is no default algorithm.
- Whether the server sends unicast or multicast rekey messages to group members and parameters related to the communication type.

- Interval at which the server sends heartbeat messages to the group member. This allows the member to determine whether the server has rebooted, which would require the member to reregister with the server. The default is 300 seconds.
- Lifetime for the key encryption key (KEK). The default is 3600 seconds.

Configuring server-member communication is necessary for the group server to send rekey messages to members, but there might be situations in which this behavior is not desired. For example, if group members are dynamic peers (such as in a home office), the devices are not always up and the IP address of a device might be different each time it is powered up. Configuring server-member communication for a group of dynamic peers can result in unnecessary transmissions by the server. If you want IKE Phase 1 SA negotiation to always be performed to protect GDOI negotiation, do not configure server-member communication.

If server-member communication for a group is not configured, the membership list displayed by the `show security group-vpn server registered-members` command shows group members who have registered with the server; members can be active or not. When server-member communication for a group is configured, the group membership list is cleared. If the communication type is configured as unicast, the `show security group-vpn server registered-members` command shows only active members. If the communication type is configured as multicast, the `show security group-vpn server registered-members` command shows members who have registered with the server after the configuration; the membership list does not necessarily represent active members because members might drop out after registration.

Understanding Group VPNv1 Group Key Operations

This topic contains the following sections:

Group Keys

The group server maintains a database to track the relationship among VPN groups, group members, and group keys. There are two kinds of group keys that the server downloads to members:

- Key Encryption Key (KEK)—Used to encrypt rekey messages. One KEK is supported per group.
- Traffic Encryption Key (TEK)—Used to encrypt and decrypt IPsec data traffic between group members.

The key associated with an SA is accepted by a group member only if there is a matching scope policy configured on the member. An accepted key is installed for the group VPN, whereas a rejected key is discarded.

Rekey Messages

If the group is configured for server-member communications, the server periodically sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or the group SA has changed (for example, a group policy is added or deleted).

Server-member communications options must be configured on the server to allow the server to send rekey messages to group members. These options specify the type of message and the intervals at which the messages are sent, as explained in the following sections:

There are two types of rekey messages:

- Unicast rekey messages—The group server sends one copy of the rekey message to each group member. Upon receipt of the rekey message, members must send an acknowledgment (ACK) to the server. If the server does not receive an ACK from a member (including retransmission of rekey messages), the server considers the member to be inactive and removes it from the membership list. The server stops sending rekey messages to the member.

  The `number-of-retransmission` and `retransmission-period` configuration statements for server-member communications control the resending of rekey messages by the server when no ACK is received from a member.

- Multicast rekey messages—The group server sends one copy of the rekey message from the specified outgoing interface to the configured multicast group address. Members do not send acknowledgment of receipt of multicast rekey messages. The registered membership list does not necessarily represent active members because members might drop out after initial registration. All members of the group must be configured to support multicast messages.

  IP multicast protocols must be configured to allow delivery of multicast traffic in the network. For detailed information about configuring multicast protocols on Juniper Networks devices, see Multicast Protocols User Guide .

The interval at which the server sends rekey messages is calculated based on the values of the `lifetime-seconds` and `activation-time-delay` configuration statements at the [`edit security group-vpn server group`] hierarchy. The interval is calculated as `lifetime-seconds` minus 4*(`activation-time-delay`).

The `lifetime-seconds` for the KEK is configured as part of the server-member communications; the default is 3600 seconds. The `lifetime-seconds` for the TEK is configured for the IPsec proposal; the default is 3600 seconds. The `activation-time-delay` is configured for the group on the server; the default is 15 seconds. Using the default values for `lifetime-seconds` and `activation-time-delay`, the interval at which the server sends rekey messages is 3600 minus 4*15, or 3540 seconds.

Member Registration

If a group member does not receive a new SA key from the server before the current key expires, the member must reregister with the server and obtain updated keys with a GDOI `groupkey-pull` exchange. In this case, the interval at which the server sends rekey messages is calculated as follows: `lifetime-seconds` minus 3*(`activation-time-delay`). Using the default values for `lifetime-seconds` and `activation-time-delay`, the interval at which the server sends rekey messages is 3600 minus 3*15, or 3555 seconds.

Member reregistration can occur for the following reasons:

- The member detects a server reboot by the absence of heartbeats received from the server.
- The rekey message from the group server is lost or delayed, and the TEK lifetime has expired.

Key Activation

When a member receives a new key from the server, it waits a period of time before using the key for encryption. This period of time is determined by the `activation-time-delay` *configuration statement* and whether the key is received through a rekey message sent from the server or as a result of the member reregistering with the server.

If the key is received through a rekey message sent from the server, the member waits 2*(`activation-time-delay`) seconds before using the key. If the key is received through member reregistration, the member waits the number of seconds specified by the `activation-time-delay` value.

A member retains the two most recent keys sent from the server for each group SA installed on the member. Both keys can be used for decryption, while the most recent key is used for encryption. The previous key is removed the number of seconds specified by the `activation-time-delay` value after the new key is activated.

The default for the `activation-time-delay` configuration statement is 15 seconds. Setting this time period too small can result in a packet being dropped at a remote group member before the new key is installed. Consider the network topology and system transport delays when you change the `activation-time-delay` value. For unicast transmissions, the system transport delay is proportional to the number of group members.

A group VPNv1 server can send multiple traffic encryption keys (TEKs) to a group VPNv1 member in response to a `groupkey-pull` request. The following describes how the group VPNv1 member handles the existing TEK and the TEKs it receives from the server:

- If the group VPNv1 member receives two or more TEKs, it holds the most recent two TEKs and deletes the existing TEK. Of the two held TEKs, the older TEK is activated immediately, and the newer TEK is activated after the `activation-time-delay` configured on the group VPNv1 server has elapsed (the default is 15 seconds).

- If the group VPNv1 member receives only one TEK, or if it receives a TEK through a `groupkey-push` message from the server, the existing TEK is not deleted until the hard lifetime expires. The lifetime is not shortened for the existing TEK.

The group VPNv1 member still installs a received TEK even if the TEK lifetime is less than two times the `activation-time-delay` value.

Understanding Group VPNv1 Heartbeat Messages

When server-member communication is configured, the group VPNv1 server sends heartbeat messages to members at specified intervals (the default interval is 300 seconds). The heartbeat mechanism allows members to reregister with the server if the specified number of heartbeats is not received. For example, members will not receive heartbeat messages during a server reboot. When the server has rebooted, members reregister with the server.

Heartbeats are transmitted through `groupkey-push` messages. The sequence number is incremented on each heartbeat message, which protects members from reply attacks. Unlike rekey messages, heartbeat messages are not acknowledged by recipients and are not retransmitted by the server.

Heartbeat messages contain the following information:

- Current state and configuration of the keys on the server
- Relative time, if antireplay is enabled

By comparing the information in the heartbeats, a member can detect whether it has missed server information or rekey messages. The member reregisters to synchronize itself with the server.

Heartbeat messages can increase network congestion and cause unnecessary member reregistrations. Thus, heartbeat detection can be disabled on the member if necessary.

Understanding Group VPNv1 Server-Member Colocation Mode

Group server and group member functions are separate and do not overlap. The server and member functions can coexist in the same physical device, which is referred as colocation mode. In colocation mode, there is no change in terms of functionality and behavior of the server or a member, but the server and member each need to be assigned different IP addresses so that packets can be delivered properly. In colocation mode, there can be only one IP address assigned to the server and one IP address assigned to the member across groups.

SEE ALSO

IPsec Overview | 12

Understanding IKE and IPsec Packet Processing | 81

Group VPNv1 Configuration Overview

This topic describes the main tasks for configuring group VPNv1.

On the group server, configure the following:

1. IKE Phase 1 negotiation. Use the [`edit security group-vpn server ike`] hierarchy to configure the IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 662.
2. Phase 2 IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv1" on page 604.
3. VPN group. See "Group VPNv1 Configuration Overview" on page 603.

On the group member, configure the following:

1. IKE Phase 1 negotiation. Use the [`edit security group-vpn member ike`] hierarchy to configure IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv1 " on page 604.
2. Phase 2 IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv1" on page 604.
3. Scope policy that determines which group policies are installed on the member. See "Understanding Dynamic Policies for Group VPNv1" on page 605.

To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for a maximum transmission unit (MTU) size no larger than 1400 bytes. Use the `set` *`interface`* `mtu` configuration statement to set the MTU size.

The VPN group is configured on the server with the `group` *configuration statement* at the [`edit security group-vpn server`] hierarchy.

The group information consists of the following information:

- Group identifier—A value between 1 and 65,535 that identifies the VPN group. The same group identifier must be configured on the group member for Autokey IKE.
- Group members, as configured with the `ike-gateway` configuration statement. There can be multiple instances of this configuration statement, one for each member of the group.
- IP address of the server (the loopback interface address is recommended).
- Group policies—Policies that are to be downloaded to members. Group policies describe the traffic to which the SA and keys apply. See "Understanding Dynamic Policies for Group VPNv1" on page 605.
- Server-member communication—Optional configuration that allows the server to send rekey messages to members. See "Group VPNv1 Overview" on page 594.

- Antireplay—Optional configuration that detects packet interception and replay. See "Understanding Antireplay for Group VPNv1" on page 606.

Understanding IKE Phase 1 Configuration for Group VPNv1

An IKE Phase 1 SA between the group server and a group member establishes a secure channel in which to negotiate IPsec SAs that are shared by a group. For standard IPsec VPNs on Juniper Networks security devices, Phase 1 SA configuration consists of specifying an IKE proposal, policy, and gateway. For group VPNv1, the IKE Phase 1 SA configuration is similar to the configuration for standard IPsec VPNs, but is performed at the [`edit security group-vpn`] hierarchy.

In the IKE proposal configuration, you set the authentication method and the authentication and encryption algorithms that will be used to open a secure channel between participants. In the IKE policy configuration, you set the mode (main or aggressive) in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal. In the IKE gateway configuration, you reference the Phase 1 policy.

Because Group VPNv2 only supports strong algorithms, the `sha-256` authentication algorithm option is supported for Group VPNv1 members. When Group VPNv1 members interoperate with Group VPNv2 servers, this option must be configured on the Group VPNv1 members with the `edit security group-vpn member ike proposal proposal-name authentication-algorithm sha-256` command. On the Group VPNv2 server, `authentication-algorithm sha-256` must be configured for IKE proposals and `authentication-algorithm hmac-sha-256-128` must be configured for IPsec proposals.

If an IKE gateway on a Group VPNv1 member is configured with more than one gateway address, the error message “Only one remote address is allowed to be configured per IKE gateway configuration” is displayed when the configuration is committed.

The IKE Phase 1 configuration on the group server must match the IKE Phase 1 configuration on group members.

Understanding IPsec SA Configuration for Group VPNv1

After the server and member have established a secure and authenticated channel in Phase 1 negotiation, they proceed through Phase 2. Phase 2 negotiation establishes the IPsec SAs that are shared by group members to secure data that is transmitted among members. While the IPsec SA configuration for group VPN is similar to the configuration for standard VPNs, a group member does not need to negotiate the SA with other group members.

Phase 2 IPsec configuration for group VPNv1 consists of the following information:

- A proposal for the security protocol, authentication, and encryption algorithm to be used for the SA. The IPsec SA proposal is configured on the group server with the `proposal` *configuration statement* at the [`edit security group-vpn server ipsec`] hierarchy.
- A group policy that references the proposal. A group policy specifies the traffic (protocol, source address, source port, destination address, and destination port) to which the SA and keys apply. The group policy is configured on the server with the `ipsec-sa` configuration statement at the [`edit security group-vpn server group`] hierarchy.
- An Autokey IKE that references the group identifier, the group server (configured with the `ike-gateway` configuration statement), and the interface used by the member to connect to the group. The Autokey IKE is configured on the member with the `ipsec vpn` configuration statement at the [`edit security group-vpn member`] hierarchy.

Understanding Dynamic Policies for Group VPNv1

The group server distributes group SAs and keys to members of a specified group. All members that belong to the same group can share the same set of IPsec SAs. But not all SAs configured for a group are installed on every group member. The SA installed on a specific member is determined by the policy associated with the group SA and the security policies configured on the member.

In a VPN group, each group SA and key that the server pushes to a member is associated with a group policy. The group policy describes the traffic on which the key should be used, including protocol, source address, source port, destination address, and destination port.

Group policies that are identical (configured with the same source address, destination address, source port, destination port, and protocol values) cannot exist for a single group. An error is returned if you attempt to commit a configuration that contains identical group policies for a group. If this is the case, you must delete one of the identical group policies.

On a group member, a scope policy must be configured that defines the scope of the group policy downloaded from the server. A group policy distributed from the server is compared against the scope policies configured on the member. For a group policy to be installed on the member, the following conditions must be met:

- Any addresses specified in the group policy must be within the range of addresses specified in the scope policy.
- The source port, destination port, and protocol specified in the group policy must match those configured in the scope policy.

A group policy that is installed on a member is called a dynamic policy.

A scope policy can be part of an ordered list of security policies for a specific from-zone and to-zone context. Junos OS performs a security policy lookup on incoming packets starting from the top of the ordered list.

Depending on the position of the scope policy within the ordered list of security policies, there are several possibilities for dynamic policy lookup:

- If the incoming packet matches a security policy before the scope policy is considered, dynamic policy lookup does not occur.
- If an incoming policy matches a scope policy, the search process continues for a matching dynamic policy. If there is a matching dynamic policy, that policy action (permit) is performed. If there is no matching dynamic policy, the search process continues to search the policies below the scope policy.

  In this release, only the `tunnel` action is allowed for a scope policy. Other actions are not supported.

You configure a scope policy on a group member by using the `policies` *configuration statement* at the [`edit security`] hierarchy. Use the `ipsec-group-vpn` configuration statement in the permit tunnel rule to reference the group VPN; this allows group members to share a single SA.

SEE ALSO

> *Security Policies Overview*
>
> *Understanding Security Policy Ordering*
>
> *Example: Configuring a Security Policy to Permit or Deny All Traffic*

Understanding Antireplay for Group VPNv1

Antireplay is an IPsec feature that can detect when a packet is intercepted and then replayed by attackers. Antireplay is enabled by default for group VPNs but can be disabled for a group with the `no-anti-replay` *configuration statement*.

When antireplay is enabled, the group server synchronizes the time between the group members. Each IPsec packet contains a timestamp. The group member checks whether the packet's timestamp falls within the configured `anti-replay-time-window` value (the default is 100 seconds). A packet is dropped if the timestamp exceeds the value.

SEE ALSO

> IPsec Overview | 12

Example: Configuring Group VPNv1 Server and Members

IN THIS SECTION

- Requirements | 607
- Overview | 607
- Configuration | 608
- Verification | 633

This example shows how to configure group VPNv1 to extend IPsec architecture to support SAs that are shared by a group of security devices.

Requirements

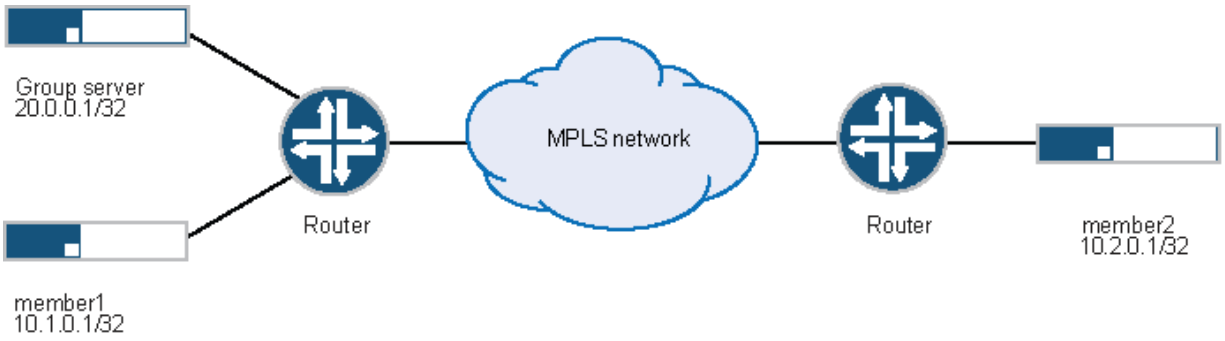
Before you begin:

- Configure the Juniper Networks security devices for network communication.
- Configure network interfaces on server and member devices. See Interfaces User Guide for Security Devices.

Overview

In Figure 40 on page 608, a group VPN consists of two member devices (member1 and member2) and a group server (the IP address of the loopback interface on the server is 20.0.0.1). The group identifier is 1.

Figure 40: Server-Member Configuration Example



The Phase 2 group VPN SAs must be protected by a Phase 1 SA. Therefore, the group VPN configuration must include configuring IKE Phase 1 negotiations on both the group server and the group members. In addition, the same group identifier must be configured on both the group server and the group members.

Group policies are configured on the group server. All group policies configured for a group are downloaded to group members. Scope policies configured on a group member determine which group policies are actually installed on the member. In this example, the following group policies are configured on the group server for downloading to all group members:

- p1—Allows all traffic from 10.1.0.0/16 to 10.2.0.0./16
- p2—Allows all traffic from 10.2.0.0./16 to 10.1.0.0/16
- p3—Allows multicast traffic from 10.1.1.1/32

The member1 device is configured with scope policies that allow all unicast traffic to and from the 10.0.0.0/8 subnetwork. There is no scope policy configured on member1 to allow multicast traffic; therefore, the SA policy p3 is not installed on member1.

The member2 device is configured with scope policies that drop traffic from 10.1.0.0/16 from the trust zone to the untrust zone and to 10.1.0.0/16 from the untrust zone to the trust zone. Therefore the SA policy p2 is not installed on member2.

Configuration

IN THIS SECTION

- Configuring the Group Server | **609**
- Configuring Member1 | **616**
- Configuring Member2 | **623**

Configuring the Group Server

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces lo0 unit 0 family inet address 20.0.0.1/32

set security group-vpn server ike proposal srv-prop authentication-method pre-
shared-keys

set security group-vpn server ike proposal srv-prop dh-group group2

set security group-vpn server ike proposal srv-prop authentication-algorithm
sha1

set security group-vpn server ike proposal srv-prop encryption-algorithm 3des-
cbc

set security group-vpn server ike policy srv-pol mode main

set security group-vpn server ike policy srv-pol proposals srv-prop

set security group-vpn server ike policy srv-pol pre-shared-key ascii-text
"$ABC123"

set security group-vpn server ike gateway gw1 ike-policy srv-pol

set security group-vpn server ike gateway gw1 address 10.1.0.1
```

```
                set security group-vpn server ike gateway gw2 ike-policy srv-pol

                set security group-vpn server ike gateway gw2 address 10.2.0.1

                set security group-vpn server ipsec proposal group-prop authentication-
algorithm hmac-sha1-96

                set security group-vpn server ipsec proposal group-prop encryption-algorithm
3des-cbc

                set security group-vpn server ipsec proposal group-prop lifetime-seconds 3600

                set security group-vpn server group grp1 group-id 1

                set security group-vpn server group grp1 ike-gateway gw1

                set security group-vpn server group grp1 ike-gateway gw2

                set security group-vpn server group grp1 anti-replay-time-window 120

                set security group-vpn server group grp1 server-address 20.0.0.1

                set security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop

                set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1
source 10.1.0.0/16

                set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1
destination 10.2.0.0/16
```

```
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1
source-port 0

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1
destination-port 0

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1
protocol 0

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2
source 10.2.0.0/16

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2
destination 10.1.0.0/16

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2
source-port 0

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2
destination-port 0

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2
protocol 0

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3
source 10.1.1.1/16

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3
destination 239.1.1.1/32

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3
```

```
source-port 0


                set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3
destination-port 0


                set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3
protocol 0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the group server:

1. Configure the loopback address on the device.

```
[edit]
user@host# edit interfaces
user@host# set lo0 unit 0 family inet address 20.0.0.1/32
```

2. Configure IKE Phase 1 SA (this configuration must match the Phase 1 SA configured on the group members).

```
[edit security group-vpn server ike proposal srv-prop]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm 3des-cbc
```

3. Define the IKE policy and set the remote gateways.

```
[edit security group-vpn server ike]
user@host# set policy srv-pol mode main proposals srv-prop pre-shared-key ascii-text "$ABC123"
user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1
```

4. Configure the Phase 2 SA exchange.

```
[edit security group-vpn server ipsec proposal group-prop]
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 3600
```

5. Configure the group identifier and IKE gateway.

```
[edit security group-vpn server group grp1]
user@host# set group-id 1
user@host# set ike-gateway gw1
user@host# set ike-gateway gw2
user@host# set anti-replay-time-window 120 server-address 20.0.0.1
```

6. Configure server-to-member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast encryption-algorithm
aes-128-cbc sig-hash-algorithm md5 certificate “srv-cert”
```

7. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa]
user@host# set proposal group-prop match-policy p1 source 10.1.0.0/16 destination 10.2.0.0/16
source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p2 source 10.2.0.0/16 destination 10.1.0.0/16
source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p3 source 10.1.1.1/16 destination
239.1.1.1/32 source-port 0 destination-port 0 protocol 0
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn server` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server
ike {
    proposal srv-prop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
    }
    policy srv-pol {
        mode main;
        proposals srv-prop;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway gw1 {
        ike-policy srv-pol;
        address 10.1.0.1;
    }
```

```
    gateway gw2 {
        ike-policy srv-pol;
        address 10.2.0.1;
    }
}
    ipsec {
        proposal group-prop {
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
    }
    group grp1 {
        group-id 1;
        ike-gateway gw1;
        ike-gateway gw2;
        anti-replay-time-window 120;
        server-address 20.0.0.1;
        ipsec-sa group-sa {
            proposal group-prop;
            match-policy p1 {
                source 10.1.0.0/16;
                destination 10.2.0.0/16;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
            match-policy p2 {
                source 10.2.0.0/16;
                destination 10.1.0.0/16;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
            match-policy p3 {
                source 10.1.1.1/16;
                destination 239.1.1.1/32;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
```

```
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Member1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security group-vpn member ike proposal prop1 authentication-method pre-
shared-keys

set security group-vpn member ike proposal prop1 dh-group group2

set security group-vpn member ike proposal prop1 authentication-algorithm sha1

set security group-vpn member ike proposal prop1 encryption-algorithm 3des-cbc

set security group-vpn member ike policy pol1 mode main

set security group-vpn member ike policy pol1 proposals prop1

set security group-vpn member ike policy pol1 pre-shared-key ascii-text
"$ABC123"

set security group-vpn member ike gateway g1 ike-policy pol1

set security group-vpn member ike gateway g1 address 20.0.0.1
```

```
                set security group-vpn member ike gateway g1 local-address 10.1.0.1

                set security group-vpn member ipsec vpn v1 ike-gateway g1

                set security group-vpn member ipsec vpn v1 group-vpn-external-interface
ge-0/1/0

                set security group-vpn member ipsec vpn v1 group 1

                set security address-book book1 address 10_subnet 10.0.0.0/8

                set security address-book book1 attach zone trust

                set security address-book book2 address 10_subnet 10.0.0.0/8

                set security address-book book2 attach zone untrust

                set security policies from-zone trust to-zone untrust policy scope1 match
source-address 10_subnet

                set security policies from-zone trust to-zone untrust policy scope1 match
destination-address 10_subnet

                set security policies from-zone trust to-zone untrust policy scope1 match
application any

                set security policies from-zone trust to-zone untrust policy scope1 then
permit tunnel ipsec-group-vpn v1
```

```
                set security policies from-zone untrust to-zone trust policy scope1 match
source-address 10_subnet


                set security policies from-zone untrust to-zone trust policy scope1 match
destination-address 10_subnet


                set security policies from-zone untrust to-zone trust policy scope1 match
application any


                set security policies from-zone untrust to-zone trust policy scope1 then
permit tunnel ipsec-group-vpn v1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure member1:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

```
[edit security group-vpn member ike proposal prop1]
user@member1# set authentication-method pre-shared-keys
user@member1# set dh-group group2
user@member1# set authentication-algorithm sha1
user@member1# set encryption-algorithm 3des-cbc
```

2. Define the IKE policy and set the remote gateways.

```
[edit security group-vpn member ike]
user@member1# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text "$ABC123"
user@member1# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address 10.1.0.1
```

3. Configure the group identifier, IKE gateway, and interface for member1.

```
[edit security group-vpn member ipsec]
user@member1# set vpn v1 group 1 ike-gateway g1 group-vpn-external-interface ge-0/1/0
```

To prevent packet fragmentation issues, we recommend that the interface used by the group members to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set` *interface* `mtu` configuration statement to set the MTU size.

4. Create address books and attach zones to them.

```
[edit security address-book book1]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone trust
```

```
[edit security address-book book2]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone untrust
```

5. Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address 10_subnet destination-address 10_subnet
application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

6. Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address 10_subnet destination-address 10_subnet
application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn member` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@member1# show security group-vpn member
ike {
    proposal prop1 {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
    }
    policy pol1 {
        mode main;
        proposals prop1;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway g1 {
        ike-policy pol1;
        address 20.0.0.1;
        local-address 10.1.0.1;
    }
```

```
}
    ipsec {
        vpn v1 {
            ike-gateway g1;
            group-vpn-external-interface ge-0/1/0;
            group 1;
        }
    }
```

```
[edit]
user@member1# show security policies
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
    from-zone trust to-zone untrust {
        policy scope1 {
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v1;
                    }
                }
            }
        }
        policy default-permit {
            match {
                source-address any;
```

```
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
        policy scope1 {
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v1;
                    }
                }
            }
        }
        policy default-deny {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                deny;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Member2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security group-vpn member ike proposal prop2 authentication-method pre-
shared-keys

set security group-vpn member ike proposal prop2 authentication-method pre-
shared-keys

set security group-vpn member ike proposal prop2 dh-group group2

set security group-vpn member ike proposal prop2 authentication-algorithm sha1

set security group-vpn member ike proposal prop2 encryption-algorithm 3des-cbc

set security group-vpn member ike policy pol2 mode main

set security group-vpn member ike policy pol2 proposals prop2

set security group-vpn member ike policy pol2 pre-shared-key ascii-text
"$ABC123"

set security group-vpn member ike gateway g2 ike-policy pol2

set security group-vpn member ike gateway g2 address 20.0.0.1
```

```
                set security group-vpn member ike gateway g2 local-address 10.2.0.1

                set security group-vpn member ipsec vpn v2 ike-gateway g2

                set security group-vpn member ipsec vpn v2 group-vpn-external-interface
ge-0/1/0

                set security group-vpn member ipsec vpn v2 group 1

                set security address-book book1 address 10_subnet 10.0.0.0/8

                set security address-book book1 address 10_1_0_0_16 10.1.0.0/16

                set security address-book book1 address multicast_net 239.0.0.0/8

                set security address-book book1 attach zone trust

                set security address-book book2 address 10_subnet 10.0.0.0/8

                set security address-book book2 address 10_1_0_0_16 10.1.0.0/16

                set security address-book book2 address multicast_net 239.0.0.0/8

                set security address-book book2 attach zone untrust

                set security policies from-zone trust to-zone untrust policy deny2 match
source-address 10_1_0_0_16

                set security policies from-zone trust to-zone untrust policy deny2 match
```

```
destination-address any

                set security policies from-zone trust to-zone untrust policy deny2 match
application any

                set security policies from-zone trust to-zone untrust policy deny2 then reject

                set security policies from-zone trust to-zone untrust policy scope2 match
source -address 10_subnet

                set security policies from-zone trust to-zone untrust policy scope2 match
destination-address 10_subnet

                set security policies from-zone trust to-zone untrust policy scope2 match
application any

                set security policies from-zone trust to-zone untrust policy scope2 then
permit tunnel ipsec-group-vpn v2

                set security policies from-zone trust to-zone untrust policy multicast-scope2
match source-address 10_subnet

                set security policies from-zone trust to-zone untrust policy multicast-scope2
match destination-address multicast-net

                set security policies from-zone trust to-zone untrust policy multicast-scope2
match application any

                set security policies from-zone trust to-zone untrust policy multicast-scope2
then permit tunnel ipsec-group-vpn v2

                set security policies from-zone untrust to-zone trust policy deny2 match
```

```
source-address any set security policies from-zone untrust to-zone trust policy multicast-scope2
ma tch application any set security policies from-zone untr


                set security policies from-zone untrust to-zone trust policy deny2 match
destination-address 10_1_0_0_16


                set security policies from-zone untrust to-zone trust policy deny2 match
application any


                set security policies from-zone untrust to-zone trust policy deny2 then reject


                set security policies from-zone untrust to-zone trust policy scope2 match
source-address 10_subnet


                set security policies from-zone untrust to-zone trust policy scope2 match
destination-address 10_subnet


                set security policies from-zone untrust to-zone trust policy scope2 match
application any


                set security policies from-zone untrust to-zone trust policy scope2 then
permit tunnel ipsec-group-vpn v2


                set security policies from-zone untrust to-zone trust policy multicast-scope2
match source-address 10_subnet


                set security policies from-zone untrust to-zone trust policy multicast-scope2
match destination-address multicast-net


                set security policies from-zone untrust to-zone trust policy multicast-scope2
match application any
```

```
                set security policies from-zone untrust to-zone trust policy multicast-scope2
then permit tunnel ipsec-group-vpn v2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure member2:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

```
[edit security group-vpn member ike proposal prop2]
user@member2# set authentication-method pre-shared-keys
user@member2# set dh-group group2
user@member2# set authentication-algorithm sha1
user@member2# set encryption-algorithm 3des-cbc
```

2. Define the IKE policy and set the remote gateway.

```
[edit security group-vpn member ike]
user@member2# set policy pol2 mode main proposals prop2 pre-shared-key ascii-text "$ABC123"
user@member2# set gateway g2 ike-policy pol2 address 20.0.0.1 local-address 10.2.0.1
```

3. Configure the group identifier, IKE gateway, and interface for member2.

```
[edit security group-vpn member ipsec]
user@member2# set vpn v2 group 1 ike-gateway g2 group-vpn-external-interface ge-0/1/0
```

To prevent packet fragmentation issues, we recommend that the interface used by the group members to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

4. Create an address book and attach it to the trust zone.

```
[edit security address-book book1]
user@member2# set address 10_subnet 10.0.0.0/8
user@member2# set address 10_1_0_0_16 10.1.0.0/16
user@member2# set address multicast_net 239.0.0.0/8
user@member2# set attach zone trust
```

5. Create another address book and attach it to the untrust zone.

```
[edit security address-book book2]
user@member2# set address 10_subnet 10.0.0.0/8
user@member2# set address 10_1_0_0_16 10.1.0.0/16
user@member2# set address multicast_net 239.0.0.0/8
user@member2# set attach zone untrust
```

6. Configure a scope policy from the trust zone to the untrust zone that blocks traffic from 10.1.0.0/16.

```
[edit security policies from-zone trust to-zone untrust]
user@member2# set policy deny2 match source-address 10_1_0_0_16 destination-address any
application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet destination-address 10_subnet
application any
user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet destination-address
multicast-net application any
```

```
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn v2
```

7. Configure a scope policy from the untrust zone to the trust zone that blocks traffic to 10.1.0.0/16.

```
[edit security policies from-zone untrust to-zone trust]
user@member2# set policy deny2 match source-address any destination-address 10_1_0_0_16
application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet destination-address 10_subnet
application any
user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet destination-address
multicast-net application any
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn v2
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn member` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@member2# show security group-vpn member
ike {
    proposal prop2 {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
    }
    policy pol2 {
        mode main;
        proposals prop2;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
```

```
    }
    gateway g2 {
        ike-policy pol2;
        address 20.0.0.1;
        local-address 10.2.0.1;
    }
}
    ipsec {
        vpn v2 {
            ike-gateway g2;
            group-vpn-external-interface ge-0/1/0;
            group 1;
        }
    }
```

```
[edit]
user@member2# show security policies
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
    from-zone trust to-zone untrust {
        policy deny2 {
            match {
                source-address 10_1_0_0_16;
                destination-address any;
                application any;
            }
            then {
                reject;
            }
        }
        policy scope2 {
```

```
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy multicast-scope2 {
            match {
                source-address 10_subnet;
                destination-address multicast-net;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy default-permit {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
        policy deny2 {
            match {
                source-address any;
                destination-address 10_1_0_0_16;
```

```
                application any;
            }
            then {
                reject;
            }
        }
        policy scope2 {
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy multicast-scope2 {
            match {
                source-address 10_subnet;
                destination-address multicast-net;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy default-deny {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                deny;
            }
```

```
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Dynamic Policies for Member1 | **633**
- Verifying Dynamic Policies for Member2 | **634**

To confirm that the configuration is working properly, perform this task:

Verifying Dynamic Policies for Member1

Purpose

View the dynamic policies installed on member1.

Action

After the group server downloads keys to member1, enter the `show security dynamic-policies` command from operational mode.

```
user@member1> show security dynamic-policies
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048580,AI: disabled, Scope
Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses: 10.1.0.0/16
  Destination addresses: 10.2.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
```

```
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048581,AI: disabled, Scope
Policy: 5
  Policy Type: Dynamic
  Sequence number: 2
  From zone: trust, To zone: untrust
  Source addresses: 10.1.0.0/16
  Destination addresses: 10.2.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
```

Meaning

The multicast policy p3 from the server is not installed on member1 because there is no scope policy configured on member1 that allows multicast traffic.

Verifying Dynamic Policies for Member2

Purpose

View the dynamic policies installed on member 2.

Action

After the group server downloads keys to member2, enter the `show security dynamic-policies` command from operational mode.

```
user@member2> show security dynamic-policies
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580,AI: disabled, Scope
Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses: 10.1.0.0/16
  Destination addresses: 10.2.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
```

```
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580,AI: disabled, Scope
Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses: 10.1.1.1/32
  Destination addresses: 239.1.1.1/32
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581,AI: disabled, Scope
Policy: 5
  Policy Type: Dynamic
  Sequence number: 2
  From zone: trust, To zone: untrust
  Source addresses: 10.2.0.0/16/0
  Destination addresses: 10.1.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581,AI: disabled, Scope
Policy: 5
  Policy Type: Dynamic
  Sequence number: 2
  From zone: trust, To zone: untrust
  Source addresses: 10.1.1.1/32
  Destination addresses: 239.1.1.1/32
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
```

Meaning

The policy p2 (for traffic from 10.1.0.0/16 to 10.2.0.0/16) from the server is not installed on member2, because it matches the deny2 security policy configured on member2.

Example: Configuring Group VPNv1 Server-Member Communication for Unicast Rekey Messages

IN THIS SECTION

- Requirements | 636
- Overview | 636
- Configuration | 637
- Verification | 637

This example shows how to enable the server to send unicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members.

Requirements

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation.
- Configure the group server and members for Phase 2 IPsec SA.
- Configure the group g1 on the group server.

Overview

In this example, you specify the following server-member communication parameters for group g1:

- The server sends unicast rekey messages to group members.
- 3des-cbc is used to encrypt traffic between the server and members.
- sha1 is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

Configuration

IN THIS SECTION

- Procedure | 637

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-member communication:

1. Set the communications type.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communications-type unicast
```

2. Set the encryption algorithm.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm 3des-cbc
```

3. Set the member authentication.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha1
```

Verification

To verify the configuration is working properly, enter the `show security group-vpn server group g1 server-member-communication` command.

SEE ALSO

Understanding IKE and IPsec Packet Processing | 81

Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages

IN THIS SECTION

- Requirements | 638
- Overview | 638
- Configuration | 639
- Verification | 641

This example shows how to enable the server to send multicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members.

Requirements

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation and Phase 2 IPsec SA. See "Example: Configuring Group VPNv1 Server and Members" on page 607 or "Example: Configuring Group VPNv1 with Server-Member Colocation" on page 641.
- Configure ge-0/0/1.0, which is the interface the server will use for sending multicast messages. See Junos OS Routing Protocols Library.
- Configure the multicast group address 226.1.1.1. See Junos OS Routing Protocols Library.

  IP multicast protocols must be configured to allow delivery of multicast traffic in the network. This example does not show multicast configuration.

Overview

In this example, you specify the following server-member communication for group g1:

- The server sends multicast rekey messages to group members by means of multicast address 226.1.1.1 and interface ge-0/0/1.0.

- 3des-cbc is used to encrypt traffic between the server and members.
- sha1 is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

Configuration

IN THIS SECTION

- Procedure | 639

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security group-vpn server group g1 server-member-communication communication-type multicast
set security group-vpn server group g1 server-member-communication multicast-group 226.1.1.1
set security group-vpn server group g1 server-member-communication multicast-outgoing-interface
ge-0/0/1.0
set security group-vpn server group g1 server-member-communication encryption-algorithm 3des-cbc
set security group-vpn server group g1 server-member-communication sig-hash-algorithm sha1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure configure server-member communication for multicast rekey messages:

1. Set the communications type.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communication-type multicast
```

2. Set the multicast group.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set multicast-group 226.1.1.1
```

3. Set the interface for outgoing multicast messages.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set multicast-outgoing-interface ge-0/0/1.0
```

4. Set the encryption algorithm.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm 3des-cbc
```

5. Set the member authentication.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha1
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn server group g1 server-member-communication` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server group g1 server-member-communication
communication-type multicast;
multicast-group 226.1.1.1;
multicast-outgoing-interface ge-0/0/1.0;
encryption-algorithm 3des-cbc;
sig-hash-algorithm sha1;
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Server-Member Communication for Multicast Rekey Messages | 641

To confirm that the configuration is working properly, perform these tasks:

Verifying Server-Member Communication for Multicast Rekey Messages

Purpose

Verify that server-member communication parameters for multicast rekey message are configured properly to ensure that valid keys are available for encrypting traffic between group members.

Action

From operational mode, enter the `show security group-vpn server group g1 server-member-communication` command.

SEE ALSO

Understanding IKE and IPsec Packet Processing | 81

Example: Configuring Group VPNv1 with Server-Member Colocation

IN THIS SECTION

- Requirements | 642
- Overview | 642
- Configuration | 643
- Verification | 653

This example shows how to configure a device for colocation mode, which allows server and member functions to coexist on the same physical device.

Requirements

Before you begin:

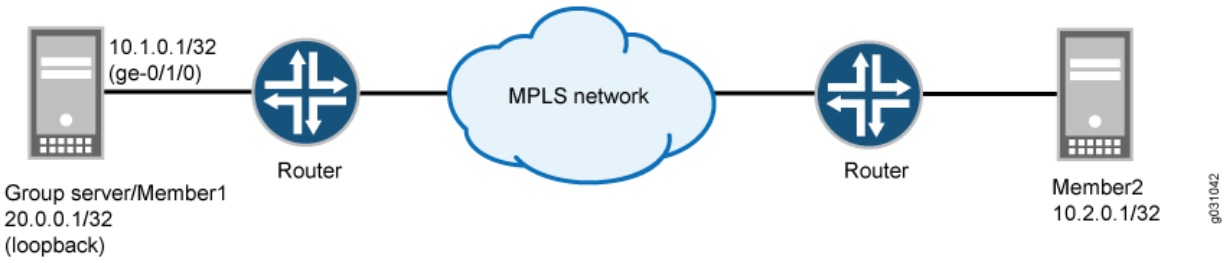
- Configure the Juniper Networks security devices for network communication.
- Configure network interfaces on server and member devices. See Interfaces User Guide for Security Devices.

Overview

When colocation mode is configured, group server and group member functions can coexist in the same device. In colocation mode, the server and member must have different IP addresses so that packets are delivered properly.

In Figure 41 on page 642, a group VPN (group identifier is 1) consists of two members (member1 and member2) and a group server (the IP address of the loopback interface is 20.0.0.1). Note that member1 coexists in the same device as the group server. In this example, the interface that member1 uses to connect to the MPLS network (ge-0/1/0) is assigned the IP address 10.1.0.1/32.

Figure 41: Server-Member Colocation Example



The configuration instructions in this topic describe how to configure the group server-member1 device for colocation mode. To configure member2, see "Example: Configuring Group VPNv1 Server and Members" on page 607.

To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set` *`interface`* `mtu` configuration statement to set the MTU size.

Configuration

IN THIS SECTION

- Procedure | 643

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces lo0 unit 0 family inet address 20.0.0.1/32
set interfaces ge-0/1/0 unit 0 family inet address 10.1.0.1/32
set security group-vpn member ike proposal prop1 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop1 dh-group group2
set security group-vpn member ike proposal prop1 authentication-algorithm sha1
set security group-vpn member ike proposal prop1 encryption-algorithm 3des-cbc
set security group-vpn member ike policy pol1 mode main
set security group-vpn member ike policy pol1 proposals prop1
set security group-vpn member ike policy pol1 pre-shared-key ascii-text "$9$c1gr K8-
VYZUHX7UHqmF3Sre"
set security group-vpn member ike gateway g1 ike-policy pol1
set security group-vpn member ike gateway g1 address 20.0.0.1
set security group-vpn member ike gateway g1 local-address 10.1.0.1
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security group-vpn member ipsec vpn v1 group 1
set security group-vpn server ike proposal srv-prop authentication-method pre-shared-keys
set security group-vpn server ike proposal srv-prop dh-group group2
set security group-vpn server ike proposal srv-prop authentication-algorithm sha1
set security group-vpn server ike proposal srv-prop encryption-algorithm 3des-cbc
set security group-vpn server ike policy srv-pol mode main
set security group-vpn server ike policy srv-pol proposals srv-prop
set security group-vpn server ike policy srv-pol pre-shared-key ascii-text "$9$c 1grK8-
VYZUHX7UHqmF3Sre"
set security group-vpn server ike gateway gw1 ike-policy srv-pol
```

```
set security group-vpn server ike gateway gw1 address 10.1.0.1
set security group-vpn server ike gateway gw2 ike-policy srv-pol
set security group-vpn server ike gateway gw2 address 10.2.0.1
set security group-vpn server ipsec proposal group-prop authentication-algorithm hmac-sha1-96
set security group-vpn server ipsec proposal group-prop encryption-algorithm 3des-cbc
set security group-vpn server ipsec proposal group-prop lifetime-seconds 3600
set security group-vpn server group grp1 group-id 1
set security group-vpn server group grp1 ike-gateway gw1
set security group-vpn server group grp1 ike-gateway gw2
set security group-vpn server group grp1 anti-replay-time-window 120
set security group-vpn server group grp1 server-address 20.0.0.1
set security group-vpn server group grp1 server-member-communication communication-type unicast
set security group-vpn server group grp1 server-member-communication encryption-algorithm
aes-128-cbc
set security group-vpn server group grp1 server-member-communication sig-hash-algorithm md5
set security group-vpn server group grp1 server-member-communication certificate srv-cert
set security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source 10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination
10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source 10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination
10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source 10.1.1.1/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination
239.1.1.1/32
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 protocol 0
set security group-vpn co-location
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security address-book book1 address 10_subnet 10.0.0.0/8
set security address-book book1 attach zone trust
set security address-book book2 address 10_subnet 10.0.0.0/8
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy scope1 match source-address
```

```
10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match destination-address
10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match application any
set security policies from-zone trust to-zone untrust policy scope1 then permit tunnel ipsec-
group-vpn v1
set security policies from-zone untrust to-zone trust policy scope1 match source-address
10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match destination-address
10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match application any
set security policies from-zone untrust to-zone trust policy scope1 then permit tunnel ipsec-
group-vpn v1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure group VPN with server-member colocation:

1. Configure the loopback address on the device.

```
[edit interfaces]
user@host# set lo0 unit 0 family inet address 20.0.0.1/32
```

2. Configure the interface that member1 uses to connect to the MPLS network.

```
[edit interfaces]
user@host# set ge-0/1/0 unit 0 family inet address 10.1.0.1/32
```

3. Configure group VPN colocation on the device.

```
[edit security group-vpn]
user@host# set co-location
```

4. Configure IKE Phase 1 SA for the server (this configuration must match the Phase 1 SA configured on group members).

```
[edit security group-vpn server ike proposal srv-prop]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm 3des-cbc
```

5. Define the IKE policy and set the remote gateways.

```
[edit security group-vpn server ike]
user@host# set policy srv-pol proposals srv-prop mode main pre-shared-key ascii-text
"$9$c1grK8-VYZUHX7UHqmF3Sre"
user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1
```

6. Configure the Phase 2 SA exchange for the server.

```
[edit security group-vpn server ipsec proposal group-prop]
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 3600
```

7. Configure the group identifier, IKE gateway, antireplay time, and server address on the server.

```
[edit security group-vpn server group grp1]
user@host# set group-id 1 anti-replay-time-window 120 server-address 20.0.0.1
user@host#set ike-gateway gw1
user@host#set ike-gateway gw2
```

8. Configure server to member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast encryption-algorithm
aes-128-cbc sig-hash-algorithm md5 certificate “srv-cert”
```

9. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa ]
user@host# set proposal group-prop match-policy p1 source 10.1.0.0/16 destination
10.2.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p2 source 10.2.0.0/16 destination
10.1.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p3 source 10.1.1.1/16 destination
239.1.1.1/32 source-port 0 destination-port 0 protocol 0
```

10. Configure Phase 1 SA for member1 (this configuration must match the Phase 1 SA configured for the group server).

```
[edit security group-vpn member ike proposal prop1]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm 3des-cbc
```

11. Define the policy and set the remote gateway for member1.

```
[edit security group-vpn member ike]
user@host# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text "$9$c1grK8-
VYZUHX7UHqmF3Sre"
user@host# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address 10.1.0.1
```

12. Configure the group identifier, IKE gateway, and interface for member1.

```
[edit security group-vpn member ipsec]
user@host# set vpn v1 group 1 ike-gateway g1 group-vpn-external-interface ge-0/1/0
```

13. Create address books and attach them to zones.

```
[edit security address-book book1]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone trust
```

```
[edit security address-book book2]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone untrust
```

14. Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address 10_subnet destination-address
10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

15. Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address 10_subnet destination-address
10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn` and `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

In the list of configured security policies, make sure that the scope policies are listed before the default policies.

```
[edit]
user@host# show security group-vpn
member {
```

```
    ike {
        proposal prop1 {
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
        }
        policy pol1 {
            mode main;
            proposals prop1;
            pre-shared-key ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"; ## SECRET-DATA
        }
        gateway g1 {
            ike-policy pol1;
            address 20.0.0.1;
            local-address 10.1.0.1;
        }
    }
    ipsec {
        vpn v1 {
            ike-gateway g1;
            group-vpn-external-interface ge-0/1/0;
            group 1;
        }
    }
    }
    server {
        ike {
            proposal srv-prop {
                authentication-method pre-shared-keys;
                dh-group group2;
                authentication-algorithm sha1;
                encryption-algorithm 3des-cbc;
            }
            policy srv-pol {
                mode main;
                proposals srv-prop;
                pre-shared-key ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"; ## SECRET-DATA
            }
            gateway gw1 {
                ike-policy srv-pol;
                address 10.1.0.1;
            }
```

```
            gateway gw2 {
                ike-policy srv-pol;
                address 10.2.0.1;
            }
        }
        ipsec {
            proposal group-prop {
                authentication-algorithm hmac-sha1-96;
                encryption-algorithm 3des-cbc;
                lifetime-seconds 3600;
            }
        }
        group grp1 {
            group-id 1;
            ike-gateway gw1;
            ike-gateway gw2;
            anti-replay-time-window 120;
            server-address 20.0.0.1;
            server-member-communication {
                communication-type unicast;
                encryption-algorithm aes-128-cbc;
                sig-hash-algorithm md5;
                certificate srv-cert;
            }
            ipsec-sa group-sa {
                proposal group-prop;
                match-policy p1 {
                    source 10.1.0.0/16;
                    destination 10.2.0.0/16;
                    source-port 0;
                    destination-port 0;
                    protocol 0;
                }
                match-policy p2 {
                    source 10.2.0.0/16;
                    destination 10.1.0.0/16;
                    source-port 0;
                    destination-port 0;
                    protocol 0;
                }
                match-policy p3 {
                    source 10.1.1.1/16;
                    destination 239.1.1.1/32;
```

```
                    source-port 0;
                    destination-port 0;
                    protocol 0;
                }
            }
        }
    }
co-location;
```

```
[edit]
user@host# show security policies
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone trust to-zone untrust {
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
    policy default-permit {
        match {
            source-address any;
```

```
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone untrust to-zone trust {
    policy default-deny {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Group VPN Member Registration | 653
- Verifying Group VPN Server Security Associations for IKE | 653
- Verifying Group VPN Server Security Associations for IPsec | 653
- Verifying Group VPN Member Security Associations for IKE | 654
- Verifying Group VPN Member Security Associations for IPsec | 654

To confirm that the configuration is working properly, perform these tasks:

Verifying Group VPN Member Registration

Purpose

Verify that the group VPN members are registered correctly.

Action

From operational mode, enter the `show security group-vpn registered-members` command.

Verifying Group VPN Server Security Associations for IKE

Purpose

Verify the SAs for the group VPN server for IKE.

Action

From operational mode, enter the `show security group-vpn server ike security-associations` command.

Verifying Group VPN Server Security Associations for IPsec

Purpose

Verify the SAs for the group VPN server for IPsec.

Action

From operational mode, enter the `show security group-vpn server ipsec security-associations` command.

Verifying Group VPN Member Security Associations for IKE

Purpose

Verify the SAs for the group VPN members for IKE.

Action

From operational mode, enter the `show security group-vpn member ike security-associations` command.

Verifying Group VPN Member Security Associations for IPsec

Purpose

Verify the SAs for the group VPN members for IPsec.

Action

From operational mode, enter the `show security group-vpn member ipsec security-associations` command.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
12.3X48-D30	Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members can interoperate with Group VPNv2 servers.
12.3X48-D30	Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members on SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, and SRX650 devices can interoperate with Group VPNv2 servers.

Group VPNv2

SUMMARY

Read this topic to learn about Group VPNv2 in Junos OS.

IN THIS SECTION

- Group VPNv2 Overview | **655**
- Group VPNv2 Configuration Overview | **661**
- Understanding IKE Phase 1 Configuration for Group VPNv2 | **662**
- Understanding IPsec SA Configuration for Group VPNv2 | **663**
- Understanding Group VPNv2 Traffic Steering | **663**
- Understanding the Group VPNv2 Recovery Probe Process | **665**
- Understanding Group VPNv2 Antireplay | **666**
- Example: Configuring a Group VPNv2 Server and Members | **666**
- Example: Configuring Group VPNv2 Server-Member Communication for Unicast Rekey Messages | **713**

Group VPNv2 introduces the concept of a trusted group to eliminate point-to-point tunnels and their associated overlay routing. All group members share a common security association (SA), also known as a group SA.

Group VPNv2 Overview

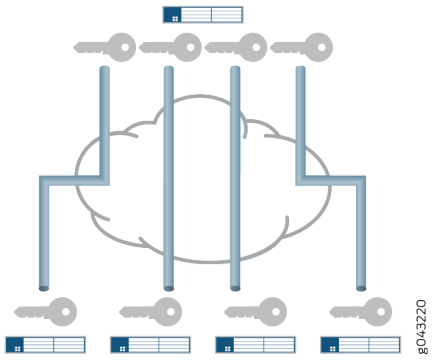
IN THIS SECTION

- Understanding the GDOI Protocol for Group VPNv2 | **657**
- Understanding Group VPNv2 Servers and Members | **657**

- Understanding Group VPNv2 Limitations | 658
- Understanding Group VPNv2 Server-Member Communication | 659
- Understanding Group VPNv2 Key Operations | 659

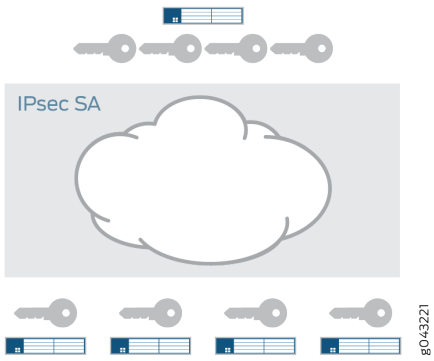
An IPsec security association (SA) is a unidirectional agreement between virtual private network (VPN) participants that defines the rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications. With many VPN implementations, the SA is a point-to-point tunnel between two security devices (see Figure 42 on page 656).

Figure 42: Point-to-Point SAs



Group VPNv2 extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 43 on page 656). With Group VPNv2, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header.

Figure 43: Shared SAs



Group VPNv2 is an enhanced version of the group VPN feature introduced in an earlier Junos OS release for SRX Series Firewalls. Group VPNv2 on Juniper devices support RFC 6407, *The Group Domain of Interpretation (GDOI)*, and interoperate with other devices that comply with RFC 6407.

Understanding the GDOI Protocol for Group VPNv2

Group VPNv2 is based on RFC 6407, *The Group Domain of Interpretation* (GDOI). This RFC describes the protocol between group members and group servers to establish SAs among group members. GDOI messages create, maintain, or delete SAs for a group of devices.

The GDOI protocol runs on UDP port 848. The Internet Security Association and Key Management Protocol (ISAKMP) defines two negotiation phases to establish SAs for an IKE IPsec tunnel. Phase 1 allows two devices to establish an ISAKMP SA for other security protocols, such as GDOI.

With Group VPNv2, Phase 1 ISAKMP SA negotiation is performed between a group server and a group member. The server and member must use the same ISAKMP policy. GDOI exchanges between the server and member establish the SAs that are shared with other group members. A group member does not need to negotiate IPsec with other group members. GDOI exchanges must be protected by ISAKMP Phase 1 SAs.

There are two types of GDOI exchanges:

- The `groupkey-pull` exchange allows a member to request SAs and keys shared by the group from the server. Group members must register with a group server through a `groupkey-pull` exchange.
- The `groupkey-push` exchange is a single rekey message that allows the server to send group SAs and keys to members before existing group SAs expire. Rekey messages are unsolicited messages sent from the server to members.

Understanding Group VPNv2 Servers and Members

The center of Group VPNv2 is the group controller/key server (GCKS). A server cluster can be used to provide GCKS redundancy.

The GCKS or group server performs the following tasks:

- Controls group membership.
- Generates encryption keys.
- Sends new group SAs and keys to members. Group members encrypt traffic based on the group SAs and keys provided by the group server.

A group server can service multiple groups. A single security device can be a member of multiple groups.

Each group is represented by a group identifier, which is a number between 1 and 4,294,967,295. The group server and group members are linked together by the group identifier. There can be only one group identifier per group, and multiple groups cannot use the same group identifier.

The following is a high-level view of Group VPNv2 server and member actions:

1. The group server listens on UDP port 848 for members to register.
2. To register with the group server, the member first establishes an IKE SA with the server. A member device must provide correct IKE Phase 1 authentication to join the group. Preshared key authentication on a per-member basis is supported.
3. Upon successful authentication and registration, the member device retrieves group SAs and keys for the specified group identifier from the server with a GDOI `groupkey-pull` exchange.
4. The server adds the member to the membership for the group.
5. Group members exchange packets encrypted with group SA keys.

The server sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. The server sends rekey messages before SAs expire to ensure that valid keys are available for encrypting traffic between group members.

A rekey message sent by the server requires an acknowledgement (ack) message from each group member. If the server does not receive an ack message from the member, the rekey message is retransmitted at the configured `retransmission-period` (the default is 10 seconds). If there is no reply from the member after the configured `number-of-retransmission` (the default is 2 times), the member is removed from the server's registered members. The IKE SA between the server and member is also removed.

The server also sends rekey messages to provide new keys to members when the group SA has changed.

Understanding Group VPNv2 Limitations

Group VPNv2 servers only operate with Group VPNv2 members that support RFC 6407, *The Group Domain of Interpretation (GDOI)*.

The following are not supported in Group VPNv2:

- SNMP.
- Deny policy from Cisco GET VPN server.
- PKI support for Phase 1 IKE authentication.
- Colocation of group server and member, where server and member functions coexist in the same physical device.
- Group members configured as chassis clusters.

- J-Web interface for configuration and monitoring.
- Multicast data traffic.

Group VPNv2 is not supported in deployments where IP addresses cannot be preserved—for example, across the Internet where NAT is used.

Understanding Group VPNv2 Server-Member Communication

Server-member communication allows the server to send GDOI `groupkey-push` (rekey) messages to members. If server-member communication is not configured for the group, members can send GDOI `groupkey-pull` messages to register and reregister with the server, but the server is not able to send `groupkey-push` messages to members.

Server-member communication is configured for the group by using the `server-member-communication` *configuration statement* at the [`edit security group-vpn server`] hierarchy. The following options can be defined:

- Authentication algorithm (sha-256 or sha-384) used to authenticate the member to the server. There is no default algorithm.
- Encryption algorithm used for communications between the server and member. You can specify aes-128-cbc, aes-192-cbc, or aes-256-cbc. There is no default algorithm.
- Unicast communication type for rekey messages sent to group members.
- Lifetime for the key encryption key (KEK). The default is 3600 seconds.
- Number of times the group server retransmits `groupkey-push` messages to a group member without a response (the default is 2 times) and the period of time between retransmissions (the default is 10 seconds).

If server-member communication for a group is not configured, the membership list displayed by the `show security group-vpn server registered-members` command shows group members who have registered with the server; members can be active or not. When server-member communication for a group is configured, the group membership list is cleared. For unicast communication type, the `show security group-vpn server registered-members` command shows only active members.

Understanding Group VPNv2 Key Operations

This topic contains the following sections:

Group Keys

The group server maintains a database to track the relationship among VPN groups, group members, and group keys. There are two kinds of group keys that the server downloads to members:

- Key Encryption Key (KEK)—Used to encrypt SA rekey (GDOI `groupkey-push`) exchanges. One KEK is supported per group.
- Traffic Encryption Key (TEK)—Used to encrypt and decrypt IPsec data traffic between group members.

The key associated with an SA is accepted by a group member only if there is a matching policy configured on the member. An accepted key is installed for the group, whereas a rejected key is discarded.

Rekey Messages

If the group is configured for server-member communications, the server sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or the group SA has changed (for example, a group policy is added or deleted).

Server-member communications options must be configured on the server to allow the server to send rekey messages to group members.

The group server sends one copy of the unicast rekey message to each group member. Upon receipt of the rekey message, members must send an acknowledgment (ACK) to the server. If the server does not receive an ACK from a member (including retransmission of rekey messages), the server considers the member to be inactive and removes it from the membership list. The server stops sending rekey messages to the member.

The `number-of-retransmission` and `retransmission-period` configuration statements for server-member communications control the resending of rekey messages by the server when no ACK is received from a member.

The interval at which the server sends rekey messages is based on the value of the `lifetime-seconds` configuration statement at the [`edit security group-vpn server group` *`group-name`*] hierarchy. New keys are generated before the expiration of the KEK and TEK keys.

The `lifetime-seconds` for the KEK is configured as part of the server-member communications; the default is 3600 seconds. The `lifetime-seconds` for the TEK is configured for the IPsec proposal; the default is 3600 seconds.

Member Registration

If a group member does not receive a new SA key from the server before the current key expires, the member must reregister with the server and obtain updated keys with a GDOI `groupkey-pull` exchange.

Group VPNv2 Configuration Overview

This topic describes the main tasks for configuring Group VPNv2.

The group controller/key server (GCKS) manages Group VPNv2 security associations (SAs), and generates encryption keys and distributes them to group members. You can use a Group VPNv2 server cluster to provide GCKS redundancy. See "Understanding Group VPNv2 Server Clusters" on page 716.

On the group server(s), configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 662.
2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 663.
3. VPN group information, including the group identifier, IKE gateways for group members, the maximum number of members in the group, and server-member communications. Group configuration includes a group policy that defines the traffic to which the SA and keys apply. Server cluster and antireplay time window can optionally be configured. See "Group VPNv2 Configuration Overview" on page 661 and "Understanding Group VPNv2 Traffic Steering" on page 663.

On the group member, configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 662.
2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 663.
3. IPsec policy that defines the incoming zone (usually a protected LAN), outgoing zone (usually a WAN) and the VPN group to which the policy applies. Exclude or fail-open rules can also be specified. See "Understanding Group VPNv2 Traffic Steering" on page 663.
4. Security policy to allow group VPN traffic between the zones specified in the IPsec policy.

Group VPNv2 operation requires a working routing topology that allows client devices to reach their intended sites throughout the network.

The group is configured on the server with the `group` *configuration statement* at the [`edit security group-vpn server`] hierarchy.

The group information consists of the following information:

- Group identifier—A value that identifies the VPN group. The same group identifier must be configured on the group member.

- Each group member is configured with the `ike-gateway` configuration statement. There can be multiple instances of this configuration statement, one for each member of the group.
- Group policies—Policies that are to be downloaded to members. Group policies describe the traffic to which the SA and keys apply. See "Understanding Group VPNv2 Traffic Steering" on page 663.
- Member threshold—The maximum number of members in the group. After the member threshold for a group is reached, a server stops responding to `groupkey-pull` initiations from new members. See "Understanding Group VPNv2 Server Clusters" on page 716.
- Server-member communication—Optional configuration that allows the server to send `groupkey-push` rekey messages to members.
- Server cluster—Optional configuration that supports group controller/key server (GCKS) redundancy. See "Understanding Group VPNv2 Server Clusters" on page 716.
- Antireplay—Optional configuration that detects packet interception and replay. See "Understanding Group VPNv2 Antireplay" on page 666.

Understanding IKE Phase 1 Configuration for Group VPNv2

An IKE Phase 1 SA between a group server and a group member establishes a secure channel in which to negotiate IPsec SAs that are shared by a group. For standard IPsec VPNs on Juniper Networks security devices, Phase 1 SA configuration consists of specifying an IKE proposal, policy, and gateway.

For Group VPNv2, the IKE Phase 1 SA configuration is similar to the configuration for standard IPsec VPNs, but is performed at the [`edit security group-vpn server ike`] and [`edit security group-vpn member ike`] hierarchies.

In the IKE proposal configuration, you set the authentication method and the authentication and encryption algorithms that will be used to open a secure channel between participants. In the IKE policy configuration, you set the mode in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal. In the IKE gateway configuration, you reference the Phase 1 policy.

The IKE proposal and policy configuration on the group server must match the IKE proposal and policy configuration on group members. On a group server, an IKE gateway is configured for each group member. On a group member, up to four server addresses can be specified in the IKE gateway configuration.

Understanding IPsec SA Configuration for Group VPNv2

After the server and member have established a secure and authenticated channel in Phase 1 negotiation, they proceed to establish the IPsec SAs that are shared by group members to secure data that is transmitted among members. While the IPsec SA configuration for Group VPNv2 is similar to the configuration for standard VPNs, a group member does not need to negotiate the SA with other group members.

IPsec configuration for Group VPNv2 consists of the following information:

- On the group server, an IPsec proposal is configured for the security protocol, authentication, and encryption algorithm to be used for the SA. The IPsec SA proposal is configured on the group server with the `proposal` configuration statement at the [`edit security group-vpn server ipsec`] hierarchy.
- On the group member, an Autokey IKE is configured that references the group identifier, the group server (configured with the `ike-gateway` configuration statement), and the interface used by the member to connect to group peers. The Autokey IKE is configured on the member with the `vpn` configuration statement at the [`edit security group-vpn member ipsec`] hierarchy.

SEE ALSO

Understanding Group VPNv2 Server Clusters | **716**

Understanding Group VPNv2 Traffic Steering

IN THIS SECTION

- Group Policies Configured on Group Servers | **664**
- IPsec Policies Configured on Group Members | **664**
- Fail-Close | **664**
- Exclude and Fail-Open Rules | **664**
- Priorities of IPsec Policies and Rules | **665**

The group server distributes IPsec security associations (SAs) and keys to members of a specified group. All members that belong to the same group share the same set of IPsec SAs. The SA that is installed on a

specific group member is determined by the policy associated with the group SA and the IPsec policy that is configured on the group member.

Group Policies Configured on Group Servers

In a VPN group, each group SA and key that the server pushes to a member are associated with a group policy. The group policy describes the traffic on which the key should be used, including protocol, source address, source port, destination address, and destination port. On the server, the group policy is configured with the `match-policy` *`policy-name`* options at the [`edit security group-vpn server group` *`name`* `ipsec-sa` *`name`*] hierarchy level.

Group policies that are identical (configured with the same source address, destination address, source port, destination port, and protocol values) cannot exist for a single group. An error is returned if you attempt to commit a configuration that contains identical group policies for a group. If this occurs, you must delete one of the identical group policies before you can commit the configuration.

IPsec Policies Configured on Group Members

On the group member, an IPsec policy consists of the following information:

- Incoming zone (`from-zone`) for group traffic.
- Outgoing zone (`to-zone`) for group traffic.
- The name of the group to which the IPsec policy applies. Only one Group VPNv2 name can be referenced by a specific from-zone/to-zone pair.

The interface that is used by the group member to connect to the Group VPNv2 must belong to the outgoing zone. This interface is specified with the `group-vpn-external-interface` statement at the [`edit security group-vpn member ipsec vpn` *`vpn-name`*] hierarchy level.

On the group member, the IPsec policy is configured at the [`edit security ipsec-policy`] hierarchy level. Traffic that matches the IPsec policy is further checked against exclude and fail-open rules that are configured for the group.

Fail-Close

By default, traffic that does not match exclude or fail-open rules or group policies received from the group server is blocked; this is known as *fail-close*.

Exclude and Fail-Open Rules

On group members, the following types of rules can be configured for each group:

- Traffic that is excluded from VPN encryption. Examples of this type of traffic can include BGP or OSPF routing protocols. To exclude traffic from a group, use the `set security group-vpn member ipsec vpn` *`vpn-name`* `exclude rule` configuration. A maximum of 10 exclude rules can be configured.
- Traffic that is critical to the customer's operation and must be sent in cleartext (unencrypted) if the group member has not received a valid traffic encryption key (TEK) for the IPsec SA. Fail-open rules allow this traffic flow while all other traffic is blocked. Enable fail-open with the `set security group-vpn member ipsec vpn` *`vpn-name`* `fail-open rule` configuration. A maximum of 10 fail-open rules can be configured.

Priorities of IPsec Policies and Rules

IPsec policies and rules have the following priorities on the group member:

1. Exclude rules that define traffic to be excluded from VPN encryption.
2. Group policies that are downloaded from the group server.
3. Fail-open rules that define traffic that is sent in cleartext if there is no valid TEK for the SA.
4. Fail-close policy that blocks traffic. This is the default if traffic does not match exclude or fail-open rules or group policies.

SEE ALSO

Understanding Configuration Changes with Group VPNv2 Server Clusters | 723

Understanding the Group VPNv2 Recovery Probe Process

Two situations could indicate that a group member is out of synchronization with the group server and other group members:

- The group member receives an Encapsulating Security Payload (ESP) packet with an unrecognized Security Parameter Index (SPI).
- There is outgoing IPsec traffic but no incoming IPsec traffic on the group member.

When either situation is detected, a recovery probe process can be triggered on the group member. The recovery probe process initiates GDOI `groupkey-pull` exchanges at specific intervals to update the member's SA from the group server. If there is a DoS attack of bad SPI packets or if the sender itself is out of synchronization, the out-of-synchronization indication on the group member might be a false

alarm. To avoid overloading the system, the `groupkey-pull` initiation is retried at intervals of 10, 20, 40, 80, 160, and 320 seconds.

The recovery probe process is disabled by default. To enable the recovery probe process, configure `recovery-probe` at the [`edit security group-vpn member ipsec vpn` *`vpn-name`*] hierarchy level.

Understanding Group VPNv2 Antireplay

Antireplay is an IPsec feature that can detect when a packet is intercepted and then replayed by attackers. Antireplay is disabled by default for a group.

Each IPsec packet contains a timestamp. The group member checks whether the packet's timestamp falls within the configured `anti-replay-time-window` value. A packet is dropped if the timestamp exceeds the value.

We recommend that NTP be configured on all devices that support Group VPNv2 antireplay.

Group members that are running on vSRX Virtual Firewall instances on a host machine where the hypervisor is running under a heavy load can experience issues that can be corrected by reconfiguring the `anti-replay-time-window` value. If data that matches the IPsec policy on the group member is not being transferred, check the `show security group-vpn member ipsec statistics` output for D3P errors. Make sure that NTP is operating correctly. If there are errors, adjust the `anti-replay-time-window` value.

SEE ALSO

Understanding Antireplay for Group VPNv1 | **606**

Example: Configuring a Group VPNv2 Server and Members

IN THIS SECTION

- Requirements | **667**
- Overview | **667**
- Configuration | **668**
- Verification | **704**

This example shows how to configure a Group VPNv2 server to provide group controller/key server (GCKS) support to Group VPNv2 group members.

Requirements

The example uses the following hardware and software components:

- A supported SRX Series Firewall or vSRX Virtual Firewall instance running Junos OS Release 15.1X49-D30 or later that supports Group VPNv2. This SRX Series Firewall or vSRX Virtual Firewall instance operates as a Group VPNv2 server.
- Two supported SRX Series Firewalls or vSRX Virtual Firewall instances running Junos OS Release 15.1X49-D30 or later that support Group VPNv2. These devices or instances operate as Group VPNv2 group members.
- Two supported MX Series devices running Junos OS Release 15.1R2 or later that support Group VPNv2. These devices operate as Group VPNv2 group members.

A hostname, a root administrator password, and management access must be configured on each device. We recommend that NTP also be configured on each device.

Group VPNv2 operation requires a working routing topology that allows client devices to reach their intended sites throughout the network. This examples focuses on the Group VPNv2 configuration; the routing configuration is not described.

Overview

IN THIS SECTION

- Topology | **668**

In this example, the Group VPNv2 network consists of a server and four members. Two of the members are SRX Series Firewalls or vSRX Virtual Firewall instances while the other two members are MX Series devices. The shared group VPN SAs secure traffic between group members.

The group VPN SAs must be protected by a Phase 1 SA. Therefore, the group VPN configuration must include configuring IKE Phase 1 negotiations on both the group server and the group members.

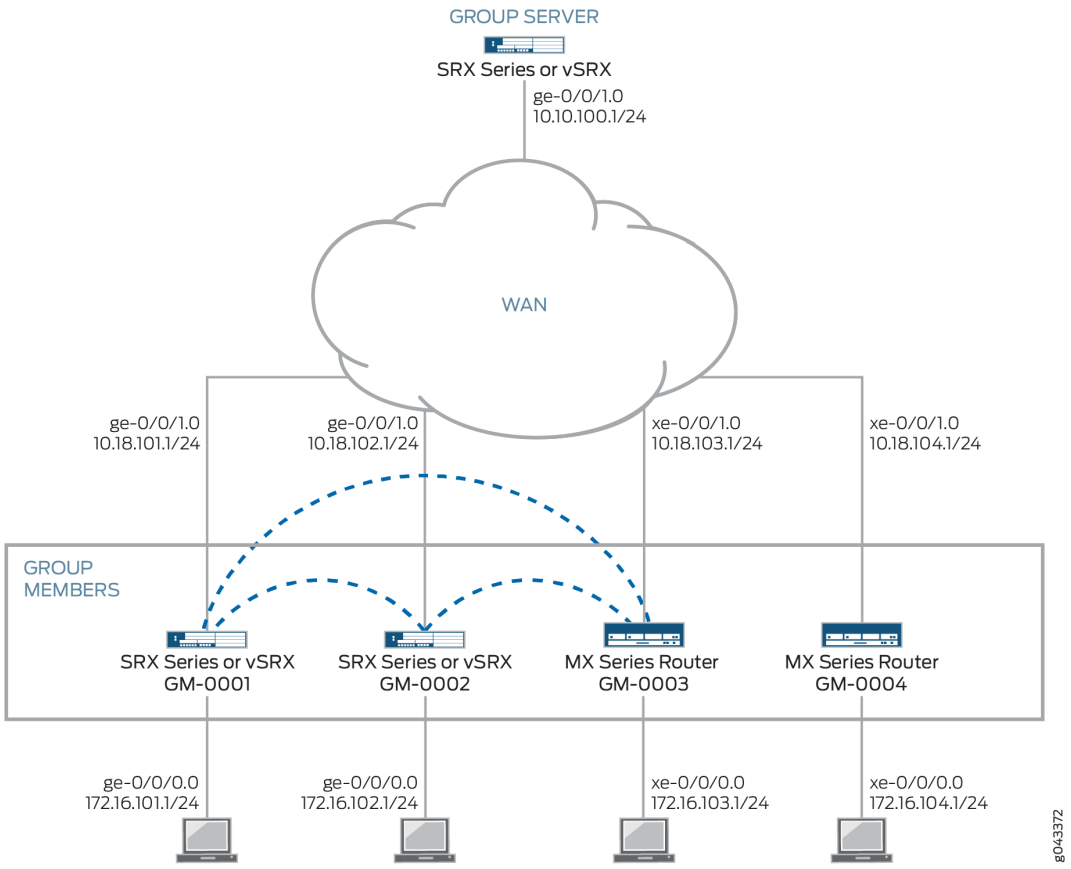
The same group identifier must be configured on both the group server and the group members. In this example, the group name is GROUP_ID-0001 and the group identifier is 1. The group policy configured on the server specifies that the SA and key are applied to traffic between subnetworks in the 172.16.0.0/12 range.

On SRX Series Firewall or vSRX Virtual Firewall group members, an IPsec policy is configured for the group with the LAN zone as the from-zone (incoming traffic) and the WAN zone as the to-zone (outgoing traffic). A security policy is also needed to allow traffic between the LAN and WAN zones.

Topology

Figure 44 on page 668 shows the Juniper Networks devices to be configured for this example.

Figure 44: Group VPNv2 Server with SRX Series Firewall or vSRX Virtual Firewall and MX Series Members



Configuration

IN THIS SECTION

- Configuring the Group Server | 669

- Configuring Group Member GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **676**
- Configuring Group Member GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **684**
- Configuring Group Member GM-0003 (MX Series Device) | **692**
- Configuring Group Member GM-0004 (MX Series Device) | **698**

Configuring the Group Server

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.10.100.1/24
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then reject
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set routing-options static route 10.18.101.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.102.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.103.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.104.0/24 next-hop 10.10.100.254
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
```

```
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.10.100.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0005
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
```

```
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 server:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.10.100.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security policies]
user@host# set global policy 1000 match source-address any
user@host# set global policy 1000 match destination-address any
user@host# set global policy 1000 match application any
user@host# set global policy 1000 match from-zone any
user@host# set global policy 1000 match to-zone any
user@host# set global policy 1000 then reject
user@host# set global policy 1000 then log session-init
user@host# set global policy 1000 then count
user@host# set default-policy deny-all
```

2. Configure the static routes.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.102.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.103.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.104.0/24 next-hop 10.10.100.254
```

3. Configure the IKE proposal, policy, and gateways.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm sha-256
user@host# set dh-group group14
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy  GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.10.100.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.10.100.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.10.100.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.10.100.1
```

4. Configure the IPsec proposal.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600 VPN Group
```

5. Configure the group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
```

```
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
```

6. Configure server-to-member communications.

```
[edit security group-vpn server group GROUP_ID-0001 server-member-communication]
user@host# set communication-type unicast
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 7200
user@host# set sig-hash-algorithm sha-256
```

7. Configure the group policy to be downloaded to the group members.

```
[edit security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001]
user@host# set proposal AES256-SHA256-L3600
user@host# set match-policy 1 source 172.16.0.0/12
user@host# set match-policy 1 destination 172.16.0.0/12
user@host# set match-policy 1 protocol 0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.100.1/24;
        }
    }
}
[edit]
user@host# show routing-options
```

```
static {
    route 10.18.101.0/24 next-hop 10.10.100.254;
    route 10.18.102.0/24 next-hop 10.10.100.254;
    route 10.18.103.0/24 next-hop 10.10.100.254;
    route 10.18.104.0/24 next-hop 10.10.100.254;
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.10.100.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.10.100.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.10.100.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.10.100.1;
            }
        }
```

```
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
```

```
                reject;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Group Member GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.101.1/24
set interfaces ge-0/0/1 unit 0 description To_KeySrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.101.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
```

```
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then reject
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set routing-options static route 10.18.102.0/24 next-hop 10.18.101.254
set routing-options static route 10.18.103.0/24 next-hop 10.18.101.254
set routing-options static route 10.18.104.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.101.254
set routing-options static route 10.10.100.0/24 next-hop 10.18.101.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy KeySrv mode main
```

```
set security group-vpn member ike policy KeySrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy KeySrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway KeySrv ike-policy KeySrv
set security group-vpn member ike gateway KeySrv server-address 10.10.100.1
set security group-vpn member ike gateway KeySrv local-address 10.18.101.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway KeySrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_LAN
user@host# set ge-0/0/0 unit 0 family inet address 172.16.101.1/24
user@host# set ge-0/0/1 unit 0 description To_KeySrv
user@host# set ge-0/0/1 unit 0 family inet address 10.18.101.1/24
[edit security zones security-zone LAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
```

```
user@host# set then log session-init
[edit security policies from-zone WAN to-zone LAN
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set then log session-init
[edit security policies]
user@host# set global policy 1000 match source-address any
user@host# set global policy 1000 match destination-address any
user@host# set global policy 1000 match application any
user@host# set global policy 1000 match from-zone any
user@host# set global policy 1000 match to-zone any
user@host# set global policy 1000 match then reject
user@host# set global policy 1000 match then log session-init
user@host# set global policy 1000 match then count
user@host# set default-policy deny-all
```

2. Configure the static routes.

```
[edit routing-options]
user@host# set static route 10.18.102.0/24 next-hop 10.18.101.254
user@host# set static route 10.18.103.0/24 next-hop 10.18.101.254
user@host# set static route 10.18.104.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.104.0/24 next-hop 10.18.101.254
user@host# set static route 10.10.100.0/24 next-hop 10.18.101.254
```

3. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm sha-256
user@host# set dh-group group14
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy  KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
```

```
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.101.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

5. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.101.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_KeySrv;
        family inet {
            address 10.18.101.1/24;
```

```
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 10.18.102.0/24 next-hop 10.18.101.254;
    route 10.18.103.0/24 next-hop 10.18.101.254;
    route 10.18.104.0/24 next-hop 10.18.101.254;
    route 172.16.101.0/24 next-hop 10.18.101.254;
    route 172.16.102.0/24 next-hop 10.18.101.254;
    route 172.16.103.0/24 next-hop 10.18.101.254;
    route 172.16.104.0/24 next-hop 10.18.101.254;
    route 10.10.100.0/24 next-hop 10.18.101.254;
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway KeySrv {
                ike-policy KeySrv;
                server-address 10.10.100.1;
                local-address 10.18.101.1;
            }
        }
        ipsec {
```

```
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
            }
        }
    }
}
ipsec-policy {
    from-zone LAN to-zone WAN {
        ipsec-group-vpn GROUP_ID-0001;
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
```

```
        }
    }
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                reject;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
```

```
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Group Member GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.102.1/24
set interfaces ge-0/0/1 unit 0 description To_KeySrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.102.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
```

```
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then reject
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set routing-options static route 10.18.101.0/24 next-hop 10.18.102.254
set routing-options static route 10.18.103.0/24 next-hop 10.18.102.254
set routing-options static route 10.18.104.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.102.254
set routing-options static route 10.10.100.0/24 next-hop 10.18.102.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy KeySrv mode main
set security group-vpn member ike policy KeySrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy KeySrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway KeySrv ike-policy KeySrv
set security group-vpn member ike gateway KeySrv server-address 10.10.100.1
set security group-vpn member ike gateway KeySrv local-address 10.18.102.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway KeySrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_LAN
user@host# set ge-0/0/0 unit 0 family inet address 172.16.102.1/24
user@host# set ge-0/0/1 unit 0 description To_KeySrv
user@host# set ge-0/0/1 unit 0 family inet address 10.18.101.1/24
[edit security zones security-zone LAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set then log session-init
[edit security policies from-zone WAN to-zone LAN
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set then log session-init
[edit security policies]
user@host# set global policy 1000 match source-address any
user@host# set global policy 1000 match destination-address any
user@host# set global policy 1000 match application any
user@host# set global policy 1000 match from-zone any
user@host# set global policy 1000 match to-zone any
user@host# set global policy 1000 match then reject
user@host# set global policy 1000 match then log session-init
user@host# set global policy 1000 match then count
user@host# set default-policy deny-all
```

2. Configure the static routes.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.18.102.254
user@host# set static route 10.18.103.0/24 next-hop 10.18.102.254
user@host# set static route 10.18.104.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.104.0/24 next-hop 10.18.102.254
user@host# set static route 10.10.100.0/24 next-hop 10.18.102.254
```

3. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm sha-256
user@host# set dh-group group14
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy  KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.102.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

5. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.102.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_KeySrv;
        family inet {
            address 10.18.102.1/24;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 10.18.101.0/24 next-hop 10.18.102.254;
    route 10.18.103.0/24 next-hop 10.18.102.254;
    route 10.18.104.0/24 next-hop 10.18.102.254;
    route 172.16.101.0/24 next-hop 10.18.102.254;
    route 172.16.102.0/24 next-hop 10.18.102.254;
    route 172.16.103.0/24 next-hop 10.18.102.254;
    route 172.16.104.0/24 next-hop 10.18.102.254;
    route 10.10.100.0/24 next-hop 10.18.102.254;
```

```
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway KeySrv {
                ike-policy KeySrv;
                server-address 10.10.100.1;
                local-address 10.18.102.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
            }
        }
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
```

```
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                reject;
                log {
                    session-init;
                }
                count;
            }
        }
```

```
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Group Member GM-0003 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.103.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.103.1/24
set interfaces ms-0/2/0 unit 0 family inet
set routing-options static route 10.18.101.0/24 next-hop 10.18.103.254
set routing-options static route 10.18.102.0/24 next-hop 10.18.103.254
set routing-options static route 10.18.104.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.103.254
set routing-options static route 10.10.100.0/24 next-hop 10.18.103.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy KeySrv mode main
set security group-vpn member ike policy KeySrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy KeySrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway KeySrv ike-policy KeySrv
set security group-vpn member ike gateway KeySrv server-address 10.10.100.1
set security group-vpn member ike gateway KeySrv local-address 10.18.103.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway KeySrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
```

```
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
set firewall family inet service-filter GroupVPN-KS term inbound-ks from destination-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
```

Step-by-Step Procedure

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.103.1/24
user@host# set xe-0/0/2 unit 0 family inet address 172.16.103.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure routing.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.18.103.254
user@host# set static route 10.18.102.0/24 next-hop 10.18.103.254
user@host# set static route 10.18.104.0/24 next-hop 10.18.103.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.103.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.103.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.103.254
```

```
user@host# set static route 172.16.104.0/24 next-hop 10.18.103.254
user@host# set static route 10.10.100.0/24 next-hop 10.18.103.254
```

3. Configure IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256 ]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.103.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

5. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from destination-address 10.10.100.1/32
user@host# set term inbound-ks from source-address 10.10.100.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.10.100.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

6. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
            }
            address 10.18.103.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.103.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
```

```
}
[edit]
user@host# show routing-options
static {
    route 10.18.101.0/24 next-hop 10.18.103.254;
    route 10.18.102.0/24 next-hop 10.18.103.254;
    route 10.18.104.0/24 next-hop 10.18.103.254;
    route 172.16.101.0/24 next-hop 10.18.103.254;
    route 172.16.102.0/24 next-hop 10.18.103.254;
    route 172.16.103.0/24 next-hop 10.18.103.254;
    route 172.16.104.0/24 next-hop 10.18.103.254;
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway KeySrv {
                ike-policy KeySrv;
                local-address 10.18.103.1;
                server-address 10.10.101.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
```

```
    }
}
[edit]
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                destination-address {
                    10.10.100.1/32;
                }
                source-address {
                    10.10.100.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
                destination-address {
                    10.10.100.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
```

```
        }
    }
}
```

Configuring Group Member GM-0004 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.104.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.104.1/24
set interfaces ms-0/2/0 unit 0 family inet
set routing-options static route 10.18.101.0/24 next-hop 10.18.104.254
set routing-options static route 10.18.102.0/24 next-hop 10.18.104.254
set routing-options static route 10.18.103.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.104.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
```

```
set security group-vpn member ike gateway SubSrv local-address 10.18.104.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
set firewall family inet service-filter GroupVPN-KS term inbound-ks from destination-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
```

Step-by-Step Procedure

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.104.1/24
user@host# set xe-0/0/2 unit 0 family inet address 172.16.104.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure routing.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.18.104.254
user@host# set static route 10.18.102.0/24 next-hop 10.18.104.254
user@host# set static route 10.18.103.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.104.0/24 next-hop 10.18.104.254
```

3. Configure IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256 ]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.104.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

5. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from destination-address 10.10.101.1/32
user@host# set term inbound-ks from source-address 10.10.101.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.17.101.1/32
user@host# set term outbound-ks from destination-address 10.17.102.1/32
user@host# set term outbound-ks from destination-address 10.17.103.1/32
user@host# set term outbound-ks from destination-address 10.17.104.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

6. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
```

```
            }
            address 10.18.104.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.104.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.18.101.0/24 next-hop 10.18.104.254;
    route 10.18.102.0/24 next-hop 10.18.104.254;
    route 10.18.103.0/24 next-hop 10.18.104.254;
    route 172.16.101.0/24 next-hop 10.18.104.254;
    route 172.16.102.0/24 next-hop 10.18.104.254;
    route 172.16.103.0/24 next-hop 10.18.104.254;
    route 172.16.104.0/24 next-hop 10.18.104.254;
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
```

```
            }
            gateway KeySrv {
                ike-policy KeySrv;
                local-address 10.18.104.1;
                server-address 10.17.101.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
    }
}
[edit]
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                destination-address {
                    10.10.100.1/32;
                }
                source-address {
                    10.10.100.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
```

```
                destination-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Group Member Registration | **705**
- Verifying That Group Keys Are Distributed | **706**
- Verifying Group VPN SAs on the Group Server | **706**
- Verifying Group VPN SAs on Group Members | **707**
- Verifying IPsec SAs on the Group Server | **709**
- Verifying IPsec SAs on the Group Members | **709**
- Verifying Group Policies (SRX Series Firewall or vSRX Virtual Firewall Group Members Only) | **712**

Confirm that the configuration is working properly.

Verifying Group Member Registration

Purpose

Verify that group members are registered on the server.

Action

From operational mode, enter the `show security group-vpn server registered-members` and `show security group-vpn server registered-members detail` commands on the server.

```
user@host> show security group-vpn server registered-members
Group: GROUP_ID-0001, Group Id: 1
  Total number of registered members: 2
  Member Gateway                  Member IP       Last Update              Vsys
  GM-0001                         10.18.101.1     Thu Nov 19 2015 16:31:09 root
  GM-0003                         10.18.103.1     Thu Nov 19 2015 16:29:47 root
```

```
user@host> show security group-vpn server registered-members detail
GGroup: GROUP_ID-0001, Group Id: 1
  Total number of registered members: 2

  Member gateway: GM-0001, Member IP: 10.18.101.1, Vsys: root
  Last Update: Thu Nov 19 2015 16:31:09
  Stats:
      Pull Succeeded               : 2
      Pull Failed                  : 0
      Push Sent                    : 0
      Push Acknowledged            : 0
      Push Unacknowledged          : 0

  Member gateway: GM-0003, Member IP: 10.18.103.1, Vsys: root
  Last Update: Thu Nov 19 2015 16:29:47
  Stats:
      Pull Succeeded               : 1
      Pull Failed                  : 0
      Push Sent                    : 0
      Push Acknowledged            : 0
      Push Unacknowledged          : 0
```

Verifying That Group Keys Are Distributed

Purpose

Verify that group keys are distributed to members.

Action

From operational mode, enter the `show security group-vpn server statistics` command on the group server.

```
user@host> show security group-vpn server statistics
Group: GROUP_ID-0001, Group Id: 1
  Stats:
      Pull Succeeded                : 4
      Pull Failed                   : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

Verifying Group VPN SAs on the Group Server

Purpose

Verify Group VPN SAs on the group server.

Action

From operational mode, enter the `show security group-vpn server kek security-associations` and `show security group-vpn server kek security-associations detail` commands on the group server.

```
user@host> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
738879  1206      a471513492db1e13  24045792a4b3dd64  1
```

```
user@host> show security group-vpn server kek security-associations detail
Index 738879, Group Name: GROUP_ID-0001, Group Id: 1
Initiator cookie: a471513492db1e13, Responder cookie: 24045792a4b3dd64
```

```
Authentication method: RSA
Lifetime: Expires in 1204 seconds, Activated
Rekey in 694 seconds
  Algorithms:
   Sig-hash             : sha256
   Encryption           : aes256-cbc
  Traffic statistics:
   Input  bytes  :                  0
   Output bytes  :                  0
   Input  packets:                  0
   Output packets:                  0
  Server Member Communication: Unicast
  Retransmission Period: 10, Number of Retransmissions: 2
  Group Key Push sequence number: 0

PUSH negotiations in progress: 0
```

Verifying Group VPN SAs on Group Members

Purpose

Verify Group VPN SAs on the group members.

Action

From operational mode, enter the `show security group-vpn member kek security-associations` and `show security group-vpn member kek security-associations detail` commands on the SRX Series Firewall or vSRX Virtual Firewall group member.

```
user@host> show security group-vpn member kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5455810 10.10.100.1     1093      a471513492db1e13  24045792a4b3dd64  1
```

```
user@host> show security group-vpn member kek security-associations detail
  Index 5455810, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.10.100.1
  Initiator cookie: a471513492db1e13, Responder cookie: 24045792a4b3dd64
  Lifetime: Expires in 1090 seconds
```

```
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-sha256-128
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Stats:
      Push received            :   0
      Delete received          :   0
```

From operational mode, enter the `show security group-vpn member kek security-associations` and `show security group-vpn member kek security-associations detail` commands on the MX Series group member.

```
user@host> show security group-vpn member kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
488598  10.10.100.1     963       a471513492db1e13  24045792a4b3dd64  1
```

```
user@host> show security group-vpn member kek security-associations detail
  Index 488598, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.10.100.1
  Initiator cookie: a471513492db1e13, Responder cookie: 24045792a4b3dd64
  Lifetime: Expires in 961 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-sha256-128
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Stats:
      Push received            :   0
      Delete received          :   0
```

Verifying IPsec SAs on the Group Server

Purpose

Verify IPsec SAs on the group server.

Action

From operational mode, enter the `show security group-vpn server ipsec security-associations` and `show security group-vpn server ipsec security-associations detail` commands on the group server.

```
user@host> show security group-vpn server ipsec security-associations
Group: GROUP_ID-0001, Group Id: 1
  Total IPsec SAs: 1
  IPsec SA          Algorithm        SPI              Lifetime
  GROUP_ID-0001     ESP:aes-256/sha256 1c548e4e       1156
```

```
user@host> show security group-vpn server ipsec security-associations detail
Group: GROUP_ID-0001, Group Id: 1
Total IPsec SAs: 1
  IPsec SA: GROUP_ID-0001
    Protocol: ESP, Authentication: sha256, Encryption: aes-256
    Anti-replay: D3P enabled
    SPI: 1c548e4e
    Lifetime: Expires in 1152 seconds, Activated
    Rekey in 642 seconds
    Policy Name: 1
      Source: 172.16.0.0/12
      Destination: 172.16.0.0/12
      Source Port: 0
      Destination Port: 0
      Protocol: 0
```

Verifying IPsec SAs on the Group Members

Purpose

Verify IPsec SAs on the group members.

Action

From operational mode, enter the `show security group-vpn member ipsec security-associations` and `show security group-vpn member ipsec security-associations detail` commands on the SRX Series Firewall or vSRX Virtual Firewall group member.

```
user@host> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>49152 10.10.100.1    848   ESP:aes-256/sha256-128 1c548e4e 1073/ unlim 1 root
```

```
user@host> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.10.100.1
  Group Id: 1
  Routing Instance: default
  Recovery Probe: Enabled
  DF-bit: clear
  Stats:
      Pull Succeeded             :   4
      Pull Failed                :   3
      Pull Timeout               :   3
      Pull Aborted               :   0
      Push Succeeded             :   6
      Push Failed                :   0
      Server Failover            :   0
      Delete Received            :   0
      Exceed Maximum Keys(4)     :   0
      Exceed Maximum Policies(10):   0
      Unsupported Algo           :   0
  Flags:
      Rekey Needed:   no

    List of policies received from server:
    Tunnel-id: 49152
      Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
      Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

      Direction: bi-directional, SPI: 1c548e4e
      Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
      Hard lifetime: Expires in 1070 seconds, Activated
```

```
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 931 seconds
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled
```

From operational mode, enter the `show security group-vpn member ipsec security-associations` and `show security group-vpn member ipsec security-associations detail` commands on the MX Series group member.

```
user@host> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>10001 10.10.100.1    848   ESP:aes-256/sha256-128 1c548e4e 947/ unlim 1 root
```

```
user@host> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.10.100.1
  Group Id: 1
  Rule Match Direction: output,  Tunnel-MTU: 1400
  Routing Instance: default
  DF-bit: clear
  Stats:
      Pull Succeeded            :   2
      Pull Failed               :   0
      Pull Timeout              :   1
      Pull Aborted              :   0
      Push Succeeded            :   2
      Push Failed               :   0
      Server Failover           :   0
      Delete Received           :   0
      Exceed Maximum Keys(4)    :   0
      Exceed Maximum Policies(1):   0
      Unsupported Algo          :   0
  Flags:
      Rekey Needed:   no

    List of policies received from server:
    Tunnel-id: 10001
      Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
      Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

      Direction: bi-directional, SPI: 1c548e4e
```

```
      Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
      Hard lifetime: Expires in 945 seconds, Activated
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 840 seconds
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled
```

Verifying Group Policies (SRX Series Firewall or vSRX Virtual Firewall Group Members Only)

Purpose

Verify group policies on SRX Series Firewall or vSRX Virtual Firewall group members.

Action

From operational mode, enter the `show security group-vpn member policy` command on the group member.

```
user@host> show security group-vpn member policy
Group VPN Name: GROUP_ID-0001, Group Id: 1
From-zone: LAN, To-zone: WAN
 Tunnel-id: 49152, Policy type: Secure
  Source      : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>
  Destination : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>

 Tunnel-id: 63488, Policy type: Fail-close
  Source      : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
  Destination : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
```

SEE ALSO

- Configuring Group VPNs in Group VPNv2 on Routing Device

Example: Configuring Group VPNv2 Server-Member Communication for Unicast Rekey Messages

IN THIS SECTION

- Requirements | 713
- Overview | 713
- Configuration | 714
- Verification | 714

This example shows how to enable the server to send unicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members.

Requirements

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation.
- Configure the group server and members for IPsec SA.
- Configure the group g1 on the group server.

Overview

In this example, you specify the following server-member communication parameters for group g1:

- The server sends unicast rekey messages to group members.
- aes-128-cbc is used to encrypt traffic between the server and members.
- sha-256 is used for member authentication.

Default values are used for KEK lifetime and retransmissions.

Configuration

IN THIS SECTION

- Procedure | 714

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-member communication:

1. Set the communications type.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communications-type unicast
```

2. Set the encryption algorithm.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm aes-128-cbc
```

3. Set the member authentication.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha-256
```

Verification

To verify the configuration is working properly, enter the `show security group-vpn server group g1 server-member-communication` command.

SEE ALSO

Example: Configuring Group VPNv1 Server and Members | 607

Example: Configuring Group VPNv1 with Server-Member Colocation | 641

RELATED DOCUMENTATION

VPN Session Affinity | 1232

Group VPNv2 Server Clusters

SUMMARY

Read this topic to learn about Group VPNv2 server clusters.

IN THIS SECTION

- Understanding Group VPNv2 Server Clusters | 716
- Understanding Group VPNv2 Server Cluster Limitations | 720
- Understanding Group VPNv2 Server Cluster Messages | 721
- Understanding Configuration Changes with Group VPNv2 Server Clusters | 723
- Migrating a Standalone Group VPNv2 Server to a Group VPNv2 Server Cluster | 727
- Example: Configuring a Group VPNv2 Server Cluster and Members | 728

Group VPNv2 server cluster provides group controller/key server (GCKS) redundancy, so there is no single point of failure for the entire group VPN network.

Understanding Group VPNv2 Server Clusters

IN THIS SECTION

- Root-Server and Sub-Servers | 716
- Group Member Registration with Server Clusters | 718
- Dead Peer Detection | 719
- Load Balancing | 719

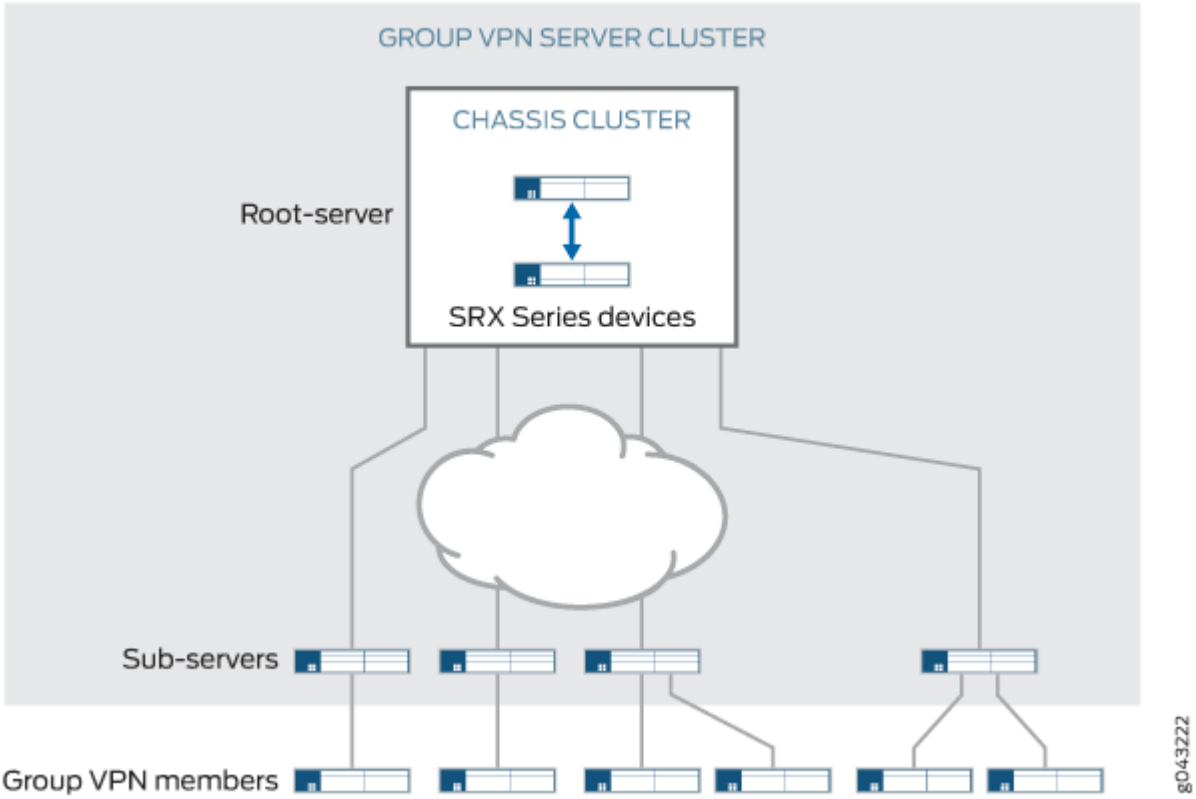
In the Group Domain of Interpretation (GDOI) protocol, the group controller/key server (GCKS) manages Group VPN security associations (SAs), and generates encryption keys and distributes them to group members. Group members encrypt traffic based on the group SAs and keys provided by the GCKS. If the GCKS fails, group members cannot register or obtain keys. A *Group VPNv2 server cluster* provides GCKS redundancy so there is no single point of failure for the entire group VPN network. Group VPNv2 server clusters can also provide load balancing, scaling, and link redundancy.

All servers in a Group VPNv2 server cluster must be supported on SRX Series Firewalls or vSRX Virtual Firewall instances. Group VPNv2 server clusters are a Juniper Networks proprietary solution and have no interoperability with other vendor's GCKS.

Root-Server and Sub-Servers

A Group VPNv2 server cluster consists of one root-server with up to four connected sub-servers. All servers in the cluster share the same SA and encryption keys that are distributed to Group VPNv2 members. Servers in the cluster can be located at different sites, as shown in Figure 45 on page 717.

Figure 45: Group VPNv2 Server Cluster



Messages between servers in the cluster are encrypted and authenticated by IKE SAs. The root-server is responsible for generating and distributing encryption keys to sub-servers; because of this responsibility, we recommend that the root-server be configured as a chassis cluster. Sub-servers are single devices and cannot be chassis clusters. Sub-servers must be able to connect to the root-server, although direct links between sub-servers are not necessary.

If a sub-server loses its connection to the root-server, no further connection to the sub-server from group members are allowed and SAs are deleted. Therefore, we recommend that you use a different link to connect each sub-server to the root-server.

Group VPNv2 server clusters are configured with the `server-cluster` statements at the [`edit security group-vpn server` *`group-name`*] hierarchy level. The following values must be configured for each server in a cluster:

- The server role—Specify either `root-server` or `sub-server`. A given server can be part of multiple Group VPNv2 server clusters, but it must have the same server role in all clusters. A server cannot be configured with the root-server role in one group and the sub-server role in another group.

  You must ensure that there is only one root-server at any time for a Group VPNv2 server cluster.

- IKE gateway—Specify the name of an IKE gateway configured at the [`edit security group-vpn server ike`] hierarchy level. For a root-server, the IKE gateway must be a sub-server in the cluster; up to four sub-servers can be specified. For sub-servers, the IKE gateway must be the root-server.

  The root-server and sub-servers must be configured with `dead-peer-detection always-send` and cannot be configured for a dynamic (unspecified) IP address. Group members are not configured with dead peer detection.

The Group VPNv2 configuration must be the same on each sub-server in a given group.

Each sub-server in the Group VPNv2 server cluster operates as a normal GCKS for registering and deleting members. Upon successful member registration, the registering server is responsible for sending updates to the member. For a given group, you can configure the maximum number of Group VPNv2 members that can be accepted by each sub-server; this number must be the same on all sub-servers in the cluster. A sub-server stops responding to registration requests by new members when it reaches the configured maximum number of Group VPNv2 members. See "Load Balancing" on page 719.

Group Member Registration with Server Clusters

Group members can register with any server in the Group VPNv2 server cluster for a given group, however we recommend that members only connect to sub-servers and not the root-server. Up to four server addresses can be configured on each group member. The server addresses configured on group members can be different. In the example shown below, group member A is configured for sub-servers 1 through 4, while member B is configured for sub-servers 4 and 3:

	Group member A:	Group member B:
Server addresses:	Sub-server 1 Sub-server 2 Sub-server 3 Sub-server 4	Sub-server 4 Sub-server 3

The order that the server addresses is configured on a member is important. A group member attempts to register with the first configured server. If registration with a configured server is not successful, the group member tries to register with the next configured server.

Each server in a Group VPNv2 server cluster operates as a normal GCKS for registering and deleting members. Upon successful registration, the registering server is responsible for sending updates to the member via `groupkey-push` exchanges. For a given group, you can configure the maximum number of group members that can be accepted by each server, however this number must be the same on all servers in

the cluster for a given group. Upon reaching the configured maximum number of group members, a server stops responding to registration requests by new members. See "Load Balancing" on page 719 for additional information.

Dead Peer Detection

To verify the availability of peer servers in a Group VPNv2 server cluster, each server in the cluster must be configured to send dead peer detection (DPD) requests regardless of whether there is outgoing IPsec traffic to the peer. This is configured with the `dead-peer-detection always-send` statement at the [`edit security group-vpn server ike gateway` *`gateway-name`*] hierarchy level.

An active server in a Group VPNv2 server cluster sends DPD probes to the IKE gateway(s) configured in the server cluster. DPD should not be configured for a group because multiple groups can share the same peer server IKE gateway configuration. When DPD detects that a server is down, the IKE SA with that server is deleted. All groups mark the server as inactive and DPD to the server is stopped.

DPD should not be configured for the IKE gateway on group members.

When DPD marks the root-server as inactive, the sub-servers stop responding to new group member requests however existing SAs for current group members remain active. An inactive sub-server does not send deletes to group members because the SAs could be still valid and group members can continue using existing SAs.

If an IKE SA expires while a peer server is still active, DPD triggers IKE SA negotiation. Because both root-servers and sub-servers can trigger IKE SAs through DPD, simultaneous negotiation might result in multiple IKE SAs. No impact on server-cluster functionality is expected in this case.

Load Balancing

Load balancing in the Group VPNv2 server cluster can be achieved by configuring the right `member-threshold` value for the group. When the number of members registered on a server exceeds the `member-threshold` value, subsequent member registration on that server is rejected. The member registration fails over to the next server configured on the group member until it reaches a server whose `member-threshold` is not yet reached.

There are two restrictions on configuring the `member-threshold`:

- For a given group, the same `member-threshold` value must be configured on the root-server and all sub-servers in a group server cluster. If the total number of members in the group exceeds the configured `member-threshold` value, then a `groupkey-pull` registration initiated by a new member is rejected (the server does not send a response).
- A server can support members in multiple groups. Each server has a maximum number of group members that it can support. If a server reaches the maximum number of members it can support,

then a groupkey-pull registration initiated by a new member is rejected even if the member-threshold value of a specific group has not been reached.

There is no member synchronization among servers in the cluster. The root-server does not have information about the number of registered members on sub-servers. Each sub-server can only show its own registered members.

SEE ALSO

Group VPNv2 Overview | 655

Understanding Group VPNv2 Server Cluster Limitations

Note the following caveats when configuring Group VPNv2 server clusters:

- Certificate authentication is not supported for server authentication; only preshared keys can be configured.
- There is no configuration synchronization between servers in the Group VPNv2 server cluster.
- When enabling a Group VPNv2 server cluster, configuration must be done on the root-server first and then on the sub-servers. Until the configuration is manually synchronized among the servers, traffic loss can be expected during the configuration change.
- In certain corner cases, the SAs on Group VPNv2 members can be out of sync. Group VPN members can synchronize SAs by getting a new key through a groupkey-pull exchange. You can manually clear SAs on a Group VPNv2 member with the clear security group-vpn member ipsec security-associations or clear security group-vpn member group commands to help speed recovery.
- The Group VPNv2 server cluster does not support ISSU.
- If the last groupkey-pull message is lost during a Group VPNv2 member's registration, a server might consider the member to be a registered member even though the member might fail over to the next server in the server cluster. In this case, the same member might appear to be registered on multiple servers. If the total member-threshold on all servers equals the total number of deployed members, subsequent group members might fail to register.

Note the following caveats for chassis cluster operations on the root-server:

- No statistics are preserved.
- No negotiation data or state is saved. If a root-server chassis cluster failover occurs during a groupkey-pull or groupkey-push negotiation, the negotiation is not restarted after the failover.

- If both chassis cluster nodes of a root-server go down during a rekey of an encryption key, some Group VPNv2 members might receive the new key while other members do not. Traffic might be impacted. Manually clearing SAs on a Group VPNv2 member with the `clear security group-vpn member ipsec security-associations` or `clear security group-vpn member group` commands might help speed up recovery when the root-server becomes reachable.
- In a large-scale environment, RG0 failover on the root-server might take time. If the DPD interval and threshold on a sub-server are configured with small values, it can result in the sub-server marking the root-server as inactive during an RG0 failover. Traffic might be impacted. We recommend that you configure the IKE gateway for the sub-server with a DPD `interval * threshold` value larger than 150 seconds.

Understanding Group VPNv2 Server Cluster Messages

IN THIS SECTION

- Cluster Exchanges | **721**
- Cluster-Init Exchanges | **722**
- Cluster-Update Messages | **723**

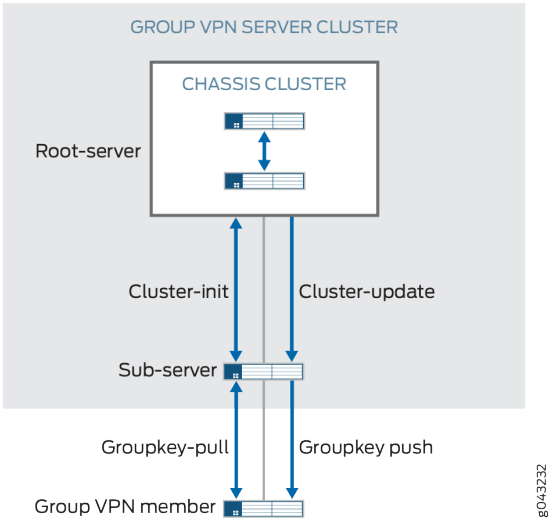
All messages between servers in a Group VPNv2 server cluster are encrypted and authenticated by an IKE security association (SA). Each sub-server initiates an IKE SA with the root-server; this IKE SA must be established before messages can be exchanged between the servers.

This section describes the messages exchanged between the root-server and sub-servers.

Cluster Exchanges

Figure 46 on page 722 shows the basic messages exchanged between the Group VPNv2 server cluster and Group VPNv2 members.

Figure 46: Group VPNv2 Server Cluster Messages



Cluster-Init Exchanges

A sub-server launches a cluster initialization (`cluster-init`) exchange with the root-server to obtain SA and encryption key information. The root-server responds by sending current SA information to the sub-server through the `cluster-init` exchange.

Sub-servers can then respond to registration requests from Group VPNv2 members through a `groupkey-pull` exchange. The `groupkey-pull` exchange allows a Group VPNv2 member to request SAs and keys shared by the group from a sub-server.

Sub-servers start a `cluster-init` exchange with the root-server when:

- The root-server is considered inactive. This is the initial assumed state of the root-server. If there is no IKE SA between the root-server and the sub-server, the sub-server initiates an IKE SA with the root-server. After a successful `cluster-init` exchange, the sub-server obtains information on SAs and marks the root-server as active.
- The soft lifetime of the SA has expired.
- A `cluster-update` message is received to delete all SAs.
- There are group configuration changes.

If the `cluster-init` exchange fails, the sub-server retries the exchange with the root-server every 5 seconds.

Cluster-Update Messages

The groupkey-push exchange is a single rekey message that allows a group controller/key server (GCKS) to send group SAs and keys to members before existing group SAs expire and to update group membership. Rekey messages are unsolicited messages sent from the GCKS to members

Upon generating new encryption keys for an SA, the root-server sends SA updates to all active sub-servers through a cluster-update message. After receiving a cluster-update from the root-server, the sub-server installs the new SA and sends the new SA information through a groupkey-push to its registered group members.

A cluster-update message sent from the root-server requires an acknowledgement from the sub-server. If there is no acknowledgement received from a sub-server, the root-server retransmits the cluster-update at the configured retransmission period (the default is 10 seconds). The root-server does not retransmit if dead peer detection (DPD) indicates that the sub-server is unavailable. If a sub-server fails to update SA information after receiving a cluster-update, it does not send an acknowledgement and the root-server retransmits the cluster-update message.

If the soft lifetime of an SA expires before a new SA is received from the root-server, the sub-server sends a cluster-init message to the root-server to get all SAs and does not send a groupkey-push message to its members until it has a new update. If the hard lifetime of an SA expires on the sub-server before it receives a new SA, the sub-server marks the root-server inactive, deletes all registered group members, and continues to send cluster-init messages to the root-server.

A cluster-update message can be sent to delete an SA or a group member; this can be the result of a clear command or a configuration change. If a sub-server receives a cluster-update message to delete an SA, it sends a groupkey-push delete message to its group members and deletes the corresponding SA. If all SAs for a group are deleted, the sub-server initiates a cluster-init exchange with the root-server. If all registered members are deleted, the sub-server deletes all locally registered members.

Understanding Configuration Changes with Group VPNv2 Server Clusters

Group VPNv2 server clusters behave differently from standalone Group VPNv2 servers when there are configuration changes that result in new encryption keys and changes to security associations (SAs). The root-server sends SA updates or deletions to sub-servers through cluster-update messages. The sub-servers then send groupkey-push messages to members. Sub-servers cannot send delete messages to group members without first receiving delete messages from the root-server.

All configuration changes must be made on the root-server first and then on sub-servers to ensure that group members receive updates or deletions as expected. Until configuration is synchronized between the servers in the Group VPNv2 server cluster, traffic loss can be expected.

Table 90 on page 724 describes the effects of various configuration changes on Group VPNv2 servers.

Table 90: Effects of Configuration Changes on Group VPNv2 Servers

Configuration Change	Standalone Group VPNv2 Server Action	Group VPNv2 Server Cluster Action	
		Root-server	Sub-server
Change IKE proposal, policy, or gateway	Delete the IKE SA for the affected gateway. For IKE proposal, policy, or gateway deletions, delete the registered members for the affected gateway.		
Change IPsec proposal	Changes take effect after the traffic encryption key (TEK) rekey.		
Group changes:			
Delete group name	Send “delete all” to group members. Delete all IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group.	Send “delete all” to sub-servers. Delete all keys in the group immediately. Mark all peers inactive. Delete sub-server IKE SAs. Delete all member IKE SAs.	Delete all member IKE SAs. Delete all keys in the group immediately. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs.
Change ID	Send “delete all” to all members. Delete all IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group. Generate new keys according to the configuration.	Send ”delete all” to sub-servers. Delete all member IKE SAs in the group. Delete all keys in the group immediately. Mark all peers inactive. Delete all peer server IKE SAs. Generate new keys according to the configuration.	Delete all member IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs. Initiate new `cluster-init` exchange.
Add or delete IKE gateway	No changes for additions. For deletions, delete the IKE SA and registered members for the affected gateway.		
Add or change anti-replay time window	New value takes effect after the TEK rekey.		

Table 90: Effects of Configuration Changes on Group VPNv2 Servers *(Continued)*

Configuration Change	Standalone Group VPNv2 Server Action	Group VPNv2 Server Cluster Action	
		Root-server	Sub-server
Add or change no anti-replay	New value takes effect after the TEK rekey.		
Server-member communication changes:			
Add	Delete all registered members. Generate key encryption key (KEK) SA.	Generate KEK SA. Send new KEK SA to sub-server. Delete all member IKE SAs.	Delete all registered members.
Change	New value takes effect after KEK rekey.		
Delete	Send delete to delete all KEK SAs. Delete KEK SA.	Send delete to sub-servers. Delete KEK SA. Delete all member IKE SAs.	Delete KEK SA.
IPsec SA:			
Add	Generate new TEK SA. Update the new TEK SA on members.	Generate new TEK SA. Send new TEK SA to sub-servers.	No action.
Change	New value takes effect after TEK rekey. If the match-policy changes, the current TEK is removed immediately and delete groupkey-push is sent because members need to be explicitly notified that this configuration is removed.	If the match-policy changes, send delete to sub-servers. Delete TEK immediately.	If the match-policy changes, delete TEK immediately.

Table 90: Effects of Configuration Changes on Group VPNv2 Servers *(Continued)*

Configuration Change	Standalone Group VPNv2 Server Action	Group VPNv2 Server Cluster Action	
		Root-server	Sub-server
Delete	Delete TEK immediately. Send delete to delete this TEK SA.	Send delete to sub-servers. Delete TEK immediately.	Delete TEK immediately.

Table 91 on page 726 describes the effects of changing Group VPNv2 server cluster configuration.

You must ensure that there is only one root-server in a server cluster at any time.

Table 91: Effects of Group VPNv2 Server Cluster Configuration Changes

Server Cluster Configuration Change	Group VPNv2 Server Cluster	
	Root-server	Sub-server
IKE proposal, policy, or gateway (cluster peer)	For additions, there is no change. For changes or deletions, delete the IKE SA for the affected peer.	
Server cluster:		
Add	None.	Send “delete all” to group members. Delete all member IKE SAs in the group. Delete all TEKs and KEKs immediately in the group. Delete all registered members in the group. Send `cluster-init` to root-server.
Change role You must ensure that there is only one root-server in a server cluster at any time.	Send “delete all” to sub-servers. Delete all member IKE SAs in the group. Delete all TEKs and KEKs immediately in the group. Mark all peers inactive. Delete all peer server IKE SAs. Send `cluster-init` to root-server.	Rekey TEK. Rekey KEK. Send new keys to sub-servers. Send new keys to members.

Table 91: Effects of Group VPNv2 Server Cluster Configuration Changes *(Continued)*

Server Cluster Configuration Change	Group VPNv2 Server Cluster	
	Root-server	Sub-server
Add peer	None.	
Delete peer	Mark peer inactive. Clear peer IKE SA.	Mark peer inactive. Clear KEK. Clear TEK. Clear peer IKE SA.
Change retransmission period	None.	
Delete server cluster	Send “delete all” to sub-servers. Delete all TEKs and KEKs immediately in the group. Mark all peers inactive. Delete all peer server IKE SAs. Generate new TEKs and KEKs according to the configuration.	Delete all member IKE SAs in the group. Delete all TEKs and KEKs immediately in the group. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs. Generate new TEK and KEK according to the configuration.

Migrating a Standalone Group VPNv2 Server to a Group VPNv2 Server Cluster

This section describes how to migrate a standalone Group VPNv2 server to a Group VPNv2 server cluster.

To migrate a standalone Group VPNv2 server to a root-server:

We highly recommend that the root-server be a chassis cluster.

1. Upgrade the standalone Group VPNv2 server to a chassis cluster. See Chassis Cluster User Guide for SRX Series Devices for more information

   A reboot is required during the upgrade of a standalone SRX Series Firewall to a chassis cluster node. Traffic loss is expected.
2. On the chassis cluster, add the Group VPNv2 server cluster root-server configuration. The configured server role for the cluster must be `root-server`.

   There should be no traffic loss among existing group members during the configuration change.

To add a sub-server to the Group VPNv2 server cluster:

1. On the root-server, configure both a Group VPNv2 server IKE gateway and a server cluster IKE gateway for the sub-server. SAs and existing member traffic should not be impacted.
2. On the sub-server, configure the server cluster. Remember that the Group VPNv2 configuration must be the same on each server in the cluster, with the exception of the Group VPNv2 server IKE gateways, the server role in the cluster, and the server cluster IKE gateway configurations. On the sub-server, the configured server role in the cluster must be `sub-server`. Configure a Group VPNv2 server IKE gateway and a server cluster IKE gateway for the root-server.

To delete a sub-server from the Group VPNv2 server cluster:

1. On the root-server, delete both the Group VPNv2 server IKE gateway and the server cluster IKE gateway configurations for the sub-server. SAs and existing member traffic should not be impacted.
2. Power off the sub-server.

SEE ALSO

Group VPNv2 Overview | 655

Example: Configuring a Group VPNv2 Server Cluster and Members

IN THIS SECTION

- Requirements | 728
- Overview | 729
- Configuration | 731
- Verification | 800

This example shows how to configure a Group VPNv2 server cluster to provide group controller/key server (GCKS) redundancy and scaling to Group VPNv2 group members.

Requirements

The example uses the following hardware and software components:

- Eight supported SRX Series Firewalls or vSRX Virtual Firewall instances running Junos OS Release 15.1X49-D30 or later that support Group VPNv2:
  - Two devices or instances are configured to operate as a chassis cluster. The chassis cluster operates as the root-server in the Group VPNv2 server cluster. The devices or instances must have the same software version and licenses.

    The root-server is responsible for generating and distributing encryption keys to sub-servers in the group VPN server cluster; because of this responsibility, we recommend that the root-server be a chassis cluster.
  - Four other devices or instances operate as sub-servers in the Group VPNv2 server cluster.
  - Two other devices or instances operate as Group VPNv2 group members.
- Two supported MX Series devices running Junos OS Release 15.1R2 or later that support Group VPNv2. These devices operate as Group VPNv2 group members.

A hostname, a root administrator password, and management access must be configured on each SRX Series Firewall or vSRX Virtual Firewall instance. We recommend that NTP also be configured on each device.

The configurations in this example focus on what is needed for Group VPNv2 operation, based on the topology shown in Figure 47 on page 731. Some configurations, such as interface, routing, or chassis cluster setups, are not included here. For example, Group VPNv2 operation requires a working routing topology that allows client devices to reach their intended sites throughout the network; this example does not cover the configuration of static or dynamic routing.

Overview

IN THIS SECTION

- Topology | **730**

In this example, the Group VPNv2 network consists of a server cluster and four members. The server cluster consists of a root-server and four sub-servers. Two of the members are SRX Series Firewalls or vSRX Virtual Firewall instances while the other two members are MX Series devices.

The group VPN SAs must be protected by a Phase 1 SA. Therefore, the group VPN configuration must include configuring IKE Phase 1 negotiations on the root-server, the sub-servers, and the group members. IKE configurations are described as follows.

On the root-server:

- The IKE policy `SubSrv` is used to establish Phase 1 SAs with each sub-server.
- An IKE gateway is configured with dead peer detection (DPD) for each sub-server.
- The server cluster role is `root-server` and each sub-server is configured as an IKE gateway for the server cluster.

The root-server should be configured to support chassis cluster operation. In the example, redundant Ethernet interfaces on the root-server connect to each of the sub-servers in the server cluster; the entire chassis cluster configuration is not shown.

On each sub-server:

- Two IKE policies are configured: `RootSrv` is used to establish a Phase 1 SA with the root-server, and `GMs` is used to establish Phase 1 SAs with each group member.

  Preshared keys are used to secure the Phase 1 SAs between the root-server and the sub-servers and between the sub-servers and the group members. Ensure that the preshared keys used are strong keys. On the sub-servers, the preshared key configured for the IKE policy `RootSrv` must match the preshared key configured on the root-server, and the preshared key configured for the IKE policy `GMs` must match the preshared key configured on the group members.
- An IKE gateway is configured with DPD for the root-server. In addition, an IKE gateway is configured for each group member.
- The server cluster role is `sub-server` and the root-server is configured as the IKE gateway for the server cluster.

On each group member:

- The IKE policy `SubSrv` is used to establish Phase 1 SAs with the sub-servers.
- The IKE gateway configuration includes the addresses for the sub-servers.

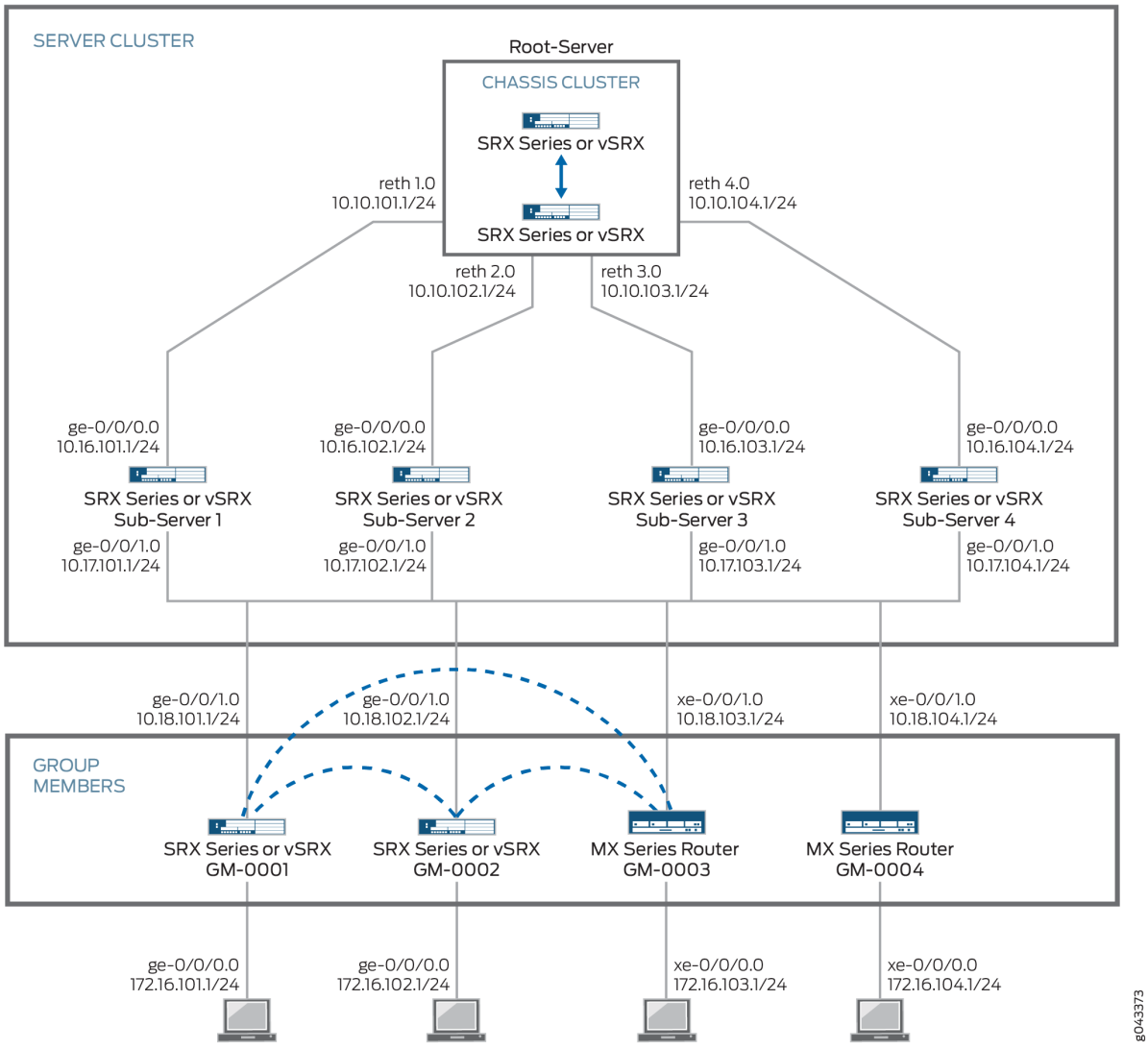
On SRX Series Firewalls or vSRX Virtual Firewall group members, an IPsec policy is configured for the group with the LAN zone as the from-zone (incoming traffic) and the WAN zone as the to-zone (outgoing traffic). A security policy is also needed to allow traffic between the LAN and WAN zones.

The same group identifier must be configured on both the group server and the group members. In this example, the group name is GROUP_ID-0001 and the group identifier is 1. The group policy configured on the server specifies that the SA and key are applied to traffic between subnetworks in the 172.16.0.0/12 range.

Topology

Figure 47 on page 731 shows the Juniper Networks devices to be configured for this example.

Figure 47: Group VPNv2 Server Cluster with SRX Series or vSRX Virtual Firewall and MX Series Members



Configuration

IN THIS SECTION

- Configuring the Root-Server | **732**
- Configuring Sub-Server 1 | **741**
- Configuring Sub-Server 2 | **749**

- Configuring Sub-Server 3 | **757**
- Configuring Sub-Server 4 | **766**
- Configuring GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **774**
- Configuring GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **781**
- Configuring GM-0003 (MX Series Device) | **788**
- Configuring GM-0004 (MX Series Device) | **794**

Configuring the Root-Server

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 description To_SubSrv01
set interfaces reth1 unit 0 family inet address 10.10.101.1/24
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth2 unit 0 description To_SubSrv02
set interfaces reth2 unit 0 family inet address 10.10.102.1/24
set interfaces reth3 redundant-ether-options redundancy-group 1
set interfaces reth3 unit 0 description To_SubSrv03
set interfaces reth3 unit 0 family inet address 10.10.103.1/24
set interfaces reth4 redundant-ether-options redundancy-group 1
set interfaces reth4 unit 0 description To_SubSrv04
set interfaces reth4 unit 0 family inet address 10.10.104.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces reth1.0
set security zones security-zone GROUPVPN interfaces reth2.0
set security zones security-zone GROUPVPN interfaces reth3.0
set security zones security-zone GROUPVPN interfaces reth4.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
```

```
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set chassis cluster reth-count 5
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy SubSrv mode main
set security group-vpn server ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy SubSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike gateway SubSrv01 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv01 address 10.16.101.1
set security group-vpn server ike gateway SubSrv01 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv01 local-address 10.10.101.1
set security group-vpn server ike gateway SubSrv02 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv02 address 10.16.102.1
set security group-vpn server ike gateway SubSrv02 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv02 local-address 10.10.102.1
set security group-vpn server ike gateway SubSrv03 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv03 address 10.16.103.1
set security group-vpn server ike gateway SubSrv03 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv03 local-address 10.10.103.1
set security group-vpn server ike gateway SubSrv04 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv04 address 10.16.104.1
set security group-vpn server ike gateway SubSrv04 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv04 local-address 10.10.104.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
```

```
set security group-vpn server group GROUP_ID-0001 server-cluster server-role root-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv01
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv02
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv03
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv04
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the root-server:

1. Configure security zones and security policies.

```
[edit interfaces]
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 description To_SubSrv01
user@host# set reth1 unit 0 family inet address 10.10.101.1/24
user@host# set reth2 redundant-ether-options redundancy-group 1
user@host# set reth2 unit 0 description To_SubSrv02
user@host# set reth2 unit 0 family inet address 10.10.102.1/24
user@host# set reth3 redundant-ether-options redundancy-group 1
user@host# set reth3 unit 0 description To_SubSrv03
```

```
user@host# set reth3 unit 0 family inet address 10.10.103.1/24
user@host# set reth4 redundant-ether-options redundancy-group 1
user@host# set reth4 unit 0 description To_SubSrv04
user@host# set reth4 unit 0 family inet address 10.10.104.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces reth1.0
user@host# set interfaces reth2.0
user@host# set interfaces reth3.0
user@host# set interfaces reth4.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the chassis cluster.

```
[edit chassis cluster]
user@host# set reth-count 5
user@host# set redundancy-group 1 node 0 priority 254
user@host# set redundancy-group 1 node 1 priority 1
user@host# set redundancy-group 0 node 0 priority 254
user@host# set redundancy-group 0 node 1 priority 1
```

3. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy SubSrv]
```

```
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike gateway SubSrv01]
user@host# set ike-policy SubSrv
user@host# set address 10.16.101.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.101.1
[edit security group-vpn server ike gateway SubSrv02]
user@host# set ike-policy SubSrv
user@host# set address 10.16.102.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.102.1
[edit security group-vpn server ike gateway SubSrv03]
user@host# set ike-policy SubSrv
user@host# set address 10.16.103.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.103.1
[edit security group-vpn server ike gateway SubSrv04]
user@host# set ike-policy SubSrv
user@host# set address 10.16.104.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.104.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

5. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role root-server
user@host# set server-cluster ike-gateway SubSrv01
user@host# set server-cluster ike-gateway SubSrv02
user@host# set server-cluster ike-gateway SubSrv03
user@host# set server-cluster ike-gateway SubSrv04
```

```
user@host# set server-cluster retransmission-period 10
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

6. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show chassis cluster`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv01;
        family inet {
            address 10.10.101.1/24;
        }
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv02;
```

```
        family inet {
            address 10.10.102.1/24;
        }
    }
}
reth3 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv03;
        family inet {
            address 10.10.103.1/24;
        }
    }
}
reth4 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv04;
        family inet {
            address 10.10.104.1/24;
        }
    }
}
[edit]
user@host# show chassis cluster
reth-count 5;
redundancy-group 1 {
    node 0 priority 254;
    node 1 priority 1;
}
redundancy-group 0 {
    node 0 priority 254;
    node 1 priority 1;
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
```

```
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv01 {
                ike-policy SubSrv;
                address 10.16.101.1;
                dead-peer-detection always-send;
                local-address 10.10.101.1;
            }
            gateway SubSrv02 {
                ike-policy SubSrv;
                address 10.16.102.1;
                dead-peer-detection always-send;
                local-address 10.10.102.1;
            }
            gateway SubSrv03 {
                ike-policy SubSrv;
                address 10.16.103.1;
                dead-peer-detection always-send;
                local-address 10.10.103.1;
            }
            gateway SubSrv04 {
                ike-policy SubSrv;
                address 10.16.104.1;
                dead-peer-detection always-send;
                local-address 10.10.104.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
```

```
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role root-server;
                ike-gateway SubSrv01;
                ike-gateway SubSrv02;
                ike-gateway SubSrv03;
                ike-gateway SubSrv04;
                retransmission-period 10;
            }
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
```

```
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            reth1.0;
            reth2.0;
            reth3.0;
            reth4.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.101.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
set interfaces ge-0/0/1 unit 0 family inet address 10.17.101.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
```

```
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.101.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
set security group-vpn server ike gateway RootSrv local-address 10.16.101.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.101.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.101.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.101.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.17.101.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
```

```
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_RootSrv
user@host# set ge-0/0/0 unit 0 family inet address 10.16.101.1/24
```

```
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.101.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
user@host# set ike-policy RootSrv
user@host# set address 10.10.101.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.101.1
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
```

```
user@host# set address 10.18.101.1
user@host# set local-address 10.17.101.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.101.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.101.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.101.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.101.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.101.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
```

```
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.101.1;
                dead-peer-detection always-send;
                local-address 10.16.101.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.101.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.101.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.101.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.101.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
```

```
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
```

```
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.102.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
set interfaces ge-0/0/1 unit 0 family inet address 10.17.102.1/24
```

```
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.102.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
set security group-vpn server ike gateway RootSrv local-address 10.16.102.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.102.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.102.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.102.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.17.102.1
```

```
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_RootSrv
```

```
user@host# set ge-0/0/0 unit 0 family inet address 10.16.102.1/24
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.102.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
user@host# set ike-policy RootSrv
user@host# set address 10.10.102.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.102.1
[edit security group-vpn server ike gateway GM-0001]
```

```
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.17.102.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.102.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.102.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.102.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.102.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.102.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
```

```
                encryption-algorithm aes-256-cbc;
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.102.1;
                dead-peer-detection always-send;
                local-address 10.16.102.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.102.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.102.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.102.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.102.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
```

```
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
```

```
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 3

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.103.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
```

```
set interfaces ge-0/0/1 unit 0 family inet address 10.17.103.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.103.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
set security group-vpn server ike gateway RootSrv local-address 10.16.103.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.103.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.103.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.103.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
```

```
set security group-vpn server ike gateway GM-0004 local-address 10.17.103.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_RootSrv
user@host# set ge-0/0/0 unit 0 family inet address 10.16.103.1/24
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.103.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
```

```
user@host# set ike-policy RootSrv
user@host# set address 10.10.103.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.103.1
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.17.103.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.103.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.103.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.103.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
```

```
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.103.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.103.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
```

```
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.103.1;
                dead-peer-detection always-send;
                local-address 10.16.103.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.103.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.103.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.103.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.103.1;
```

```
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
```

```
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 4

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.104.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
set interfaces ge-0/0/1 unit 0 family inet address 10.17.104.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.104.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
```

```
set security group-vpn server ike gateway RootSrv local-address 10.16.104.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.104.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.104.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.104.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.17.104.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
```

```
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_RootSrv
user@host# set ge-0/0/0 unit 0 family inet address 10.16.104.1/24
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.104.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
```

```
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
user@host# set ike-policy RootSrv
user@host# set address 10.10.104.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.104.1
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.17.104.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.104.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.104.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.104.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.104.1/24;
        }
```

```
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.104.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.104.1;
                dead-peer-detection always-send;
                local-address 10.16.104.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.104.1;
            }
            gateway GM-0002 {
```

```
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.104.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.104.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.104.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
```

```
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
```

```
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.101.1/24
set interfaces ge-0/0/1 unit 0 description To_SubSrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.101.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
```

```
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.101.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_LAN
user@host# set ge-0/0/0 unit 0 family inet address 172.16.101.1/24
user@host# set ge-0/0/1 unit 0 description To_SubSrv
user@host# set ge-0/0/1 unit 0 family inet address 10.18.101.1/24
[edit security zones security-zone LAN]
```

```
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies from-zone WAN to-zone LAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit]
user@host# set security policies default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
```

```
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.101.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

4. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.101.1/24;
        }
```

```
    }
}
ge-0/0/1 {
    unit 0 {
        description To_SubSrv;
        family inet {
            address 10.18.101.1/24;
        }
    }
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.101.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
```

```
            }
        }
    }
}
ipsec-policy {
    from-zone LAN to-zone WAN {
        ipsec-group-vpn GROUP_ID-0001;
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    global {
        policy 1000 {
            match {
```

```
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
```

```
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.102.1/24
set interfaces ge-0/0/1 unit 0 description To_SubSrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.102.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
```

```
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.102.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_LAN
user@host# set ge-0/0/0 unit 0 family inet address 172.16.102.1/24
user@host# set ge-0/0/1 unit 0 description To_SubSrv
user@host# set ge-0/0/1 unit 0 family inet address 10.18.102.1/24
[edit security zones security-zone LAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
```

```
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies from-zone WAN to-zone LAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit]
user@host# set security policies default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
```

```
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.102.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

4. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.102.1/24;
        }
    }
}
```

```
ge-0/0/1 {
    unit 0 {
        description To_SubSrv;
        family inet {
            address 10.18.102.1/24;
        }
    }
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.102.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
            }
        }
```

```
    }
}
ipsec-policy {
    from-zone LAN to-zone WAN {
        ipsec-group-vpn GROUP_ID-0001;
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
```

```
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
```

```
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0003 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.103.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.103.1/24
set interfaces ms-0/2/0 unit 0 family inet
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.103.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
```

```
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.103.1/24
```

```
user@host# set xe-0/0/2 unit 0 family inet address 172.16.103.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.103.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

4. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from source-address 10.17.101.1/32
user@host# set term inbound-ks from source-address 10.17.102.1/32
user@host# set term inbound-ks from source-address 10.17.103.1/32
user@host# set term inbound-ks from source-address 10.17.104.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.17.101.1/32
```

```
user@host# set term outbound-ks from destination-address 10.17.102.1/32
user@host# set term outbound-ks from destination-address 10.17.103.1/32
user@host# set term outbound-ks from destination-address 10.17.104.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

5. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
            }
            address 10.18.103.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
```

```
            address 172.16.103.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.103.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
    }
}
[edit]
```

```
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                source-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
                destination-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
```

```
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0004 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.104.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.104.1/24
set interfaces ms-0/2/0 unit 0 family inet
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.104.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
```

```
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.104.1/24
```

```
user@host# set xe-0/0/2 unit 0 family inet address 172.16.104.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.104.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

4. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from source-address 10.17.101.1/32
user@host# set term inbound-ks from source-address 10.17.102.1/32
user@host# set term inbound-ks from source-address 10.17.103.1/32
user@host# set term inbound-ks from source-address 10.17.104.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.17.101.1/32
```

```
user@host# set term outbound-ks from destination-address 10.17.102.1/32
user@host# set term outbound-ks from destination-address 10.17.103.1/32
user@host# set term outbound-ks from destination-address 10.17.104.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

5. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
            }
            address 10.18.104.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
```

```
            address 172.16.104.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text ""$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.104.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
    }
}
[edit]
```

```
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                source-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
                destination-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
```

```
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Server Cluster Operation | **800**
- Verifying That SAs Are Distributed to Members | **803**
- Verifying IKE SAs on the Servers | **806**
- Verifying IPsec SAs on the Servers and Group Members | **809**
- Verifying IPsec Policies on Group Members | **812**

Confirm that the configuration is working properly.

Verifying Server Cluster Operation

Purpose

Verify that devices in the server cluster recognize peer servers in the group. Ensure that the servers are active and roles in the cluster are properly assigned.

Action

From operational mode, enter the `show security group-vpn server server-cluster`, `show security group-vpn server server-cluster detail`, and `show security group-vpn server statistics` commands on the root-server.

```
user@RootSrv> show security group-vpn server server-cluster
Group: GROUP_ID-0001, Group Id: 1
Role: Root-server, Version Number: 2,
  Peer Gateway                    Peer IP         Role                  Status
  SubSrv01                        10.16.101.1   Sub-server            Active
  SubSrv02                        10.16.102.1   Sub-server            Active
```

```
  SubSrv03                        10.16.103.1   Sub-server              Active
  SubSrv04                        10.16.104.1   Sub-server              Active
```

```
user@RootSrv> show security group-vpn server server-cluster detail
Group: GROUP_ID-0001, Group Id: 1
Role: Root-server, Version Number: 2

Peer gateway: SubSrv01
  Peer IP: 10.16.101.1, Local IP: 10.10.101.1, VR: default
  Role: Sub-server, Status: Active
  CLUSTER-INIT send:                 0
  CLUSTER-INIT recv:                 1
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               2
  CLUSTER-UPDATE recv:               0
  CLUSTER-UPDATE success:            2
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          677
  DPD send fail:                     0
  DPD ACK recv:                      677
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0

Peer gateway: SubSrv02
  Peer IP: 10.16.102.1, Local IP: 10.10.102.1, VR: default
  Role: Sub-server, Status: Active
  CLUSTER-INIT send:                 0
  CLUSTER-INIT recv:                 1
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
```

```
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               2
  CLUSTER-UPDATE recv:               0
  CLUSTER-UPDATE success:            2
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          676
  DPD send fail:                     0
  DPD ACK recv:                      676
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0

user@RootSrv> show security group-vpn server statistics
Group: GROUP_ID-0001, Group Id: 1
  Stats:
      Pull Succeeded                : 0
      Pull Failed                   : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

From operational mode, enter the `show security group-vpn server server-cluster`, `show security group-vpn server server-cluster detail`, and `show security group-vpn server statistics` commands on each sub-server.

```
user@SubSrv01> show security group-vpn server server-cluster
Group: GROUP_ID-0001, Group Id: 1
Role: Sub-server, Version Number: 2,
  Peer Gateway                    Peer IP         Role                   Status
  RootSrv                         10.10.101.1     Root-server            Active
```

```
user@SubSrv01> show security group-vpn server server-cluster detail
Group: GROUP_ID-0001, Group Id: 1
Role: Sub-server, Version Number: 2
```

```
Peer gateway: RootSrv
  Peer IP: 10.10.101.1, Local IP: 10.16.101.1, VR: default
  Role: Root-server, Status: Active
  CLUSTER-INIT send:                 1
  CLUSTER-INIT recv:                 0
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               0
  CLUSTER-UPDATE recv:               2
  CLUSTER-UPDATE success:            2
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          812
  DPD send fail:                     0
  DPD ACK recv:                      812
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0

user@SubSrv01> show security group-vpn server statistics
Group: GROUP_ID-0001, Group Id: 1
  Stats:
      Pull Succeeded               : 4
      Pull Failed                  : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                    : 8
      Push Acknowledged            : 8
      Push Unacknowledged          : 0
```

Verifying That SAs Are Distributed to Members

Purpose

Verify that the sub-servers have received SAs for distribution to group members and the group members have received the SAs.

Action

From operational mode, enter the `show security group-vpn server kek security-associations` and `show security group-vpn server kek security-associations detail` commands on the root-server.

```
user@RootSrv> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
738885  2888      5742c24020056c6a  d6d479543b56404c  1
```

```
user@RootSrv> show security group-vpn server kek security-associations detail
Index 738885, Group Name: GROUP_ID-0001, Group Id: 1
Initiator cookie: 5742c24020056c6a, Responder cookie: d6d479543b56404c
Authentication method: RSA
Lifetime: Expires in 2883 seconds, Activated
Rekey in 2373 seconds
  Algorithms:
   Sig-hash              : sha256
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Server Member Communication: Unicast
  Retransmission Period: 10, Number of Retransmissions: 2
  Group Key Push sequence number: 0

PUSH negotiations in progress: 0
```

From operational mode, enter the `show security group-vpn server kek security-associations` and `show security group-vpn server kek security-associations detail` commands on each sub-server.

```
user@SubSrv01> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
738885  1575      5742c24020056c6a  d6d479543b56404c  1
```

```
user@SubSrv01> show security group-vpn server kek security-associations detail
Index 738879, Group Name: GROUP_ID-0001, Group Id: 1
Initiator cookie: 114e4a214891e42f, Responder cookie: 4b2848d14372e5bd
```

```
Authentication method: RSA
Lifetime: Expires in 4186 seconds, Activated
Rekey in 3614 seconds
  Algorithms:
   Sig-hash              : sha256
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Server Member Communication: Unicast
  Retransmission Period: 10, Number of Retransmissions: 2
  Group Key Push sequence number: 0

PUSH negotiations in progress: 0
```

From operational mode, enter the `show security group-vpn member kek security-associations` and `show security group-vpn member kek security-associations detail` commands on each group member.

For SRX Series Firewall or vSRX Virtual Firewall group members:

```
user@GM-0001> show security group-vpn server kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5455799 10.17.101.1     1466      5742c24020056c6a  d6d479543b56404c  1

user@GM-0001> show security group-vpn server kek security-associations detail
  Index 5455799, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.17.101.1
  Initiator cookie: 5742c24020056c6a, Responder cookie: d6d479543b56404c
  Lifetime: Expires in 1464 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-sha256-128
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
```

```
  Stats:
      Push received            :   0
      Delete received          :   0
```

For MX group members:

```
user@GM-0003> show security group-vpn member kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5184329 10.17.101.1     1323      5742c24020056c6a  d6d479543b56404c  1

user@GM-0003> show security group-vpn member kek security-associations detail
  Index 5184329, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.17.101.1
  Initiator cookie: 5742c24020056c6a, Responder cookie: d6d479543b56404c
  Lifetime: Expires in 1321 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-sha256-128
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Stats:
      Push received            :   0
      Delete received          :   0
```

Verifying IKE SAs on the Servers

Purpose

Display IKE security associations (SAs) on the servers.

Action

From operational mode, enter the `show security group-vpn server ike security-associations` and `show security group-vpn server ike security-associations detail` commands on the root-server.

```
user@RootSrv> show security group-vpn server ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
738880  UP     2221001e980eb08b  5af00708f5da289c  Main           10.16.104.1
738881  UP     59e8c1d328b1d9fd  d63e823fb8be1f22  Main           10.16.101.1
738883  UP     9cb3a49c6771819e  8df3be8c9ddeb2a7  Main           10.16.102.1
738882  UP     9a8a75f05a1384c5  c6d58696c896b730  Main           10.16.103.1
```

```
user@RootSrv> show security group-vpn server ike security-associations detail
IKE peer 10.16.101.1, Index 738881, Gateway Name: SubSrv01
  Role: Responder, State: UP
  Initiator cookie: 59e8c1d328b1d9fd, Responder cookie: d63e823fb8be1f22
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.10.101.1:848, Remote: 10.16.101.1:848
  Lifetime: Expires in 21890 seconds
  Peer ike-id: 10.16.101.1
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :              150112
   Output bytes  :              153472
   Input  packets:                1387
   Output packets:                1387
  Flags: IKE SA is created
IKE peer 10.16.102.1, Index 738883, Gateway Name: SubSrv02
  Role: Responder, State: UP
  Initiator cookie: 9cb3a49c6771819e, Responder cookie: 8df3be8c9ddeb2a7
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.10.102.1:848, Remote: 10.16.102.1:848
  Lifetime: Expires in 21899 seconds
  Peer ike-id: 10.16.102.1
  Xauth user-name: not available
```

```
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :               149788
   Output bytes  :               153148
   Input  packets:                 1384
   Output packets:                 1384
  Flags: IKE SA is created
```

From operational mode, enter the `show security group-vpn server ike security-associations` and `show security group-vpn server ike security-associations detail` commands on each sub-server.

```
user@SubSrv01> show security group-vpn server ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
738878  UP     59e8c1d328b1d9fd  d63e823fb8be1f22  Main           10.10.101.1
```

```
user@SubSrv01> show security group-vpn server ike security-associations detail
IKE peer 10.10.101.1, Index 738878, Gateway Name: RootSrv
  Role: Initiator, State: UP
  Initiator cookie: 59e8c1d328b1d9fd, Responder cookie: d63e823fb8be1f22
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.16.101.1:848, Remote: 10.10.101.1:848
  Lifetime: Expires in 20589 seconds
  Peer ike-id: 10.10.101.1
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :               181444
   Output bytes  :               178084
   Input  packets:                 1646
```

```
   Output packets:                1646
  Flags: IKE SA is created
```

Verifying IPsec SAs on the Servers and Group Members

Purpose

Display IPsec security associations (SAs) on the servers and group members.

Action

From operational mode, enter the `show security group-vpn server ipsec security-associations` and `show security group-vpn server ipsec security-associations detail` commands on the root-server.

```
user@RootSrv> show security group-vpn server ipsec security-associations
Group: GROUP_ID-0001, Group Id: 1
  Total IPsec SAs: 1
  IPsec SA          Algorithm        SPI              Lifetime
  GROUP_ID-0001     ESP:aes-256/sha256 dddef414       2773

user@RootSrv> show security group-vpn server ipsec security-associations detail
Group: GROUP_ID-0001, Group Id: 1
Total IPsec SAs: 1
  IPsec SA: GROUP_ID-0001
    Protocol: ESP, Authentication: sha256, Encryption: aes-256
    Anti-replay: D3P enabled
    SPI: dddef414
    Lifetime: Expires in 1670 seconds, Activated
    Rekey in 1160 seconds
    Policy Name: 1
      Source: 172.16.0.0/12
      Destination: 172.16.0.0/12
      Source Port: 0
      Destination Port: 0
      Protocol: 0
```

From operational mode, enter the `show security group-vpn server ipsec security-associations` and `show security group-vpn server ipsec security-associations detail` commands on each sub-server.

```
user@SubSrv01> show security group-vpn server ipsec security-associations
Group: GROUP_ID-0001, Group Id: 1
  Total IPsec SAs: 1
  IPsec SA          Algorithm        SPI              Lifetime
  GROUP_ID-0001     ESP:aes-256/sha256 dddef414       1520

user@SubSrv01> show security group-vpn server ipsec security-associations detail
Group: GROUP_ID-0001, Group Id: 1
Total IPsec SAs: 1
  IPsec SA: GROUP_ID-0001
    Protocol: ESP, Authentication: sha256, Encryption: aes-256
    Anti-replay: D3P enabled
    SPI: dddef414
    Lifetime: Expires in 1518 seconds, Activated
    Rekey in 1230 seconds
    Policy Name: 1
      Source: 172.16.0.0/12
      Destination: 172.16.0.0/12
      Source Port: 0
      Destination Port: 0
      Protocol: 0
```

From operational mode, enter the `show security group-vpn member ipsec security-associations` and `show security group-vpn member ipsec security-associations detail` commands on each group member

For SRX Series Firewall or vSRX Virtual Firewall group members:

```
user@GM-0001> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>49152 10.17.101.1    848   ESP:aes-256/sha256-128 dddef414 1412/ unlim 1 root

user@GM-0001> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.17.101.1
  Group Id: 1
  Routing Instance: default
  Recovery Probe: Enabled
  DF-bit: clear
```

```
  Stats:
      Pull Succeeded             :   1
      Pull Failed                :   0
      Pull Timeout               :   0
      Pull Aborted               :   0
      Push Succeeded             :   2
      Push Failed                :   0
      Server Failover            :   0
      Delete Received            :   0
      Exceed Maximum Keys(4)     :   0
      Exceed Maximum Policies(10):   0
      Unsupported Algo           :   0
  Flags:
      Rekey Needed:   no

    List of policies received from server:
    Tunnel-id: 49152
      Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
      Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

      Direction: bi-directional, SPI: dddef414
      Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
      Hard lifetime: Expires in 1409 seconds, Activated
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 1193 seconds
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled
```

For MX group members:

```
user@GM-0003> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>10001 10.17.101.1    848   ESP:aes-256/sha256-128 dddef414 1308/ unlim 1 root

user@GM-0003> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.17.101.1
  Group Id: 1
  Rule Match Direction: output,  Tunnel-MTU: 1400
  Routing Instance: default
  DF-bit: clear
```

```
Stats:
    Pull Succeeded            :   1
    Pull Failed               :   0
    Pull Timeout              :   0
    Pull Aborted              :   0
    Push Succeeded            :   2
    Push Failed               :   0
    Server Failover           :   0
    Delete Received           :   0
    Exceed Maximum Keys(4)    :   0
    Exceed Maximum Policies(1):   0
    Unsupported Algo          :   0
Flags:
    Rekey Needed:   no

  List of policies received from server:
  Tunnel-id: 10001
    Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
    Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

    Direction: bi-directional, SPI: dddef414
    Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
    Hard lifetime: Expires in 1305 seconds, Activated
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1087 seconds
    Mode: Tunnel, Type: Group VPN, State: installed
    Anti-replay service: D3P enabled
```

Verifying IPsec Policies on Group Members

Purpose

Display the IPsec policy on an SRX Series Firewall or vSRX Virtual Firewall group member.

This command is not available for MX Series group members.

Action

From operational mode, enter the **show security group-vpn member policy** command on SRX Series Firewall or vSRX Virtual Firewall group members.

```
user@GM-0001> show security group-vpn member policy
Group VPN Name: GROUP_ID-0001, Group Id: 1
From-zone: LAN, To-zone: WAN
  Tunnel-id: 49152, Policy type: Secure
    Source      : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>
    Destination : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>

  Tunnel-id: 63488, Policy type: Fail-close
    Source      : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
    Destination : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
```

SEE ALSO

Group VPNv2 Configuration Overview | 661

Configuring Group VPNs in Group VPNv2 on Routing Device

RELATED DOCUMENTATION

Group VPNv1 | 593

8
CHAPTER

ADVPN

IN THIS CHAPTER

- Auto Discovery VPNs | **815**

Auto Discovery VPNs

SUMMARY

Learn about Auto Discovery VPN and how to configure the feature on the firewall.

IN THIS SECTION

- Understanding Auto Discovery VPN | **815**
- Understanding Traffic Routing with Shortcut Tunnels | **823**
- Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels | **825**
- Example: Configuring ADVPN with OSPFv3 for IPv6 Traffic | **879**
- Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established | **915**
- Platform-Specific Multicast in ADVPN Behavior | **916**

Auto Discovery VPN (ADVPN) dynamically establishes VPN tunnels between spokes to avoid routing traffic through the hub.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific Multicast in ADVPN Behavior" on page 916 section for notes related to your platform.

Understanding Auto Discovery VPN

IN THIS SECTION

- ADVPN Protocol | **816**
- Establishing a Shortcut | **816**
- Shortcut Initiator and Responder Roles | **818**

- Shortcut Attributes | **819**
- Shortcut Termination | **820**
- Multicast Support Using PIM | **821**
- ADVPN Configuration Limitations | **821**

Auto Discovery VPN (ADVPN) is a technology that allows the central Hub to dynamically inform spokes about a better path for traffic between two spokes. When both spokes acknowledge the information from the Hub, they establish a shortcut tunnel and change the routing topology for the host to reach the other side without sending traffic through the Hub.

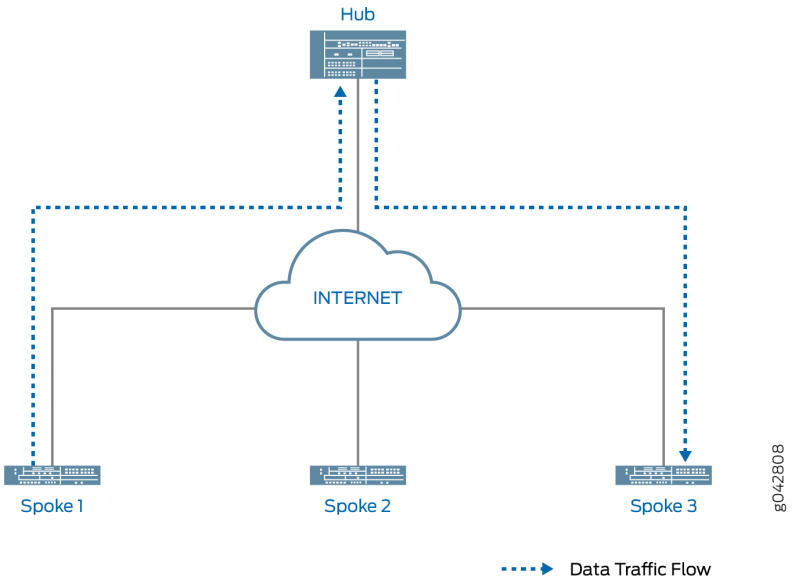
ADVPN Protocol

ADVPN uses an extension of IKEv2 protocol to exchange messages between two peers, that allows the spokes to establish a shortcut tunnel between each other. Devices that support the ADVPN extension send an `ADVPN_SUPPORTED` notification in the *IKEv2 Notify* payload including its capability information and the ADVPN version number during the initial IKE exchange. A device that supports ADVPN can act as either a *shortcut suggester* or a shortcut partner, but not both.

Establishing a Shortcut

An IPsec VPN gateway can act as a *shortcut suggester* when it notices that traffic is exiting a tunnel with one of its peers and entering a tunnel with another peer. Figure 48 on page 817 shows traffic from Spoke 1 to Spoke 3 passing through the Hub.

Figure 48: Spoke-to-Spoke Traffic Passing Through Hub

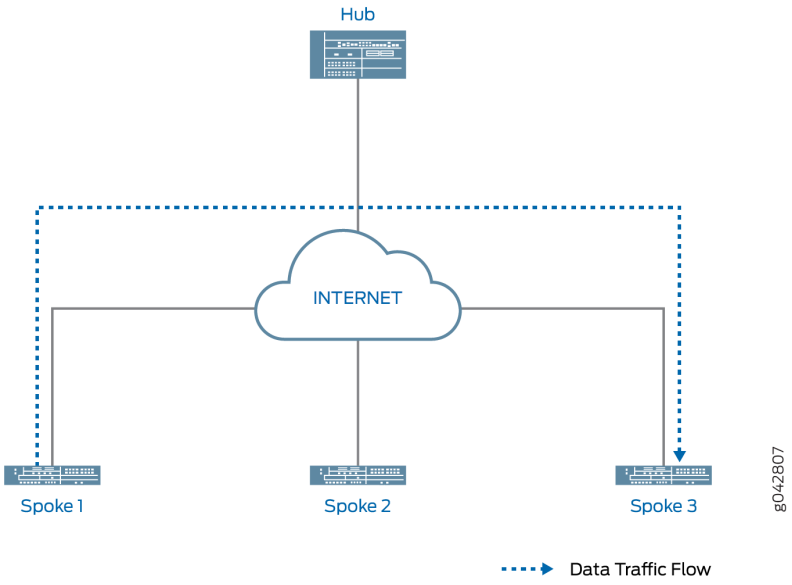


When ADVPN is configured on the devices, ADVPN shortcut capability information is exchanged between the hub and the spokes. As long as Spokes 1 and 3 have previously advertised ADVPN shortcut partner capability to the Hub, the Hub can suggest that Spokes 1 and 3 establish a shortcut between each other.

The shortcut suggester uses its already established IKEv2 SAs with the peers to begin a shortcut exchange with one of the two peers. If the peer accepts the shortcut exchange, then the shortcut suggester begins a shortcut exchange with the other peer. The shortcut exchange includes information to allow the peers (referred to as *shortcut partners*) to establish IKE and IPsec SAs with each other. The creation of the shortcut between the shortcut partners starts only after both peers accept the shortcut exchange.

Figure 49 on page 818 shows traffic passing through a shortcut between Spokes 1 and 3. Traffic from Spoke 1 to Spoke 3 does not need to traverse the Hub.

Figure 49: Spoke-to-Spoke Traffic Passing Through Shortcut



Shortcut Initiator and Responder Roles

The shortcut suggester chooses one of the shortcut partners to act as the initiator for the shortcut; the other partner acts as the responder. If one of the partners is behind a NAT device, then the partner behind the NAT device is chosen as the initiator. If none of the partners is behind a NAT device, the suggester randomly chooses one of the partners as the initiator; the other partner acts as the responder. If both partners are behind NAT devices, then a shortcut cannot be created between them; the suggester does not send a shortcut exchange to any of the peers.

The shortcut suggester begins the shortcut exchange with the responder first. If the responder accepts the shortcut suggestion, then the suggester notifies the initiator.

Using information contained in the shortcut suggester's notification, the shortcut initiator establishes an IKEv2 exchange with the responder, and a new IPsec SA is established between the two partners. On each partner, the route to the network behind its partner now points to the shortcut instead of to the tunnel between the partner and the suggester. Traffic originating behind one of the partners that is destined to a network behind the other shortcut partner flows over the shortcut.

If the partners decline the shortcut suggestion, then the partners notify the suggester with the reason for the rejection. In this case, traffic between the partners continues to flow through the shortcut suggester.

Shortcut Attributes

The shortcut receives some of its attributes from the shortcut suggester while other attributes are inherited from the suggester-partner VPN tunnel configuration. Table 92 on page 819 shows the parameters of the shortcut.

Table 92: Shortcut Parameters

Attributes	Received/Inherited From
ADVPN	Configuration
Antireplay	Configuration
Authentication algorithm	Configuration
Dead peer detection	Configuration
DF bit	Configuration
Encryption algorithm	Configuration
Establish tunnels	Suggester
External interface	Configuration
Gateway policy	Configuration
General IKE ID	Configuration
IKE version	Configuration
Install interval	Configuration
Local address	Configuration

Table 92: Shortcut Parameters *(Continued)*

Attributes	Received/Inherited From
Local identity	Suggester
NAT traversal	Configuration
Perfect forward secrecy	Configuration
Protocol	Configuration
Proxy ID	Not applicable
Remote address	Suggester
Remote identity	Suggester
Respond bad SPI	Configuration
Traffic selector	Not applicable

Shortcut Termination

By default, the shortcut lasts indefinitely. Shortcut partners terminate the shortcut if traffic falls below a specified rate for a specified time. By default, the shortcut gets terminated if traffic falls below 5 packets per second for 300 seconds; the idle time and idle threshold values are configurable for partners. You can manually delete the shortcut on either shortcut partner with the `clear security ike security-association` or `clear security ipsec security-association` commands to clear the corresponding IKE or IPsec SA. Either of the shortcut partners can terminate the shortcut at any time by sending an IKEv2 delete payload to the other shortcut partner.

When the shortcut is terminated, the corresponding IKE SA and all child IPsec SAs are deleted. After the shortcut is terminated, the corresponding route is deleted on both shortcut partners and traffic between the two peers again flows through the suggester. Shortcut termination information is sent from a partner to the suggester.

The lifetime of a shortcut is independent of the tunnel between the shortcut suggester and shortcut partner. The shortcut is not terminated simply because the tunnel between the suggester and partner is terminated.

Multicast Support Using PIM

The firewalls support Protocol Independent Multicast (PIM) in point-to-multipoint (P2MP) mode in ADVPN infrastructure. You can enable PIM on the firewall's secure tunnel interface, st0, with P2MP mode. The support for multicast traffic using PIM in ADVPN is similar to the support provided in AutoVPN. ADVPN follows same considerations as AutoVPN when configuring multicast support. For more details on understanding multicast support using PIM on P2MP infrastructure, see "Understand AutoVPN" on page 920. To enable PIM on st0 P2MP interface:

- In Multinode High Availability environment, P2MP multicast is achieved using node-local tunnels. The routing protocol over the st0 interface doesn't support synced-state tunnel. See IPsec VPN Support in Multinode High Availability.

One of the firewalls is a shortcut suggester and rest of the firewalls are shortcut partners. Typically, the multicast sender resides behind the shortcut suggester, while the multicast receivers are behind the shortcut partners. For multicast support, the secure tunnel interface, st0, on the suggester and the partner devices are configured with PIM P2MP mode. On each of these devices, the st0 P2MP interface tracks all PIM joins per neighbor to ensure that the multicast forwarding or replication happens only to those neighbors that are in joined state.

The firewalls support IP multicast traffic in PIM sparse mode over the st0 P2MP interface. The suggester acts as the first-hop router (FHR) or the rendezvous point (RP). The partners can act as the last-hop routers (LHR) in the P2MP network. The devices in the network replicate the multicast data packets to neighbors that join the multicast group.

For details on how to configure PIM on P2MP infrastructure, see "Configure Multicast Support on P2MP Infrastructure" on page 1198.

ADVPN Configuration Limitations

Note the following limitations when configuring ADVPN:

- ADVPN is only supported for site-to-site communications. Configuring an ADVPN suggester is only allowed on AutoVPN hubs.
- You cannot configure both suggester and partner roles. When ADVPN is enabled on a gateway, you cannot disable both suggester and partner roles on the gateway.
- You cannot create a shortcut between partners that are both behind NAT devices. The suggester can initiate a shortcut exchange only if one of the partners is behind a NAT device or if no partners are behind NAT devices.

- To use an IPv6 address for ADVPN:
  - You must configure the st0 interface with P2MP support on all the hub and spoke devices.
  - You must run dynamic routing protocols (DRPs) such as the OSPFv3 to update the routing preference to shortcut tunnel over static tunnel.
  - Note that you cannot configure the VPN monitor feature with IPv6 P2MP st0 interface based ADVPN.
- You can run the ADVPN service with a DRP that supports either the IPv6 address or IPv4 address but not both at the same time.
- For configuration changes on the partner, such as enable, disable or role change, the iked:
  1. Tears down and renegotiates the static IKE SA and the IPsec SA to exchange the new capability.
  2. Cleans the shortcut IKE SA and the IPsec SA, and the suggestion information that exists.
- For non-ADVPN configuration changes, such as:
  1. The static tunnel configuration change that leads to clearing of both the static IKE SA and the IPsec SA, the iked tears down the shortcut IKE SA and the IPsec SA. The iked cleans the suggestion information. The shortcut tunnel doesn't renegotiate again, until it receives shortcut suggestion from the suggester.
  2. The static tunnel configuration change that leads to clearing of the static tunnel IPsec SA only, the iked tears down the shortcut IKE SA and the IPsec SA. The iked cleans the suggestion information. The shortcut tunnel doesn't renegotiate again, until it receives shortcut suggestion from the suggester.

We do not support the following configurations with ADVPN with both the kmd and the iked processes:

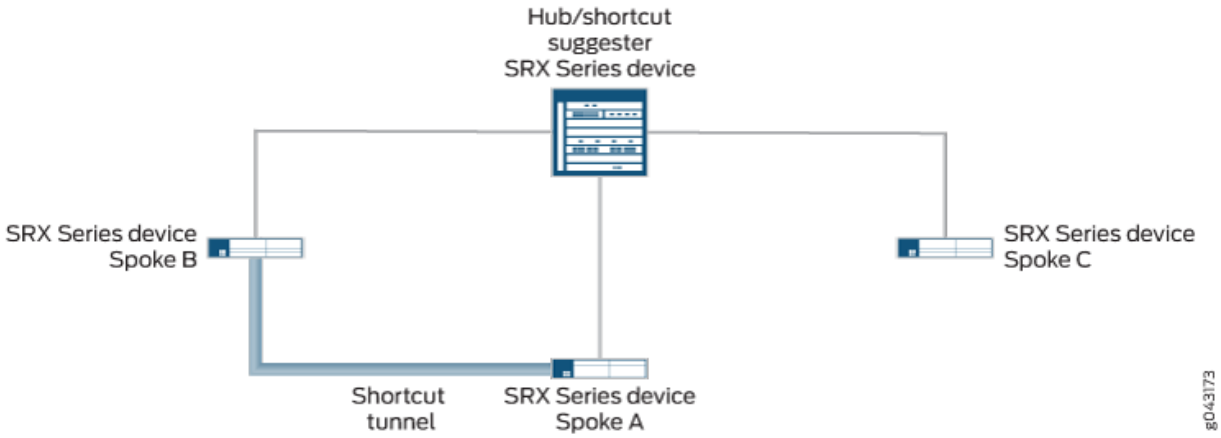
- IKEv1
- Policy-based VPN
- IKEv2 configuration payload
- Traffic selectors
- Point-to-point secure tunnel interfaces
- Seeded preshared key
- Shared preshared key—No support with kmd process

Understanding Traffic Routing with Shortcut Tunnels

Tunnel flaps or catastrophic changes can cause both static tunnels and shortcut tunnels to go down. When this happens, traffic to a specific destination might be routed through an unexpected shortcut tunnel instead of through an expected static tunnel.

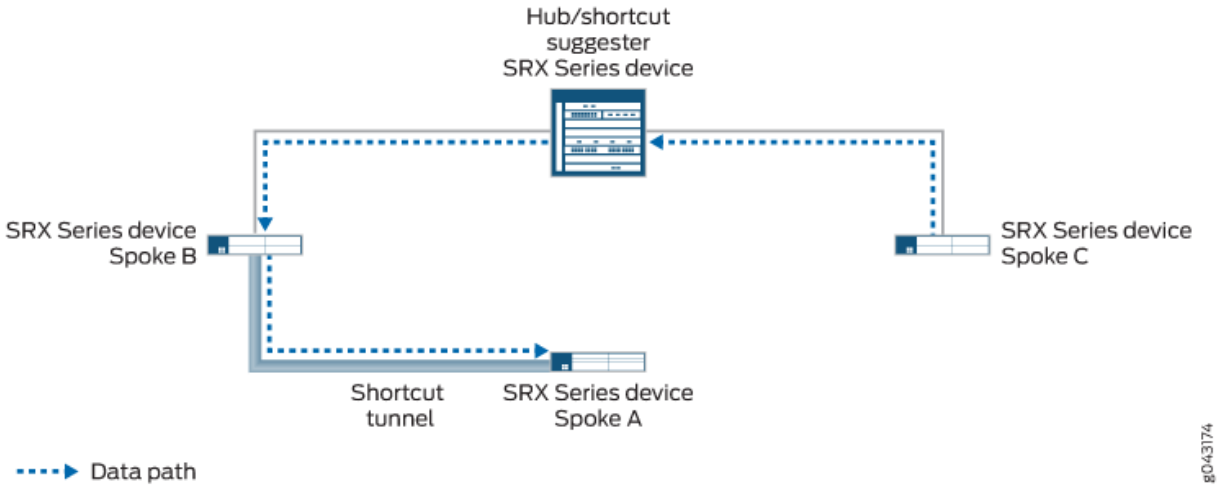
In Figure 50 on page 823, static tunnels exist between the hub and each of the spokes. OSPF adjacencies are established between the hub and spokes. Spoke A also has a shortcut tunnel with Spoke B and OSPF adjacencies are established between the spokes. The hub (the shortcut suggester) recognizes that if connectivity between the hub and Spoke A goes down, Spoke A's network can be reached through the shortcut tunnel between Spoke B and Spoke A.

Figure 50: Static Tunnels and Shortcut Tunnel Established in Hub-and-Spoke Network



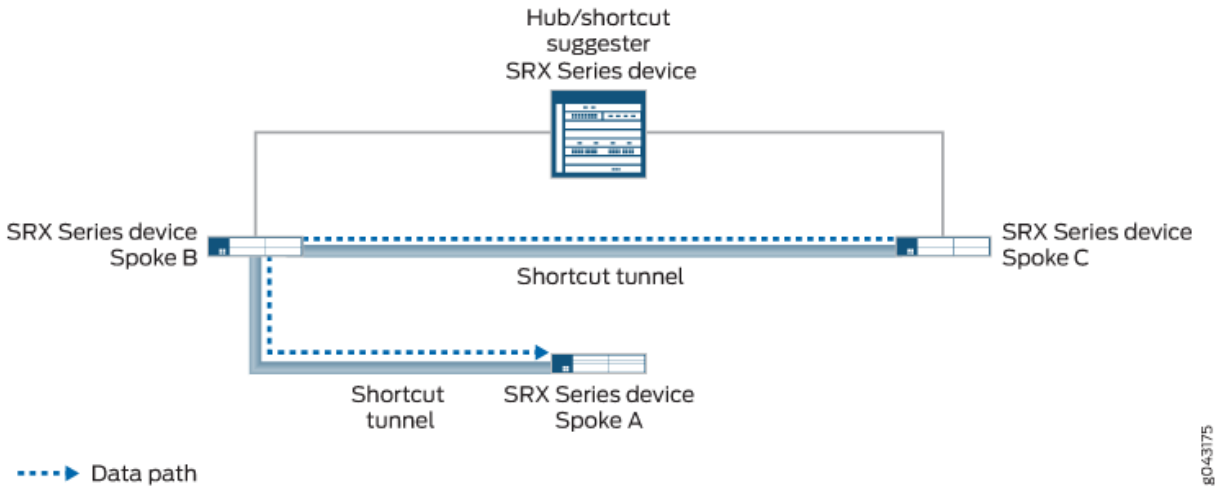
In Figure 51 on page 824, the static tunnel between the hub and Spoke A is down. If there is new traffic from Spoke C to Spoke A, Spoke C forwards the traffic to the hub because it does not have a shortcut tunnel with Spoke A. The hub does not have an active static tunnel with Spoke A but it recognizes that there is a shortcut tunnel between Spoke A and Spoke B, so it forwards the traffic from Spoke C to Spoke B.

Figure 51: Traffic Path from Spoke C to Spoke A



As long as both Spoke B and Spoke C support Auto Discovery VPN (ADVPN) partner capability, the hub can suggest that the spokes establish a direct shortcut between each other. This occurs even though there is no direct traffic between the two spokes. Traffic from Spoke C to Spoke A travels through the shortcut tunnel between Spoke C and Spoke B, and then through the shortcut tunnel between Spoke B and Spoke A (see Figure 52 on page 824).

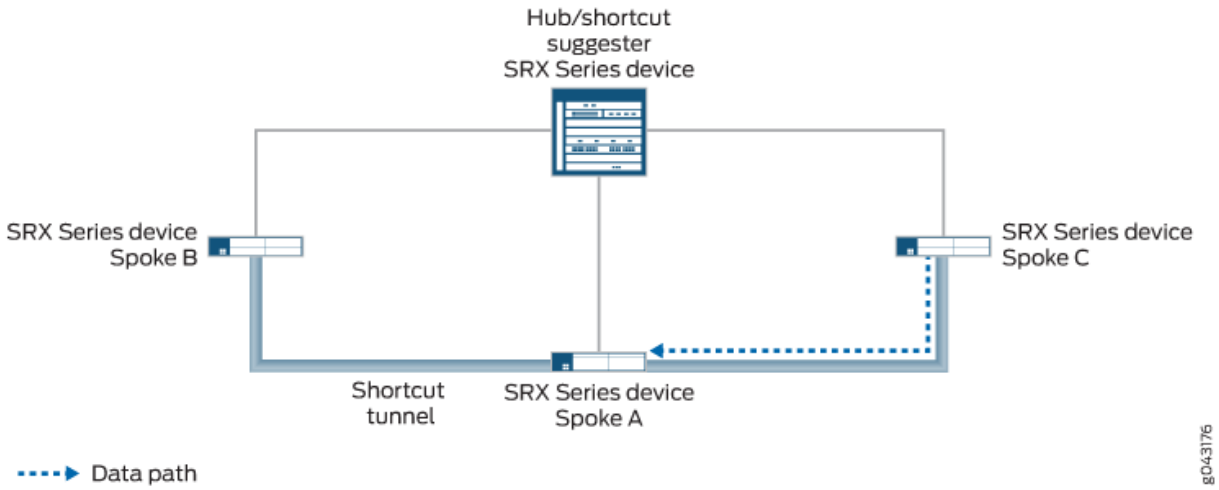
Figure 52: Traffic Path from Spoke C to Spoke A Through Shortcut Tunnels



When the static tunnel between the hub and Spoke A is reestablished, the tunnel is advertised to all spokes. Spoke C learns that there is a better route to reach Spoke A; instead of passing traffic through

Spoke B, it forwards traffic for Spoke A to the hub. The hub suggests that a shortcut tunnel be established between Spoke C and Spoke A. When the shortcut tunnel is established between Spoke C and Spoke A, traffic flows through the shortcut tunnel (see Figure 53 on page 825). Traffic between Spoke C and Spoke A no longer travels through Spoke B, and the shortcut tunnel between Spoke B and Spoke C eventually disappears.

Figure 53: Traffic Path from Spoke C to Spoke A Through Shortcut Tunnel



You can use the `connection-limit` option at the [`edit security ike gateway` *`gateway-name`* `advpn partner`] hierarchy level to set the maximum number of shortcut tunnels that can be created with different shortcut partners using a particular gateway. The maximum number, which is also the default, is platform-dependent.

Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels

IN THIS SECTION

- Requirements | **826**
- Overview | **826**
- Configuration | **831**

- Verification | 856

If you are deploying an AutoVPN network, you might be able to increase your network resource utilization by configuring Auto Discovery VPN (ADVPN). In AutoVPN networks, VPN traffic flows through the hub even when the traffic is travelling from one spoke to another. ADVPN allows VPN tunnels to be established dynamically between spokes, which can result in better network resource utilization. Use this example to configure ADVPN to enable dynamic spoke-to-spoke VPN tunnels in your AutoVPN network.

Requirements

This example uses the following hardware and software components:

- Three supported firewalls as AutoVPN hub and spokes.
- Junos OS Release that support ADVPN.
- Digital certificates enrolled in the hub and spokes that allow the devices to authenticate each other.

Before you begin:

1. Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.
2. Enroll the digital certificates in each device.
3. See Enroll Certificate.

This example uses the OSPF dynamic routing protocol as well as static route configurations to forward packets through VPN tunnels. You should be familiar with the OSPF dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

- Topology | 829

This example shows the configurations of an AutoVPN hub and two spokes for ADVPN. The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well

as to access resources on the hub. While traffic is initially passed from one spoke to the other through the hub, ADVPN allows the spokes to establish a direct security association between each other. The hub acts as the shortcut suggester. On the hub, the ADVPN configuration disables the `partner` role. On the spokes, ADVPN configuration disables the `suggester` role.

Certain Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and spokes must have the same values. Table 93 on page 827 shows the values used in this example.

Table 93: Phase 1 and Phase 2 Options for AutoVPN Hub and Spokes for ADVPN Example

Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5
Authentication algorithm	sha1
Encryption algorithm	aes-256-cbc
IKE policy:	
Certificate	local-certificate
IKE gateway:	
Version	v2-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96

Table 93: Phase 1 and Phase 2 Options for AutoVPN Hub and Spokes for ADVPN Example *(Continued)*

Option	Value
Encryption algorithm	aes-256-cbc
IPsec policy:	
Perfect Forward Secrecy (PFS) group	group5

The IKE gateway configuration on the hub and spokes include remote and local values that identify VPN peers. Table 94 on page 828 shows the IKE gateway configuration for the hub and spokes in this example.

Table 94: IKE Gateway Configuration for ADVPN Example

Option	Hub	Spokes
Remote IP address	Dynamic	Spoke 1: 11.1.1.1 Spoke 2: 11.1.1.1
Local IP address	11.1.1.1	Spoke 1: 21.1.1.2 Spoke 2: 31.1.1.2
Remote IKE ID	Distinguished name (DN) with the string “XYZ” in the organization (O) field and “Sales” in the organization unit (OU) field in the spokes’ certificates	DN with the string “Sales” in the OU field in the hub’s certificate
Local IKE ID	DN on the hub’s certificate	DN on the spokes’ certificate

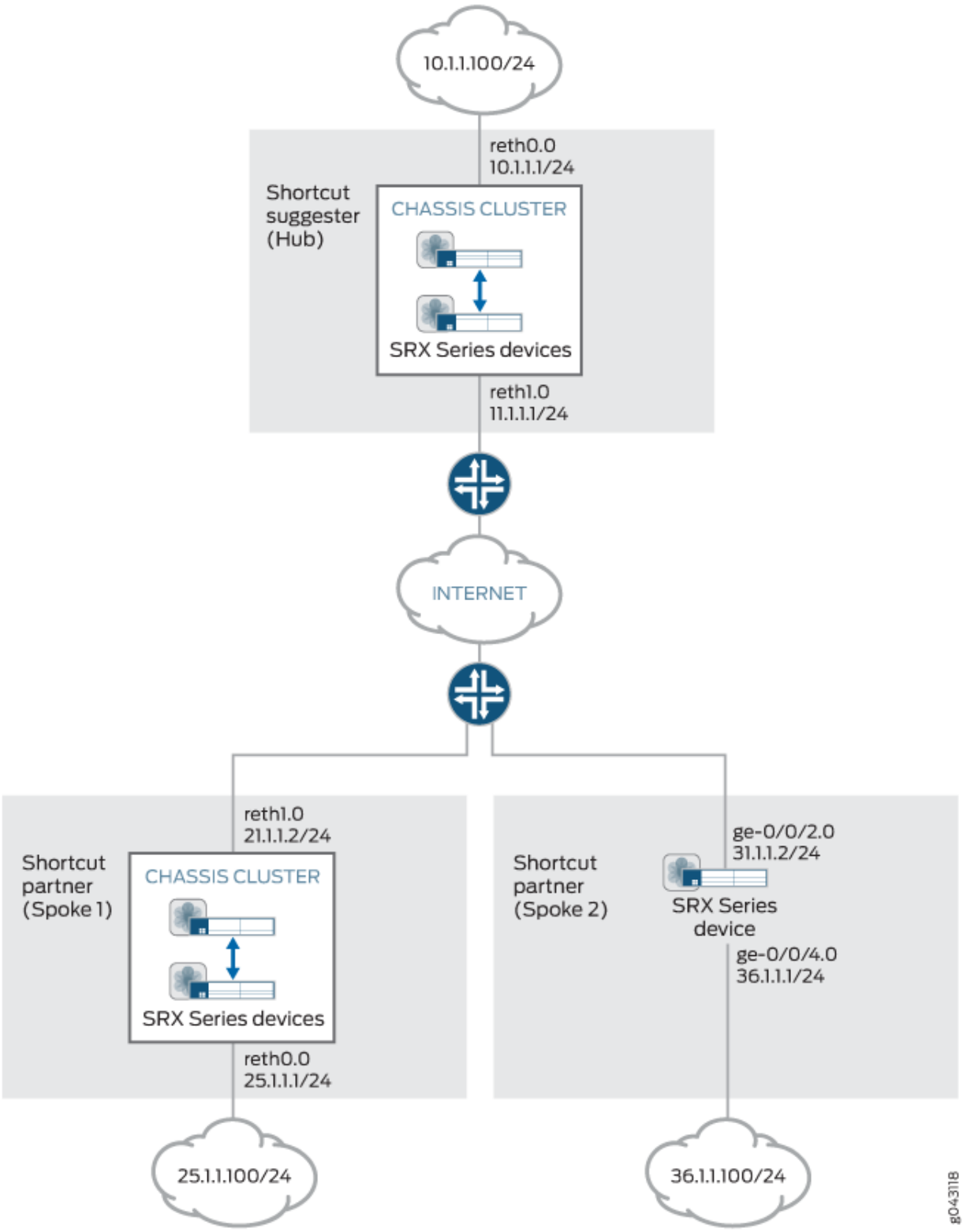
The hub authenticates the spokes’ IKE ID if the subject fields of the spokes’ certificates contain the string “XYZ” in the O field and “Sales” in the OU field.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments.

Topology

Figure 54 on page 830 shows the firewalls to be configured for this example.

Figure 54: AutoVPN Deployment with ADVPN



Configuration

IN THIS SECTION

- Configuring the Suggester (Hub) | 831
- Configuring the Partner (Spoke 1) | 840
- Configuring the Partner (Spoke 2) | 848

Configuring the Suggester (Hub)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-0/0/4 gigether-options redundant-parent reth1
set interfaces ge-7/0/3 gigether-options redundant-parent reth0
set interfaces ge-7/0/4 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 11.1.1.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 172.16.1.1/24
set protocols ospf graceful-restart restart-duration 300
set protocols ospf graceful-restart notify-duration 300
set protocols ospf graceful-restart no-strict-lsa-checking
set protocols ospf area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.1 metric 10
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
set protocols ospf area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface reth0.0
set routing-options graceful-restart
set routing-options static route 21.1.1.0/24 next-hop 11.1.1.2
```

```
set routing-options static route 31.1.1.0/24 next-hop 11.1.1.2
set routing-options router-id 172.16.1.1
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate Suggester_Certificate_ID
set security ike gateway SUGGESTER_GW ike-policy IKE_POL
set security ike gateway SUGGESTER_GW dynamic distinguished-name wildcard O=XYZ, OU=Sales
set security ike gateway SUGGESTER_GW dynamic ike-user-type group-ike-id
set security ike gateway SUGGESTER_GW dead-peer-detection
set security ike gateway SUGGESTER_GW local-identity distinguished-name
set security ike gateway SUGGESTER_GW external-interface reth1.0
set security ike gateway SUGGESTER_GW local-address 11.1.1.1
set security ike gateway SUGGESTER_GW advpn partner disable
set security ike gateway SUGGESTER_GW advpn suggester
set security ike gateway SUGGESTER_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn SUGGESTER_VPN bind-interface st0.1
set security ipsec vpn SUGGESTER_VPN ike gateway SUGGESTER_GW
set security ipsec vpn SUGGESTER_VPN ike ipsec-policy IPSEC_POL
set security pki ca-profile advpn ca-identity advpn
set security pki ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure the suggester:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/3 gigether-options redundant-parent reth0
user@host# set ge-0/0/4 gigether-options redundant-parent reth1
user@host# set ge-7/0/3 gigether-options redundant-parent reth0
user@host# set ge-7/0/4 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 10.1.1.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 11.1.1.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 172.16.1.1/24
```

2. Configure the routing protocol and static routes.

```
[edit protocols ospf]
user@host# set graceful-restart restart-duration 300
user@host# set graceful-restart notify-duration 300
user@host# set graceful-restart no-strict-lsa-checking
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 metric 10
user@host# set area 0.0.0.0 interface st0.1 retransmit-interval 1
user@host# set area 0.0.0.0 interface st0.1 dead-interval 40
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface reth0.0
[edit routing-options]
user@host# set graceful-restart
user@host# set static route 21.1.1.0/24 next-hop 11.1.1.2
user@host# set static route 31.1.1.0/24 next-hop 11.1.1.2
user@host# set router-id 172.16.1.1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
```

```
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate Suggester_Certificate_ID
[edit security ike gateway SUGGESTER_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard O=XYZ, OU=Sales
user@host# set dynamic ike-user-type group-ike-id
user@host# set dead-peer-detection
user@host# set local-identity distinguished-name
user@host# set external-interface reth1.0
user@host# set local-address 11.1.1.1
user@host# set advpn partner disable
user@host# set advpn suggester
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security isec vpn SUGGESTER_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway SUGGESTER_GW
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile advpn ca-identity advpn
user@host# set ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
```

6. Configure zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth1.0
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
    user@host# show interfaces
    ge-0/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-0/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-7/0/3 {
        gigether-options {
            redundant-parent reth0;
```

```
        }
    }
    ge-7/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.1.1.1/24;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 11.1.1.1/24;
            }
        }
    }
    st0 {
        unit 1 {
            multipoint;
            family inet {
                address 172.16.1.1/24;
            }
        }
    }
    [edit]
    user@host# show protocols
    ospf {
        graceful-restart {
            restart-duration 300;
            notify-duration 300;
            no-strict-lsa-checking;
        }
```

```
        area 0.0.0.0 {
            interface st0.1 {
                interface-type p2mp;
                metric 10;
                retransmit-interval 1;
                dead-interval 40;
                demand-circuit;
                dynamic-neighbors;
            }
            interface reth0.0;
        }
    }
    [edit]
    user@host# show routing-options
    graceful-restart;
    static {
        route 21.1.1.0/24 next-hop 11.1.1.2;
        route 31.1.1.0/24 next-hop 11.1.1.2;
    }
    router-id 172.16.1.1;
    [edit]
    user@host# show security ike
    proposal IKE_PROP {
        authentication-method rsa-signatures;
        dh-group group5;
        authentication-algorithm sha1;
        encryption-algorithm aes-256-cbc;
    }
    policy IKE_POL {
        proposals IKE_PROP;
        certificate {
            local-certificate Suggester_Certificate_ID;
        }
    }
    gateway SUGGESTER_GW {
        ike-policy IKE_POL;
        dynamic {
            distinguished-name {
                wildcard O=XYZ, OU=Sales;
            }
            ike-user-type group-ike-id;
        }
        dead-peer-detection {
```

```
        }
        local-identity distinguished-name;
        external-interface reth1.0
        local-address 11.1.1.1;
        advpn {
            partner {
                disable;
                }
                suggester {
            ]
        }
        version v2-only;
    }
    [edit]
    user@host# show security ipsec
    proposal IPSEC_PROP {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm aes-256-cbc;
    }
    policy IPSEC_POL {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals IPSEC_PROP;
    }
    vpn SUGGESTER_VPN {
        bind-interface st0.1;
        ike {
            gateway SUGGESTER_GW;
            ipsec-policy IPSEC_POL;
        }
    }
    [edit]
    user@host# show security pki
    ca-profile advpn {
        ca-identity advpn;
        enrollment {
            url http://10.157.92.176:8080/scep/advpn/;
        }
    }
    [edit]
    user@host# show security zones
```

```
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            st0.1;
            reth0.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            reth1.0;
        }
    }
    [edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Partner (Spoke 1)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-0/0/4 gigether-options redundant-parent reth1
set interfaces ge-7/0/3 gigether-options redundant-parent reth0
set interfaces ge-7/0/4 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 25.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 21.1.1.2/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 172.16.1.2/24
set protocols ospf graceful-restart restart-duration 300
set protocols ospf graceful-restart notify-duration 300
set protocols ospf graceful-restart no-strict-lsa-checking
set protocols ospf area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.1 metric 15
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
set protocols ospf area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface reth0.0
set routing-options graceful-restart
set routing-options static route 11.1.1.0/24 next-hop 21.1.1.1
set routing-options static route 31.1.1.0/24 next-hop 21.1.1.1
set routing-options router-id 172.16.1.2
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate Partner1_Certificate_ID
set security ike gateway PARTNER_GW ike-policy IKE_POL
set security ike gateway PARTNER_GW address 11.1.1.1
set security ike gateway PARTNER_GW local-identity distinguished-name
set security ike gateway PARTNER_GW remote-identity distinguished-name container OU=Sales
```

```
set security ike gateway PARTNER_GW external-interface reth1
set security ike gateway PARTNER_GW local-address 21.1.1.2
set security ike gateway PARTNER_GW advpn suggester disable
set security ike gateway PARTNER_GW advpn partner
set security ike gateway PARTNER_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn PARTNER_VPN bind-interface st0.1
set security ipsec vpn PARTNER_VPN ike gateway PARTNER_GW
set security ipsec vpn PARTNER_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn PARTNER_VPN establish-tunnels immediately
set security pki ca-profile advpn ca-identity advpn
set security pki ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/3 gigether-options redundant-parent reth0
user@host# set ge-0/0/4 gigether-options redundant-parent reth1
user@host# set ge-7/0/3 gigether-options redundant-parent reth0
user@host# set ge-7/0/4 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 25.1.1.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
```

```
user@host# set reth1 unit 0 family inet address 21.1.1.2/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 172.16.1.2/24
```

2. Configure the routing protocol and static routes.

```
[edit protocols ospf]
user@host# set graceful-restart restart-duration 300
user@host# set graceful-restart notify-duration 300
user@host# set graceful-restart no-strict-lsa-checking
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 metric 15
user@host# set area 0.0.0.0 interface st0.1 retransmit-interval 1
user@host# set area 0.0.0.0 interface st0.1 dead-interval 40
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set protocols ospf area 0.0.0.0 interface reth0.0
[edit routing-options]
user@host# set graceful-restart
user@host# set static route 11.1.1.0/24 next-hop 21.1.1.1
user@host# set static route 31.1.1.0/24 next-hop 21.1.1.1
user@host# set router-id 172.16.1.2
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate Partner1_Certificate_ID
[edit security ike gateway PARTNER_GW]
user@host# set ike-policy IKE_POL
user@host# set address 11.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=Sales
user@host# set external-interface reth1
user@host# set local-address 21.1.1.2
user@host# set advpn suggester disable
```

```
user@host# set advpn partner
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security isec vpn PARTNER_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway PARTNER_GW
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile advpn ca-identity advpn
user@host# set ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
```

6. Configure zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth1.0
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
    user@host# show interfaces
    ge-0/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-0/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-7/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-7/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
```

```
            address 25.1.1.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 21.1.1.2/24;
        }
    }
}
st0 {
    unit 1 {
        multipoint;
        family inet {
            address 172.16.1.2/24;
        }
    }
}
[edit]
user@host# show protocols
ospf {
    graceful-restart {
        restart-duration 300;
        notify-duration 300;
        no-strict-lsa-checking;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            metric 15;
            retransmit-interval 1;
            dead-interval 40;
            demand-circuit;
            dynamic-neighbors;
        }
        interface reth0.0;
    }
}
[edit]
```

```
user@host# show routing-options
graceful-restart;
static {
    route 11.1.1.0/24 next-hop 21.1.1.1;
    route 31.1.1.0/24 next-hop 21.1.1.1;
}
router-id 172.16.1.2;
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate Partner1_Certificate_ID;
    }
}
gateway PARTNER_GW {
    ike-policy IKE_POL;
    address 11.1.1.1;
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=Sales;
    external-interface reth1;
    local-address 21.1.1.2;
    advpn {
        suggester {
            disable;
        }
        partner {
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
```

```
    }
    policy IPSEC_POL {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals IPSEC_PROP;
    }
    vpn PARTNER_VPN {
        bind-interface st0.1;
        ike {
            gateway PARTNER_GW;
            ipsec-policy IPSEC_POL;
        }
        establish-tunnels immediately;
    }
    [edit]
    user@host# show security pki
    ca-profile advpn {
        ca-identity advpn;
        enrollment {
            url http://10.157.92.176:8080/scep/advpn/;
        }
    }
    [edit]
    user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            st0.1;
            reth0.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
```

```
            }
            protocols {
                all;
            }
        }
        interfaces {
            reth1.0;
        }
    }
    [edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Partner (Spoke 2)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/2 unit 0 family inet address 31.1.1.2/24
set interfaces ge-0/0/4 unit 0 family inet address 36.1.1.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 172.16.1.3/24
set protocols ospf graceful-restart restart-duration 300
set protocols ospf graceful-restart notify-duration 300
set protocols ospf graceful-restart no-strict-lsa-checking
set protocols ospf area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.1 metric 15
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
set protocols ospf area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
set routing-options graceful-restart
set routing-options static route 11.1.1.0/24 next-hop 31.1.1.1
set routing-options static route 21.1.1.0/24 next-hop 31.1.1.1
```

```
set routing-options router-id 172.16.1.3
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate Partner2_Certificate_ID
set security ike gateway PARTNER_GW ike-policy IKE_POL
set security ike gateway PARTNER_GW address 11.1.1.1
set security ike gateway PARTNER_GW dead-peer-detection
set security ike gateway PARTNER_GW local-identity distinguished-name
set security ike gateway PARTNER_GW remote-identity distinguished-name container OU=Sales
set security ike gateway PARTNER_GW external-interface ge-0/0/2.0
set security ike gateway PARTNER_GW local-address 31.1.1.2
set security ike gateway PARTNER_GW advpn suggester disable
set security ike gateway PARTNER_GW advpn partner
set security ike gateway PARTNER_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn PARTNER_VPN bind-interface st0.1
set security ipsec vpn PARTNER_VPN ike gateway PARTNER_GW
set security ipsec vpn PARTNER_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn PARTNER_VPN establish-tunnels immediately
set security pki ca-profile advpn ca-identity advpn
set security pki ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/4.0
set security zones security-zone trust interfaces st0.1
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/2.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/2 unit 0 family inet address 31.1.1.2/24
user@host# set ge-0/0/4 unit 0 family inet address 36.1.1.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 172.16.1.3/24
```

2. Configure the routing protocol and static routes.

```
[edit protocols ospf
user@host# set graceful-restart restart-duration 300
user@host# set graceful-restart notify-duration 300
user@host# set graceful-restart no-strict-lsa-checking
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 metric 15
user@host# set area 0.0.0.0 interface st0.1 retransmit-interval 1
user@host# set area 0.0.0.0 interface st0.1 dead-interval 40
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/4.0
[edit routing-options]
user@host# set graceful-restart
user@host# set static route 11.1.1.0/24 next-hop 31.1.1.1
user@host# set static route 21.1.1.0/24 next-hop 31.1.1.1
user@host# set router-id 172.16.1.3
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate Partner2_Certificate_ID
[edit security ike gateway PARTNER_GW]
```

```
user@host# set ike-policy IKE_POL
user@host# set address 11.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=Sales
user@host# set external-interface ge-0/0/2.0
user@host# set local-address 31.1.1.2
user@host# set advpn suggester disable
user@host# set advpn partner
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security isec vpn PARTNER_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway PARTNER_GW
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile advpn ca-identity advpn
user@host# set ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
```

6. Configure zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/4.0
user@host# set interfaces st0.1
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
```

```
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/2.0
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
    user@host# show interfaces
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 31.1.1.2/24;
            }
        }
    }
    ge-0/0/4{
        unit 0 {
            family inet {
                address 36.1.1.1/24;
            }
        }
    }
    st0 {
        unit 1 {
            multipoint;
            family inet {
                address 172.16.1.3/24;
            }
        }
    }
    [edit]
```

```
user@host# show protocols
ospf {
    graceful-restart {
        restart-duration 300;
        notify-duration 300;
        no-strict-lsa-checking;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            metric 15;
            retransmit-interval 1;
            dead-interval 40;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/4.0;
    }
}
[edit]
user@host# show routing-options
graceful-restart;
static {
    route 11.1.1.0/24 next-hop 31.1.1.1;
    route 21.1.1.0/24 next-hop 31.1.1.1;
}
router-id 172.16.1.3;
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate Partner2_Certificate_ID
    }
}
gateway PARTNER_GW {
    ike-policy IKE_POL;
```

```
        address 11.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name container OU=Sales;
        external-interface ge-0/0/2.0;
        local-address 31.1.1.2;
        advpn {
            suggester{
                disable;
            }
            partner {
            }
        }
        version v2-only;
    }
    [edit]
    user@host# show security ipsec
    proposal IPSEC_PROP {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm aes-256-cbc;
    }
    policy IPSEC_POL {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals IPSEC_PROP;
    }
    vpn PARTNER_VPN {
        bind-interface st0.1;
        ike {
            gateway PARTNER_GW;
            ipsec-policy IPSEC_POL;
        }
        establish-tunnels immediately;
    }
    [edit]
    user@host# show security pki
    ca-profile advpn {
        ca-identity advpn;
        enrollment {
            url http://10.157.92.176:8080/scep/advpn/;
        }
    }
```

```
    [edit]
    user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/4.0;
            st0.1;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/2.0;
        }
    }
    [edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Tunnels Between the Hub and Spokes | 856
- Verifying the Shortcut Tunnel Between Partners | 866

Confirm that the configuration is working properly. First, verify that tunnels are established between the AutoVPN hub and spokes. When traffic is passed from one spoke to another through the hub, a shortcut can be established between the spokes. Verify that the shortcut partners have established a tunnel between them and that a route to the peer is installed on the partners.

Verifying Tunnels Between the Hub and Spokes

Purpose

Verify that tunnels are established between the AutoVPN hub and spokes. Initial traffic from one spoke to another must travel through the hub.

Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the hub and spokes.

The following commands are entered on the hub:

```
user@host> show security ike security-associations
node1:
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
10957048 UP    2d58d8fbc396762d  46145be580c68be0  IKEv2          31.1.1.2
10957049 UP    fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2          21.1.1.2
```

```
user@host> show security ike security-associations detail
node1:
--------------------------------------------------------------------------
IKE peer 31.1.1.2, Index 10957048, Gateway Name: SUGGESTER_GW
```

```
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Responder, State: UP
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 11.1.1.1:500, Remote: 31.1.1.2:500
  Lifetime: Expires in 28196 seconds
  Peer ike-id: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                2030
   Output bytes  :                2023
   Input  packets:                   4
   Output packets:                   4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 31.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created

IKE peer 21.1.1.2, Index 10957049, Gateway Name: SUGGESTER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Responder, State: UP
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
```

```
  Local: 11.1.1.1:500, Remote: 21.1.1.2:500
  Lifetime: Expires in 28219 seconds
Peer ike-id: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2030
   Output bytes  :                 2023
   Input  packets:                    4
   Output packets:                    4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 21.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node1:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <201326593 ESP:aes-cbc-256/sha1 44ccf265 2999/ unlim - root 500 31.1.1.2
  >201326593 ESP:aes-cbc-256/sha1 a9d301b0 2999/ unlim - root 500 31.1.1.2
  <201326594 ESP:aes-cbc-256/sha1 98a2b155 3022/ unlim - root 500 21.1.1.2
  >201326594 ESP:aes-cbc-256/sha1 de912bcd 3022/ unlim - root 500 21.1.1.2
```

```
user@host> show security ipsec security-associations detail
node1:
--------------------------------------------------------------------------

ID: 201326593 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 31.1.1.2
```

```
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 12:57:48 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:57:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:57:48 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2414 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2414 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 201326594 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 21.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 3, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 12:58:11 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:58:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:58:11 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 3014 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2436 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
```

```
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 3014 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2436 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 28 destinations, 28 routes (27 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

25.1.1.0/24        *[OSPF/10] 00:00:27, metric 11
                    > to 172.16.1.2 via st0.1
36.1.1.0/24        *[OSPF/10] 00:00:27, metric 11
                    > to 172.16.1.3 via st0.1
172.16.1.2/32      *[OSPF/10] 00:00:27, metric 10
                    > to 172.16.1.2 via st0.1
172.16.1.3/32      *[OSPF/10] 00:00:27, metric 10
                    > to 172.16.1.3 via st0.1
224.0.0.5/32       *[OSPF/10] 00:00:48, metric 1
                       MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.3       st0.1                  Full      172.16.1.3       128    -
172.16.1.2       st0.1                  Full      172.16.1.2       128    -
```

The following commands are entered on spoke 1:

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
```

```
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
578872  UP     fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2          11.1.1.1
```

```
user@host> show security ike security-associations detail
node0:
--------------------------------------------------------------------------
IKE peer 11.1.1.1, Index 578872, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 21.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 28183 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2023
   Output bytes  :                 2030
   Input  packets:                    4
   Output packets:                    4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 21.1.1.2:500, Remote: 11.1.1.1:500
    Local identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
```

```
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 de912bcd 2985/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 98a2b155 2985/ unlim - root 500 11.1.1.1
```

```
user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------

ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 21.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
    Tue Jan 13 2015 12:58:11 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:58:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:58:11 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 2980 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2358 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 2980 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2358 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
```

```
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:11:46, metric 16
                    > to 172.16.1.1 via st0.1
36.1.1.0/24        *[OSPF/10] 00:11:46, metric 26
                    > to 172.16.1.1 via st0.1
172.16.1.1/32      *[OSPF/10] 00:11:46, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.3/32      *[OSPF/10] 00:11:46, metric 25
                    > to 172.16.1.1 via st0.1
224.0.0.5/32       *[OSPF/10] 00:16:52, metric 1
                       MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.1       st0.1                  Full      172.16.1.1       128     -
```

The following commands are entered on spoke 2:

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
2299162 UP     2d58d8fbc396762d  46145be580c68be0  IKEv2          11.1.1.1
```

```
user@host> show security ike security-associations detail
IKE peer 11.1.1.1, Index 2299162, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Initiator, State: UP
```

```
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 31.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 28135 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2023
   Output bytes  :                 2030
   Input  packets:                    4
   Output packets:                    4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 31.1.1.2:500, Remote: 11.1.1.1:500
    Local identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 a9d301b0 2936/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 44ccf265 2936/ unlim - root 500 11.1.1.1
```

```
user@host> show security ipsec security-associations detail
ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 31.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
```

```
  Tunnel events:
    Tue Jan 13 2015 12:57:48 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:57:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:57:48 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2933 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2311 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2933 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2311 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 36 destinations, 36 routes (35 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:00:09, metric 16
                    > to 172.16.1.1 via st0.1
25.1.1.0/24        *[OSPF/10] 00:00:09, metric 26
                    > to 172.16.1.1 via st0.1
172.16.1.1/32      *[OSPF/10] 00:00:09, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.2/32      *[OSPF/10] 00:00:09, metric 25
                    > to 172.16.1.1 via st0.1
224.0.0.5/32       *[OSPF/10] 00:17:52, metric 1
                       MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.1       st0.1                  Full      172.16.1.1       128     -
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. The hub shows two active tunnels, one to each spoke. Each spoke shows an active tunnel to the hub.

If no SAs are listed for IKE Phase 1, then there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

If no SAs are listed for IKE Phase 2, then there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

The `show route protocol ospf` command displays entries in the routing table that were learned from the OSPF protocol. The `show ospf neighbor` command displays information about OSPF neighbors.

Verifying the Shortcut Tunnel Between Partners

Purpose

The AutoVPN hub can act as a shortcut suggester when it notices that traffic is exiting a tunnel with one of its spokes and entering a tunnel with another spoke. A new IPsec SA, or shortcut, is established between the two shortcut partners. On each partner, the route to the network behind its partner now points to the shortcut tunnel instead of to the tunnel between the partner and the suggester (hub).

Action

From operational mode, enter the `show security ike security-associations`, `show security ipsec security-associations`, `show route protocol ospf`, and `show ospf neighbor` commands on the spokes.

The following commands are entered on the hub:

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
10957048 UP    2d58d8fbc396762d  46145be580c68be0  IKEv2          31.1.1.2
10957049 UP    fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2          21.1.1.2
```

```
user@host> show security ike security-associations detail
node0:
```

```
--------------------------------------------------------------------------
IKE peer 31.1.1.2, Index 10957048, Gateway Name: SUGGESTER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Responder, State: UP
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 11.1.1.1:500, Remote: 31.1.1.2:500
  Lifetime: Expires in 27781 seconds
  Peer ike-id: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 260
   Output bytes  :                 548
   Input  packets:                   3
   Output packets:                   3
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 31.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created

IKE peer 21.1.1.2, Index 10957049, Gateway Name: SUGGESTER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Responder, State: UP
```

```
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 11.1.1.1:500, Remote: 21.1.1.2:500
  Lifetime: Expires in 27804 seconds
  Peer ike-id: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 244
   Output bytes  :                 548
   Input  packets:                   3
   Output packets:                   3
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 21.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
s  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <201326593 ESP:aes-cbc-256/sha1 44ccf265 2584/ unlim - root 500 31.1.1.2
  >201326593 ESP:aes-cbc-256/sha1 a9d301b0 2584/ unlim - root 500 31.1.1.2
  <201326594 ESP:aes-cbc-256/sha1 98a2b155 2607/ unlim - root 500 21.1.1.2
  >201326594 ESP:aes-cbc-256/sha1 de912bcd 2607/ unlim - root 500 21.1.1.2
```

```
user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------
```

```
ID: 201326593 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 31.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 13:09:48 -0800: Bind-interface's address received. Information updated (1
times)
    Tue Jan 13 2015 13:09:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
  Direction: inbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2578 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2001 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2578 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2001 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 201326594 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 21.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 13:09:48 -0800: Bind-interface's address received. Information updated (1
times)
    Tue Jan 13 2015 13:09:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
  Direction: inbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 2601 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2023 seconds
```

```
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 2601 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2023 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 28 destinations, 28 routes (27 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

25.1.1.0/24        *[OSPF/10] 00:04:49, metric 11
                    > to 172.16.1.2 via st0.1
36.1.1.0/24        *[OSPF/10] 00:04:49, metric 11
                    > to 172.16.1.3 via st0.1
172.16.1.2/32      *[OSPF/10] 00:04:49, metric 10
                    > to 172.16.1.2 via st0.1
172.16.1.3/32      *[OSPF/10] 00:04:49, metric 10
                    > to 172.16.1.3 via st0.1
224.0.0.5/32       *[OSPF/10] 00:05:10, metric 1
                       MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.3       st0.1                  Full      172.16.1.3       128     -
172.16.1.2       st0.1                  Full      172.16.1.2       128     -
```

The following commands are entered on spoke 1:

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
```

```
578872  UP     fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2          11.1.1.1
578873  UP     895e4d9c7c5da7a4  17de7f18b45139b4  IKEv2          31.1.1.2
```

```
user@host> show security ike security-associations detail
node0:
--------------------------------------------------------------------------
IKE peer 11.1.1.1, Index 578872, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 21.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 27906 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                2495
   Output bytes  :                2274
   Input  packets:                   6
   Output packets:                   7
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 21.1.1.2:500, Remote: 11.1.1.1:500
    Local identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created

IKE peer 31.1.1.2, Index 578873, Gateway Name: PARTNER_GW
```

```
  Auto Discovery VPN:
   Type: Shortcut, Local Capability: Partner, Peer Capability: Partner
  Role: Initiator, State: UP
  Initiator cookie: 895e4d9c7c5da7a4, Responder cookie: 17de7f18b45139b4
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 21.1.1.2:500, Remote: 31.1.1.2:500
  Lifetime: Expires in 28787 seconds
  Peer ike-id: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 1855
   Output bytes  :                 1990
   Input  packets:                    2
   Output packets:                    2
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 21.1.1.2:500, Remote: 31.1.1.2:500
    Local identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 de912bcd 2709/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 98a2b155 2709/ unlim - root 500 11.1.1.1
```

```
  <67108868 ESP:aes-cbc-256/sha1 75d0177b 3590/ unlim - root 500 31.1.1.2
  >67108868 ESP:aes-cbc-256/sha1 e4919d73 3590/ unlim - root 500 31.1.1.2
```

```
user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------

ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 21.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
    Tue Jan 13 2015 12:58:11 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:58:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:58:11 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 2701 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2079 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 2701 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2079 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 67108868 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 21.1.1.2, Remote Gateway: 31.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Auto Discovery VPN:
    Type: Shortcut, Shortcut Role: Initiator
  Version: IKEv2
```

```
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x40608a29
  Tunnel events:
    Tue Jan 13 2015 13:12:52 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 13:12:52 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 13:12:52 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: 75d0177b, AUX-SPI: 0
    Hard lifetime: Expires in 3582 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2959 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: e4919d73, AUX-SPI: 0
    Hard lifetime: Expires in 3582 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2959 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:03:29, metric 16
                    > to 172.16.1.1 via st0.1
36.1.1.0/24        *[OSPF/10] 00:00:35, metric 16
                    > to 172.16.1.3 via st0.1
172.16.1.1/32      *[OSPF/10] 00:03:29, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.3/32      *[OSPF/10] 00:00:35, metric 15
                    > to 172.16.1.3 via st0.1
224.0.0.5/32       *[OSPF/10] 00:20:22, metric 1
                       MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
```

```
172.16.1.3       st0.1                Full      172.16.1.3       128     -
172.16.1.1       st0.1                Full      172.16.1.1       128
```

The following commands are entered on spoke 2:

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
2299162 UP     2d58d8fbc396762d  46145be580c68be0  IKEv2          11.1.1.1
2299163 UP     895e4d9c7c5da7a4  17de7f18b45139b4  IKEv2          21.1.1.2
```

```
user@host> show security ike security-associations detail
IKE peer 11.1.1.1, Index 2299162, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 31.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 27835 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                2571
   Output bytes  :                2290
   Input  packets:                   7
   Output packets:                   7
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 31.1.1.2:500, Remote: 11.1.1.1:500
```

```
    Local identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created

IKE peer 21.1.1.2, Index 2299163, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Shortcut, Local Capability: Partner, Peer Capability: Partner
  Role: Responder, State: UP
  Initiator cookie: 895e4d9c7c5da7a4, Responder cookie: 17de7f18b45139b4
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 31.1.1.2:500, Remote: 21.1.1.2:500
  Lifetime: Expires in 28739 seconds
  Peer ike-id: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2066
   Output bytes  :                 1931
   Input  packets:                    3
   Output packets:                    3
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 31.1.1.2:500, Remote: 21.1.1.2:500
    Local identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 a9d301b0 2638/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 44ccf265 2638/ unlim - root 500 11.1.1.1
```

```
  <67108868 ESP:aes-cbc-256/sha1 e4919d73 3542/ unlim - root 500 21.1.1.2
  >67108868 ESP:aes-cbc-256/sha1 75d0177b 3542/ unlim - root 500 21.1.1.2
```

```
user@host> show security ipsec security-associations detail
ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 31.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
    Tue Jan 13 2015 12:57:48 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:57:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:57:48 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2632 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2010 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2632 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2010 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 67108868 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 31.1.1.2, Remote Gateway: 21.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Auto Discovery VPN:
    Type: Shortcut, Shortcut Role: Responder
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x40608aa9
  Tunnel events:
```

```
    Tue Jan 13 2015 13:12:52 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 13:12:52 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 13:12:52 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: e4919d73, AUX-SPI: 0
    Hard lifetime: Expires in 3536 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2958 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 75d0177b, AUX-SPI: 0
    Hard lifetime: Expires in 3536 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2958 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 36 destinations, 36 routes (35 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:03:55, metric 16
                    > to 172.16.1.1 via st0.1
25.1.1.0/24        *[OSPF/10] 00:01:02, metric 16
                    > to 172.16.1.2 via st0.1
172.16.1.1/32      *[OSPF/10] 00:03:55, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.2/32      *[OSPF/10] 00:01:02, metric 15
                    > to 172.16.1.2 via st0.1
224.0.0.5/32       *[OSPF/10] 00:21:38, metric 1
                       MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.2       st0.1                  Full      172.16.1.2       128     -
172.16.1.1       st0.1                  Full      172.16.1.1       128     -
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. The hub still shows two active tunnels, one to each spoke. Each spoke shows two active tunnels, one to the hub and one to its shortcut partner.

The `show route protocol ospf` command shows the addition of routes to the partner and to the hub.

Example: Configuring ADVPN with OSPFv3 for IPv6 Traffic

IN THIS SECTION

- Requirements | **879**
- Overview | **880**
- Configuration | **883**
- Verification | **912**

This example shows how to configure an ADVPN hub and two spokes to create a shortcut tunnel and change the routing topology for the host to reach the other side without sending traffic through the hub. This example configures ADVPN for IPv6 environment using OSPFv3 to forward packets through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Three supported firewalls as ADVPN hub and spokes
- Supported Junos OS Release for the firewall that runs the kmd process.
- Supported Junos OS Release for the firewall that runs the iked process.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

- Topology | 882

This example shows the configuration of an ADVPN hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the ADVPN hub and all spokes must have the same values. Table 95 on page 880 shows the options used in this example.

Table 95: Phase 1 and Phase 2 Options for ADPN Hub and Spoke Basic OSPFv3 Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	19
Authentication algorithm	SHA-384
Encryption algorithm	AES 256 CBC
IKE policy:	
Mode	Main

Table 95: Phase 1 and Phase 2 Options for ADPN Hub and Spoke Basic OSPFv3 Configurations *(Continued)*

Option	Value
IPsec proposal:	
Protocol	ESP
Lifetime seconds	3000
Encryption algorithm	AES 256 GCM
IPsec policy:	
Perfect Forward Secrecy (PFS) group	19

The same certificate authority (CA) is configured on all devices.

Table 96 on page 881 shows the options configured on the hub and on all spokes.

Table 96: ADVPN OSPFv3 Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	2001:db8:2000::1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string SLT in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	reth1	Spoke 1: ge-0/0/0.0 Spoke 2: ge-0/0/0.0

Table 96: ADVPN OSPFv3 Configuration for Hub and All Spokes *(Continued)*

Option	Hub	All Spokes
VPN:		
Bind interface	st0.1	st0.1
Establish tunnels	(not configured)	establish-tunnels immediately

Table 97 on page 882 shows the configuration options that are different on each spoke.

Table 97: Comparison Between the OSPFv3 Spoke Configurations

Option	Spoke 1	Spoke 2
st0.1 interface	2001:db8:9000::2/64	2001:db8:9000::3/64
Interface to internal network	(ge-0/0/1.0) 2001:db8:4000::1/64	(ge-0/0/1.0) 2001:db8:6000::1/64
Interface to Internet	(ge-0/0/0.0) 2001:db8:3000::2/64	(ge-0/0/0.0) 2001:db8:5000::2/64

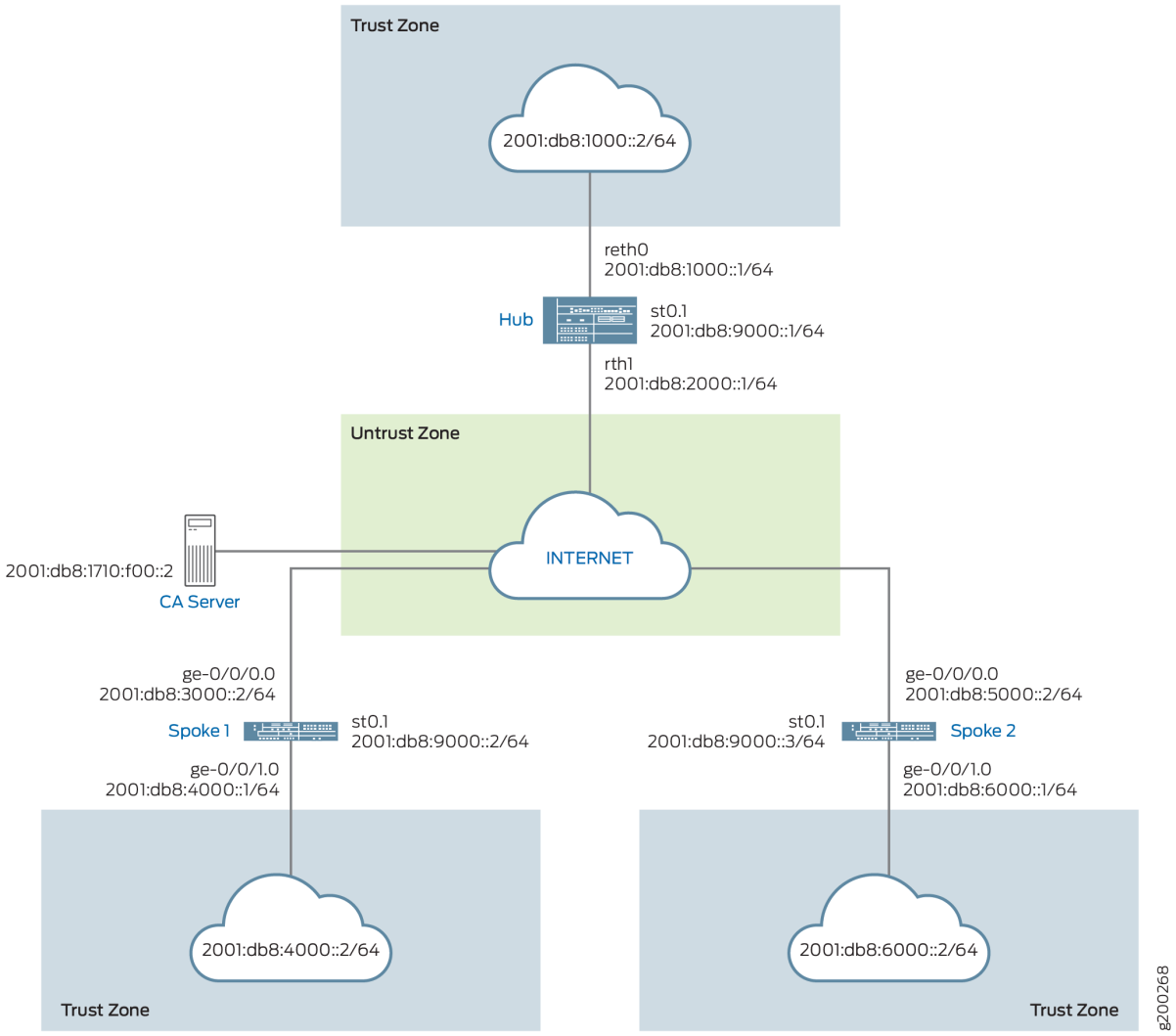
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. .

Topology

Figure 55 on page 883 shows the firewalls to be configured for ADVPN in this example.

Figure 55: ADVPN Deployment with OSPFv3



Configuration

IN THIS SECTION

- Enroll Device Certificates with SCEP | **884**
- Configuring the Hub | **889**
- Configuring Spoke 1 | **898**
- Configuring Spoke 2 | **905**

To configure ADVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
```

```
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Tumkur, Common name: spoke2, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
  Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 6-2012 10:02
    Not after: 11- 6-2013 10:12
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
    27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
    77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
    44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
    7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
```

```
    7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
    58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
    00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set chassis cluster reth-count 2
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
```

```
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate HUB
set security ike gateway IKE_GWA_1 ike-policy IKE_POL
set security ike gateway IKE_GWA_1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway IKE_GWA_1 dynamic ike-user-type group-ike-id
set security ike gateway IKE_GWA_1 dead-peer-detection always-send
set security ike gateway IKE_GWA_1 dead-peer-detection interval 10
set security ike gateway IKE_GWA_1 dead-peer-detection threshold 3
set security ike gateway IKE_GWA_1 local-identity distinguished-name
set security ike gateway IKE_GWA_1 external-interface reth1
set security ike gateway IKE_GWA_1 advpn partner disable
set security ike gateway IKE_GWA_1 version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPNA_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPNA_1 ike gateway IKE_GWA_1
set security ipsec vpn IPSEC_VPNA_1 ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces reth0.0
set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces ge-0/0/1 gigether-options redundant-parent reth0
set interfaces ge-7/0/0 gigether-options redundant-parent reth1
set interfaces ge-7/0/1 gigether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet
set interfaces reth0 unit 0 family inet6 address 2001:db8:1000::1/64
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet
set interfaces reth1 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:9000::1/64
set routing-options rib inet6.0 static route 2001:db8:3000::0/64 next-hop 2001:db8:2000::2
set routing-options rib inet6.0 static route 2001:db8:5000::0/64 next-hop 2001:db8:2000::2
```

```
set protocols ospf3 area 0.0.0.0 interface reth0.0
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 gigether-options redundant-parent reth1
user@host# set ge-0/0/1 gigether-options redundant-parent reth0
user@host# set ge-7/0/0 gigether-options redundant-parent reth1
user@host# set ge-7/0/1 gigether-options redundant-parent reth0
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet
user@host# set reth0 unit 0 family inet6 address 2001:db8:1000::1/64
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet
user@host# set reth1 unit 0 family inet6 address 2001:db8:2000::1/64
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet6 address 2001:db8:9000::1/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set ospf3 area 0.0.0.0 interface reth0.0
user@host# set ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:3000::0/64 next-hop 2001:db8:2000::2
user@host# set rib inet6.0 static route 2001:db8:5000::0/64 next-hop 2001:db8:2000::2
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
```

```
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate HUB
[edit security ike gateway IKE_GWA_1]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set ike-user-type group-ike-id
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set external-interface reth1
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_1
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces reth1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
```

```
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces reth0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set pki ca-profile ROOT-CA revocation-check disable
```

8. Configure chassis cluster

```
[edit chassis cluster]
set reth-count 2
set node 0
set node 1
set redundancy-group 0 node 0 priority 254
set redundancy-group 0 node 1 priority 1
set redundancy-group 1 node 0 priority 254
set redundancy-group 1 node 1 priority 1
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show`

`security pki` `show chassis cluster` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-0/0/1 {
    gigether-options {
        redundant-parent reth0;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet;
            family inet6 {
                address 2001:db8:1000::1/64;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet;
            family inet6 {
                address 2001:db8:2000::1/64;
            }
        }
    }
    st0 {
        unit 1 {
            multipoint;
            family inet6 {
                address 2001:db8:9000::1/64 {
```

```
                    primary;
                }
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
        interface reth0.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route 2001:db8:3000::/64 next-hop 2001:db8:2000::2;
        route 2001:db8:5000::/64 next-hop 2001:db8:2000::2;
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate HUB;
    }
}
gateway IKE_GWA_1 {
```

```
    ike-policy IKE_POL;
    dynamic {
        distinguished-name {
            wildcard OU=SLT;
        }
        ike-user-type group-ike-id;
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface reth1;
    advpn {
        partner {
            disable;
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_1;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
```

```
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        st0.1;
        reth1.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        reth0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
```

```
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE1
set security ike gateway IKE_GW_SPOKE_1 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_1 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_1 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_1 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_1 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_1 advpn suggester disable
set security ike gateway IKE_GW_SPOKE_1 version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
```

```
set security ipsec vpn IPSEC_VPN_SPOKE_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike gateway IKE_GW_SPOKE_1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_1 establish-tunnels immediately
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:9000::2/64
set routing-options rib inet6.0 static route 2001:db8:2000::0/64 next-hop 2001:db8:3000::1
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet6 address 2001:db8:9000::2/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
set area 0.0.0.0 interface ge-0/0/1.0
set area 0.0.0.0 interface st0.1 interface-type p2mp
set area 0.0.0.0 interface st0.1 dynamic-neighbors
```

```
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE1
[edit security ike gateway IKE_GW_SPOKE_1]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
user@host# set advpn suggester disable
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP1]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GW_SPOKE_1
```

```
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show`

`security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:3000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:4000::1/64;
        }
    }
}
st0 {
    unit 1 {
        multipoint;
        family inet6 {
            address 2001:db8:9000::2/64;
        }
    }
}
[edit]
user@host# show protocols
ospf3 {
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
```

```
        route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE1;
    }
}
gateway IKE_GW_SPOKE_1 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    advpn {
        suggester {
            disable;
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
```

```
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_1;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        st0.1;
        ge-0/0/0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
         ge-0/0/1.0;
    }
```

```
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
```

```
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE2
set security ike gateway IKE_GW_SPOKE_2 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_2 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_2 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_2 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_2 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_2 advpn suggester disable
set security ike gateway IKE_GW_SPOKE_2 version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_2 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike gateway IKE_GW_SPOKE_2
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_2 establish-tunnels immediately
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:9000::3/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:9000::3/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/1.0
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE2
[edit security ike gateway IKE_GW_SPOKE_2]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
user@host# set advpn suggester disable
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROPl]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_2]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GW_SPOKE_2
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
```

```
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:5000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:6000::1/64;
        }
    }
}
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:9000::3/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            dynamic-neighbors;
        }
```

```
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE2;
    }
}
gateway IKE_GW_SPOKE_2 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    advpn {
        suggester {
        disable
        }
    }
    version v2-only;
```

```
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_2 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_2;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
```

```
            protocols {
                ospf3;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Status | **913**
- Verifying IPsec Status | **913**
- Verifying IPsec Next-Hop Tunnels | **914**
- Verifying OSPFv3 | **915**

Confirm that the configuration is working properly.

Verifying IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the **show security ike sa** command.

```
user@host> show security ike sa
Index   State Initiator cookie          Responder cookie          Mode Remote Address

4295070 UP    2001:db8:1ad4ba7a115fa229 2001:db8:32e6382a058bb296 Main 2001:db8:3000::2

295069  UP    2001:db8:88a1520c20cbbe04 2001:db8:7fa4c8e365393c48 Main 2001:db8:5000::2
```

Meaning

The `show security ike sa` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec sa** command.

```
user@host> show security ipsec sa
Total active tunnels: 2     Total Ipsec sas: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108881 ESP:aes-gcm-256/None 3dba3f80 2979/ unlim - root 500 2001:db8:5000::2
  >67108881 ESP:aes-gcm-256/None 46746d5d 2979/ unlim - root 500 2001:db8:5000::2
```

```
<67108882 ESP:aes-gcm-256/None 16dceb60 2992/ unlim - root 500 2001:db8:3000::2
>67108882 ESP:aes-gcm-256/None 681209c2 2992/ unlim - root 500 2001:db8:3000::2
```

Meaning

The `show security ipsec sa` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway              interface  IPSec VPN name Flag IKE-ID
XAUTH username
2001:db8:9000::2              st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available
2001:db8:9000::3              st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
2001:db8::5668:ad10:fcd8:10c8 st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
2001:db8::5668:ad10:fcd8:112f st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying OSPFv3

Purpose

Verify that OSPFv3 references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show ospf3 neighbor interface** command.

```
user@host> show ospf3 neighbor interface
ID                        Interface              State   Pri   Dead
2001:db8:9000:2   st0.1                  Full    128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:110e


2001:db8:20:54:49.693           INFO     ${ret} = ID      Interface  State Pri Dead
2001:db8:9000:3   st0.1                  Full    128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:110e
```

Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established

IN THIS SECTION

- Problem | 915
- Solution | 916

Problem

Description

OSPF can take up to 9 seconds to update a shortcut route in the routing table. It can take up to 10 seconds before traffic is forwarded to the shortcut tunnel.

Symptoms

When a shortcut tunnel is established between two shortcut partners, OSPF initiates an OSPF hello packet. Because of the timing of the shortcut tunnel establishment and the OSPF neighbor installation, the first packet in the tunnel might be dropped. This can cause OSPF to try again to establish an OSPF adjacency.

By default, the interval at which the OSPF retries to establish an adjacency is 10 seconds. After a shortcut tunnel is established, it can take more than 10 seconds for OSPF to establish an adjacency between the partners.

Solution

Configuring a smaller retry interval, such as 1 or 2 seconds, can enable OSPF to establish adjacencies faster over the shortcut tunnel. For example, use the following configurations:

```
[edit]
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
```

Platform-Specific Multicast in ADVPN Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform.

Table 98: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX300, SRX320, SRX340, SRX345, SRX1500, and vSRX 3.0 devices that support multicast, to enable PIM on st0 `p2mp` interface, you must run IPsec VPN service with the kmd process. • On SRX1500, SRX1600, SRX2300, SRX4100, SRX4120, SRX4200, SRX4300, SRX4600, SRX4700, and vSRX 3.0 devices that support multicast, you can enable PIM on st0 `p2mp` interface when the IPsec VPN service runs with the iked process. • SRX400, SRX440, and SRX440-2AC do not support multicast.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.2R1	Support for IPv6 address with ADVPN for firewalls running the iked process is added in Junos OS Release 24.2R1.
24.2R1	Support for multicast traffic (IPv4 address) with ADVPN for firewalls running the iked process is added in Junos OS Release 24.2R1.
23.4R1	Support for ADVPN with firewalls running the iked process is added in Junos OS Release 23.4R1.
19.2R1	Starting in Junos OS Release 19.2R1, PIM using P2MP mode supports ADVPN in which a new `p2mp` interface type is introduced for PIM for IPsec VPN with kmd process.
18.1R1	Starting with Junos OS Release 18.1R1, ADVPN supports IPv6 with the kmd process.

RELATED DOCUMENTATION

IPv6 IPsec VPNs

9

CHAPTER

AutoVPN

IN THIS CHAPTER

- AutoVPN on Hub-And-Spoke Devices | **919**

AutoVPN on Hub-And-Spoke Devices

SUMMARY

Learn about AutoVPN and how to configure the feature on the firewall.

IN THIS SECTION

- Understanding AutoVPN | **920**
- Understanding Spoke Authentication in AutoVPN Deployments | **925**
- AutoVPN Configuration Overview | **928**
- Example: Configuring Basic AutoVPN with iBGP | **929**
- Example: Configuring Basic AutoVPN with iBGP for IPv6 Traffic | **966**
- Example: Configuring AutoVPN with iBGP and ECMP | **1004**
- Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels | **1037**
- Example: Configuring Basic AutoVPN with OSPF | **1074**
- Example: Configuring AutoVPN with OSPFv3 for IPv6 Traffic | **1107**
- Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors | **1144**
- Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors | **1165**
- Example: Configuring AutoVPN with Pre-Shared Key | **1193**
- Configure Multicast Support on P2MP Infrastructure | **1198**
- Platform-Specific AutoVPN Behavior | **1199**
- Additional Platform Information | **1200**

AutoVPN supports an IPsec VPN aggregator (known as a hub) that serves as a single termination point for multiple tunnels to remote sites (known as spokes). AutoVPN allows network administrators to configure a hub for current and future spokes.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific AutoVPN Behavior" on page 1199 section for notes related to your platform.

See the "Additional Platform Information" on page 1200 section for more information.

Understanding AutoVPN

IN THIS SECTION

- Secure Tunnel Modes | 920
- Authentication | 921
- Configuration and Management | 922
- Multicast Support Using PIM | 922
- Understanding AutoVPN Limitations | 924
- Understanding AutoVPN with Traffic Selectors | 924

AutoVPN supports an IPsec VPN aggregator (known as a *hub*) that serves as a single termination point for multiple tunnels to remote sites (known as *spokes*). AutoVPN allows network administrators to configure a hub for current and future spokes. No configuration changes are required on the hub when spoke devices are added or deleted, thus allowing administrators flexibility in managing large-scale network deployments.

Secure Tunnel Modes

AutoVPN is supported on route-based IPsec VPNs. For route-based VPNs, you configure a secure tunnel (st0) interface and bind it to an IPsec VPN tunnel. st0 interfaces in AutoVPN networks can be configured in one of two modes:

- Point-to-point mode—By default, a st0 interface configured at the [`edit interfaces st0 unit` *x*] hierarchy level is in point-to-point mode.
- Point-to-multipoint mode—In this mode, the `multipoint` option is configured at the [`edit interfaces st0 unit` *x*] hierarchy level on both AutoVPN hub and spokes. st0 interfaces on the hub and spokes must be numbered and the IP address configured on a spoke must exist in the hub's st0 interface subnetwork.

Table 99 on page 921 compares AutoVPN point-to-point and point-to-multipoint secure tunnel interface modes.

Table 99: Comparison Between AutoVPN Point-to-Point and Point-to-Multipoint Secure Tunnel Modes

Point-to-Point Mode	Point-to-Multipoint Mode
Supports IKEv1 or IKEv2.	Supports IKEv1 or IKEv2.
Supports IPv4 and IPv6 traffic.	Supports IPv4 or IPv6.
Traffic selectors	Dynamic routing protocols (OSPF, OSPFv3 and iBGP)
Dead peer detection	Dead peer detection
Allows spoke devices to be Juniper Networks firewall or third-party devices.	This mode is only supported with Juniper Networks firewalls.

Authentication

AutoVPNs support both certificate and preshared key based authentication methods.

For certificate based authentication in AutoVPN hubs and spokes, you can use X.509 public key infrastructure (PKI) certificates. The group IKE user type configured on the hub allows strings to be specified to match the alternate subject field in spoke certificates. Partial matches for the subject fields in spoke certificates can also be specified.

We support AutoVPN with the following two options:

- **AutoVPN seeded PSK**: Multiple peers connecting to same gateway having different pre-shared key.
- **AutoVPN shared PSK**: Multiple peers connecting to same gateway having same pre-shared key.

Seeded PSK is different from non-seeded PSK (that is, same shared PSK). Seeded PSK uses master key to generate the shared PSK for the peer. So each peer will have different PSK connecting to the same gateway. For example: Consider a scenario where peer 1 with the IKE ID *user1@juniper.net* and peer 2 with IKE ID *user2@juniper.net* attempts to connect to gateway. In this scenario the gateway that is configured as `HUB_GW` containing the master key configured as `ThisIsMySecretPreSharedkey` will have the different PSK as follows:

Peer 1 : `79e4ea39f5c06834a3c4c031e37c6de24d46798a`

Peer 2: 3db8385746f3d1e639435a882579a9f28464e5c7

This means, for different users with different user id and same master key will generate a different or unique preshared key.

You can use either seeded-pre-shared-key or pre-shared-key for Auto-VPN PSK:

- **Different preshared key**: If the seeded-pre-shared-key is set, different IKE preshared key is used by the VPN gateway to authenticate each remote peer. The peer preshared keys are generated using the master-key set in the IKE gateway and shared across the peers.

  To enable the VPN gateway to use a different IKE preshared key (PSK) for authenticating each remote peer, use the new CLI commands seeded-pre-shared-key *ascii-text* or seeded-pre-shared-key *hexadecimal* under the [edit security ike policy *policy_name*] hierarchy level.

  This command is mutually exclusive with pre-shared-key command under the same hierarchy.

- **Shared/Same preshared key**: If pre-shared-key-type is not configured, then the PSK is considered to be shared. Same IKE preshared key is used by the VPN gateway to authenticate all remote peers.

  To enable the VPN gateway to use the same IKE PSK for authenticating all remote peers, use the existing CLI commands pre-sharedkey *ascii-text* or pre-shared-key *hexadecimal* .

At the VPN gateway, you can bypass the IKE ID validation using the general-ikeid configuration statement under the [edit security ike gateway *gateway_name* dynamic] hierarchy level. If this option is configured, then during authentication of remote peer, the VPN gateway allows any remote IKE ID connection.

See "Additional Platform Information" on page 1200 to know about AutoVPN PSK support with different IKE modes in firewalls running the iked process with the junos-ike package.

See "Example: Configuring AutoVPN with Pre-Shared Key" on page 1193.

Configuration and Management

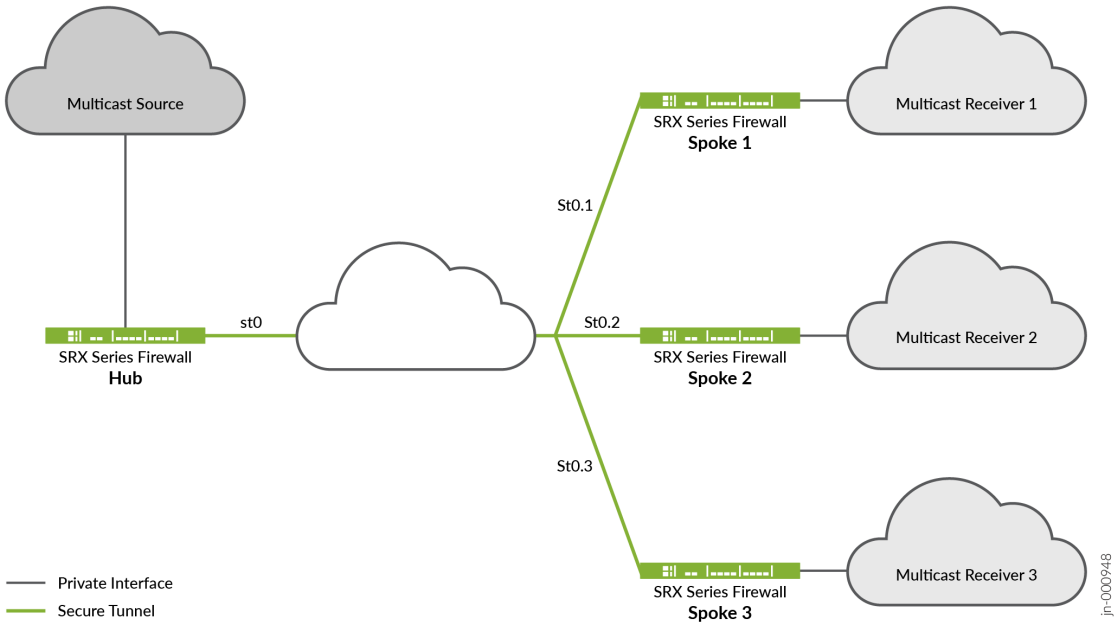
AutoVPN is configured and managed on firewalls using the CLI. Multiple AutoVPN hubs can be configured on a single firewall. The maximum number of spokes supported by a configured hub is specific to the model of the firewall.

Multicast Support Using PIM

IP multicast delivers traffic to more than one intended receivers by replicating the data packets. You can use multicast data for applications such as video streaming. Your firewall supports Protocol Independent Multicast (PIM) in point-to-multipoint (P2MP) mode. You can enable PIM on the firewall's secure tunnel, st0, interface with P2MP mode. The protocol detects the P2MP interface from the interface configuration and supports multicast traffic. To understand PIM, see PIM Overview.

Figure 56 on page 923 illustrates multicast topology in P2MP infrastructure.

Figure 56: Multicast Topology in P2MP Infrastructure



The topology shows that one of the firewalls acting as a hub and the rest of the three acting as spokes. You can also have two spokes in your topology. Typically, the multicast sender resides behind the hub, while the multicast receivers are behind the spokes. For multicast support, notice that the secure tunnel st0 logical interface on the hub-and-spoke devices are configured with PIM P2MP mode. On each of these devices, the st0 P2MP interface tracks all PIM joins per neighbor to ensure that the multicast forwarding or replication happens only to those neighbors that are in joined state.

The firewalls support IP multicast traffic in PIM sparse mode over the st0 P2MP interfaces. The hub acts as the first-hop router (FHR) or the rendezvous point (RP). The spokes can act as the last-hop routers (LHR) in the P2MP network. The devices in the network replicate the multicast data packets to neighbors that join the multicast group.

Note the following considerations when you configure multicast traffic support:

- You cannot configure IPv6 multicast on P2MP interfaces.
- For IP multicast configuration to work, you must disable PowerMode IPsec (PMI).
- You cannot perform multicast ping from or to P2MP interfaces.
- Note that IGMP is enable by default when you enable PIM, but it doesn't work on P2MP interface.

For details on how to configure multicast support on P2MP infrastructure, see "Configure Multicast Support on P2MP Infrastructure" on page 1198.

Understanding AutoVPN Limitations

The following features are not supported for AutoVPN:

- Policy-based VPNs are not supported.
- The RIP dynamic routing protocol is not supported with AutoVPN tunnels.
- Manual keys and Autokey IKE with preshared keys are not supported.
- Configuring static next-hop tunnel binding (NHTB) on the hub for spokes is not supported.
- IPv6 multicast is not supported.
- The group IKE ID user type is not supported with an IP address as the IKE ID.
- When the group IKE ID user type is used, the IKE ID should not overlap with other IKE gateways configured on the same external interface.

Understanding AutoVPN with Traffic Selectors

AutoVPN hubs can be configured with multiple traffic selectors to protect traffic to spokes. This feature provides the following benefits:

- A single VPN configuration can support many different peers.
- VPN peers can be third-party devices.
- A single peer can establish multiple tunnels with the same VPN.
- A larger number of tunnels can be supported than with AutoVPN with dynamic routing protocols.

AutoVPN networks that use secure tunnel interfaces in point-to-point mode support IPv6 addresses for traffic selectors and for IKE peers.

When the hub-to-spoke tunnel is established, the hub uses *auto route insertion (ARI)*, known in previous releases as *reverse route insertion (RRI)*, to insert the route to the spoke prefix in its routing table. The ARI route can then be imported to routing protocols and distributed to the core network.

AutoVPN with traffic selectors can be configured with the secure tunnel (st0) interface in point-to-point mode for both IKEv1 and IKEv2.

Dynamic routing protocols are not supported on st0 interfaces when traffic selectors are configured.

Note the following caveats when configuring AutoVPN with traffic selectors:

- Dynamic routing protocols are not supported with traffic selectors with st0 interfaces in point-to-point mode.
- Auto Discovery VPN and IKEv2 configuration payload cannot be configured with AutoVPN with traffic selectors.
- Spokes can be third-party devices; however, note the following differences:
  - In IKEv2, a third-party spoke can propose multiple traffic selectors in a single SA negotiation. This is not supported on Juniper Networks firewalls and the negotiation is rejected.
  - A third-party spoke can identify specific ports or protocols for traffic selector use. Ports and protocols are not supported with traffic selectors on Juniper Networks firewalls and the negotiation is rejected.

SEE ALSO

Understanding Spoke Authentication in AutoVPN Deployments | 925

Understand Traffic Selectors in Route-Based VPNs | 451

Example: Configure Traffic Selectors in a Route-Based VPN | 458

Understanding Spoke Authentication in AutoVPN Deployments

IN THIS SECTION

Group IKE ID Configuration on the Hub | 926

Excluding a Spoke Connection | 928

In AutoVPN deployments, the hub and spoke devices must have valid X.509 PKI certificates loaded. You can use the `show security pki local-certificate detail` command to display information about the certificates loaded in a device.

This topic covers the configuration on the hub that allows spokes to authenticate and connect to the hub using certificates:

Group IKE ID Configuration on the Hub

The group IKE ID feature allows a number of spoke devices to share an IKE configuration on the hub. The certificate holder's identification, in the subject or alternate subject fields in each spoke's X.509 certificate, must contain a part that is common to all spokes; the common part of the certificate identification is specified for the IKE configuration on the hub.

For example, the IKE ID `example.net` can be configured on the hub to identify spokes with the hostnames `device1.example.net`, `device2.example.net`, and `device3.example.net`. The certificate on each spoke must contain a hostname identity in the alternate subject field with `example.net` in the right-most part of the field; for example, `device1.example.net`. In this example, all spokes use this hostname identity in their IKE ID payload. During IKE negotiation, the IKE ID from a spoke is used to match the common part of the peer IKE identity configured on the hub. A valid certificate authenticates the spoke.

The common part of the certificate identification can be one of the following:

- A partial hostname in the right-most part of the alternate subject field of the certificate, for example `example.net`.
- A partial e-mail address in the right-most part of the alternate subject field of the certificate, for example `@example.net`.
- A container string, a set of wildcards, or both to match the subject fields of the certificate. The subject fields contain details of the digital certificate holder in Abstract Syntax Notation One (ASN.1) distinguished name (DN) format. Fields can include organization, organizational unit, country, locality, or common name.

  To configure a group IKE ID to match subject fields in certificates, you can specify the following types of identity matches:

  - Container—The hub authenticates the spoke's IKE ID if the subject fields of the spoke's certificate exactly match the values configured on the hub. Multiple entries can be specified for each subject field (for example, `ou=eng,ou=sw`). The order of values in the fields must match.
  - Wildcard—The hub authenticates the spoke's IKE ID if the subject fields of the spoke's certificate match the values configured on the hub. Wildcard matching supports only one value per DN field in the configuration. For example, you can specify either `ou=eng` or `ou=sw`. The CLI doesn't support multiple values such as `ou=eng, ou=sw`. The order of DN fields does not affect the matching behavior.

The following example configures a group IKE ID with the partial hostname `example.net` in the alternate subject field of the certificate.

```
[edit]
security {
```

```
    ike {
        policy common-cert-policy {
            proposals common-ike-proposal;
            certificate {
                local-certificate hub-local-certificate;
            }
        }
        gateway common-gateway-to-all-spoke-peer {
            ike-policy common-cert-policy;
            dynamic {
                hostname example.net;
                ike-user-type group-ike-id;
            }
            external-interface fe-0/0/2;
        }
    }
}
```

In this example, `example.net` is the common part of the hostname identification used for all spokes. All X.509 certificates on the spokes must contain a hostname identity in the alternate subject field with `example.net` in the right-most part. All spokes must use the hostname identity in their IKE ID payload.

The following example configures a group IKE ID with wildcards to match the values `sales` in the organizational unit and `example` in the organization subject fields of the certificate.

```
[edit]
security {
    ike {
        policy common-cert-policy {
            proposals common-ike-proposal;
            certificate {
                local-certificate hub-local-certificate;
            }
        }
        gateway common-gateway-to-all-spoke-peer {
            ike-policy common-cert-policy;
            dynamic {
                distinguished-name {
                    wildcard ou=sales,o=example;
                }
                ike-user-type group-ike-id;
            }
```

```
            external-interface fe-0/0/2;
        }
    }
}
```

In this example, the fields `ou=sales,o=example` are the common part of the subject field in the certificates expected from the spokes. During IKE negotiation, if a spoke presents a certificate with the subject fields `cn=alice,ou=sales,o=example` in its certificate, authentication succeeds and the tunnel is established. If a spoke presents a certificate with the subject fields `cn=thomas,ou=engineer,o=example` in its certificate, the certificate is rejected by the hub as the organization unit should be `sales`.

Excluding a Spoke Connection

To exclude a particular spoke from connecting to the hub, the certificate for that spoke must be revoked. The hub needs to retrieve the latest certificate revocation list (CRL) from the CA that contains the serial number of the revoked certificate. The hub will then refuse a VPN connection from the revoked spoke. Until the latest CRL is available in the hub, the hub might continue to establish a tunnel from the revoked spoke. For more information, see Enroll Certificate and Certificate Authority Profiles.

SEE ALSO

> IPsec VPN with Autokey IKE Configuration Overview

AutoVPN Configuration Overview

The following steps describe the basic tasks for configuring AutoVPN on hub and spoke devices. The AutoVPN hub is configured *once* for all current and new spokes.

To configure the AutoVPN hub:

1. Enroll a CA certificate and the local certificate in the device.
   - You can use preshared key based authentication if you do not have CA certificates.
2. Create a secure tunnel (st0) interface and configure it in point-to-multipoint mode.
3. Configure a single IKE policy.
4. Configure an IKE gateway with a group IKE ID that is common to all spokes.
5. Configure a single IPsec policy and VPN.
6. Configure a dynamic routing protocol.

To configure a Juniper Networks firewall AutoVPN spoke device:

1. Enroll a CA certificate and the local certificate in the device.
   - Use the preshared key based authentication method, if you configure preshared key authentication on the hub.
2. Create an st0 interface and configure it in point-to-multipoint mode.
3. Configure an IKE policy to match the IKE policy configured on the hub.
4. Configure an IKE gateway with an ID to match the group IKE ID configured on the hub.
5. Configure an IPsec policy to match the IPsec policy configured on the hub.
6. Configure a dynamic routing protocol.

The examples listed in this topic use firewalls running Junos OS for the hub and spoke configurations. If your spoke devices are not running Junos OS, you need to configure Next-Hop Tunnel Binding.

SEE ALSO

Next-Hop Based Tunnels for Layer 3 VPNs

Example: Configuring Basic AutoVPN with iBGP

IN THIS SECTION

- Requirements | 930
- Overview | 930
- Configuration | 934
- Verification | 962

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures iBGP to forward packets through the VPN tunnels and uses certificate based authentication.

For authentication with preshared key, see ‘Configure Phase 1 options’ step at " hub" on page 941 to configure the hub, " spoke1" on page 949 to configure the spoke1, and the " spoke2" on page 956 to configure the spoke2.

Requirements

This example uses the following hardware and software components:

- Three supported Juniper Networks firewalls as AutoVPN hub and spokes
- Junos OS Release that supports AutoVPN

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels. For more information about specific requirements for a dynamic routing protocol, see the Routing Protocols Overview.

Overview

IN THIS SECTION

Topology | 933

This example shows the configuration of an AutoVPN hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 100 on page 930 shows the options used in this example.

Table 100: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations

Option	Value
IKE proposal:	

Table 100: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations *(Continued)*

Option	Value
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 101 on page 932 shows the options configured on the hub and on all spokes.

Table 101: AutoVPN Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	10.1.1.1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string SLT in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/1.0	Spoke 1: fe-0/0/1.0 Spoke 2: ge-0/0/1.0
VPN:		
Bind interface	st0.0	st0.0
Establish tunnels	(not configured)	Immediately on configuration commit

Table 102 on page 932 shows the configuration options that are different on each spoke.

Table 102: Comparison Between the Spoke Configurations

Option	Spoke 1	Spoke 2
st0.0 interface	10.10.10.2/24	10.10.10.3/24
Interface to internal network	(fe-0.0/4.0) 10.60.60.1/24	(fe-0.0/4.0) 10.70.70.1/24
Interface to Internet	(fe-0/0/1.0) 10.2.2.1/30	(ge-0/0/1.0) 10.3.3.1/30

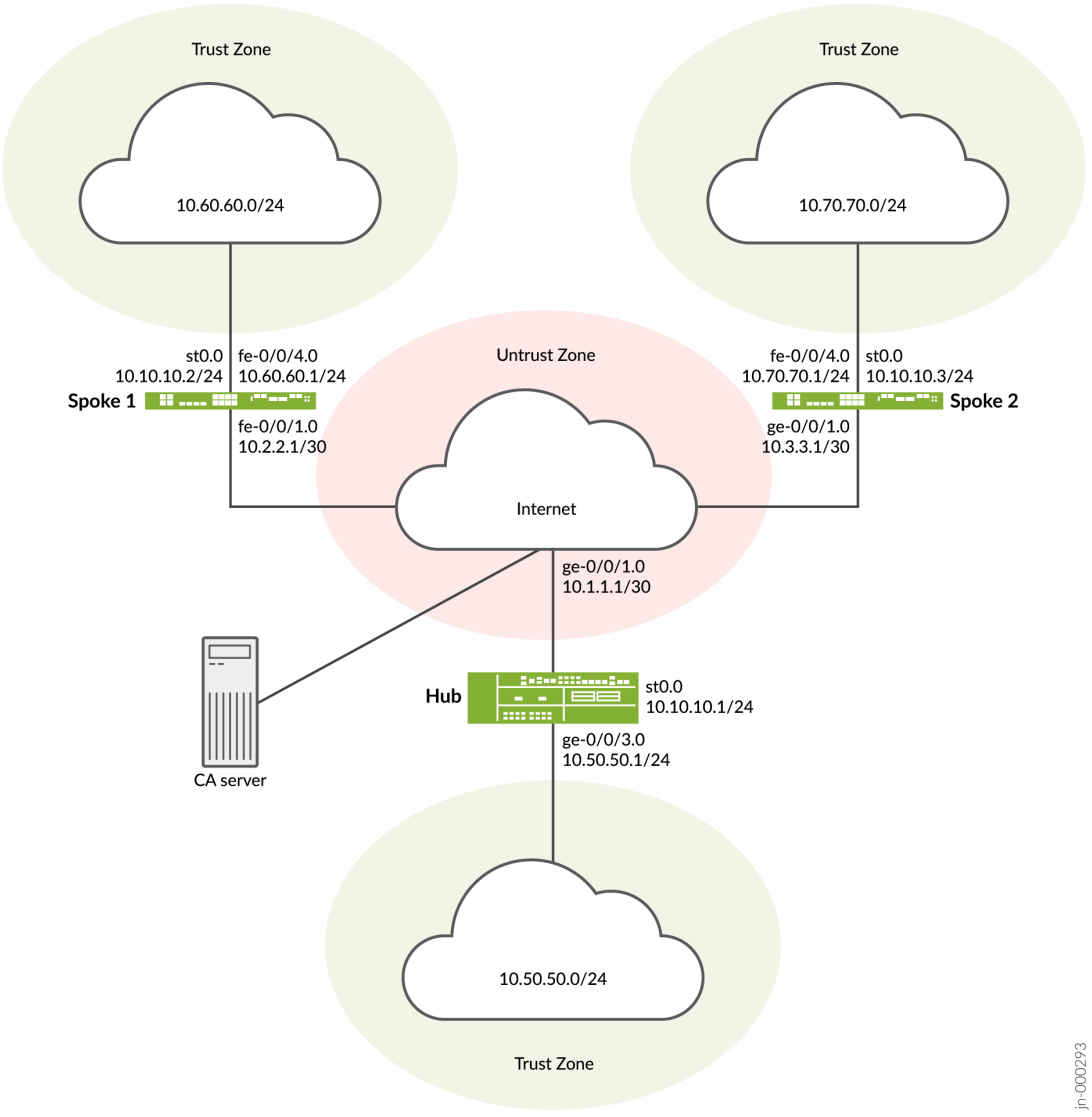
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments.

Topology

Figure 57 on page 933 shows the firewalls to be configured for AutoVPN in this example.

Figure 57: Basic AutoVPN Deployment with iBGP



Configuration

IN THIS SECTION

- Enroll Device Certificates with SCEP | **934**
- Configuring the Hub | **939**
- Configuring Spoke 1 | **948**
- Configuring Spoke 2 | **955**

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices. Ignore this step, if you are using PSK.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
```

```
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
```

```
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Tumkur, Common name: spoke2, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
  Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 6-2012 10:02
    Not after: 11- 6-2013 10:12
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
    27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
    77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
    44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
    7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
    7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
    58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
    00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
```

```
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.10.10.1
set protocols bgp group ibgp export lan_nw
set protocols bgp group ibgp cluster 10.2.3.4
set protocols bgp group ibgp peer-as 65010
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set policy-options policy-statement bgp_nh_self term 1 from protocol bgp
set policy-options policy-statement bgp_nh_self term 1 then next-hop self
set policy-options policy-statement bgp_nh_self term 1 then accept
set protocols bgp group ibgp export bgp_nh_self
set protocols bgp group ibgp allow 10.10.10.0/24
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.1.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway hub-to-spoke-gw ike-policy ike-policy1
set security ike gateway hub-to-spoke-gw dynamic distinguished-name wildcard OU=SLT
set security ike gateway hub-to-spoke-gw dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw local-identity distinguished-name
set security ike gateway hub-to-spoke-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn ike gateway hub-to-spoke-gw
set security ipsec vpn hub-to-spoke-vpn ike ipsec-policy vpn-policy1
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
```

```
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface ge-0/0/3.0
user@host# set policy-statement lan_nw then accept
user@host# set policy-statement bgp_nh_self term 1 from protocol bgp
user@host# set policy-statement bgp_nh_self term 1 then next-hop self
user@host# set policy-statement bgp_nh_self term 1 then accept
[edit protocols bgp]
user@host# set group ibgp type internal
user@host# set group ibgp local-address 10.10.10.1
user@host# set group ibgp export lan_nw
user@host# set group ibgp cluster 10.2.3.4
user@host# set group ibgp peer-as 65010
user@host# set group ibgp allow 10.10.10.0/24
user@host# set group ibgp export bgp_nh_self
[edit routing-options]
user@host# set static route 10.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 10.3.3.0/30 next-hop 10.1.1.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

   If you intend to use preshared keys instead of certificates for the authentication, make the following changes in your configuration:

   - In the ike proposal, at the [`edit security ike proposal ike-proposal`] hierarchy level, replace `authentication-method rsa-signatures` with the `authentication-method pre-shared-keys`.

     For details about the options, see proposal (Security IKE).

   - In the ike policy, at the [`edit security ike policy` *`policy-name`*] hierarchy level, replace `certificate local-certificate Local1` with the `pre-shared-key ascii-text` *`key`*.

     - For example, `set pre-shared-key ascii-text juniper123`

     For details about the options, see policy (Security IKE).

   - In the ike gateway, at the [`edit security ike gateway hub-to-spoke-gw`] hierarchy level,

     - Replace `dynamic distinguished-name wildcard OU=SLT` with the `dynamic hostname` *`domain-name`*.

       - For example, `set dynamic hostname juniper.net`

         Ensure your device is able to resolve the hostname. Alternatively, you can use `set dynamic general-ikeid` and `set dynamic ike-user-type group-ike-id` for the spoke dynamic identity.

     - Replace `local-identity distinguished-name` with the `local-identity hostname` *`hub-hostname`*.

       - For example, `set local-identity hostname hub.juniper.net`.

         Ensure your device is able to resolve the hostname. Alternatively, you can use `inet` *`ip-address`* as in `set local-identity inet 192.168.1.100`.

     For details about the options, see gateway (Security IKE).

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway hub-to-spoke-gw]
user@host# set ike-policy ike-policy1
user@host# set dynamic distinguished-name wildcard OU=SLT
```

```
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn hub-to-spoke-vpn]
user@host# set bind-interface st0.0
user@host# set ike gateway hub-to-spoke-gw
user@host# set ike ipsec-policy vpn-policy1
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile. Ignore this step, if you are using PSK.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
```

```
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement bgp_nh_self {
    term 1 {
        from protocol bgp;
        then {
            next-hop self;
```

```
            accept;
        }
    }
}
policy-statement lan_nw {
    from interface ge-0/0/3.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp {
        type internal;
        local-address 10.10.10.1;
        export lan_nw;
        cluster 10.2.3.4;
        peer-as 65010;
        allow 10.10.10.0/24;
        export bgp_nh_self;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.1.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
```

```
    gateway hub-to-spoke-gw {
        ike-policy ike-policy1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn {
        bind-interface st0.0;
        ike {
            gateway hub-to-spoke-gw;
            ipsec-policy vpn-policy1;
        }
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
```

```
        st0.0;
        ge-0/0/1.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.10.10.2
set protocols bgp group ibgp export lan_nw
set protocols bgp group ibgp neighbor 10.10.10.1
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface fe-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
```

```
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.2/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface fe-0/0/4.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp type internal
user@host# set group ibgp local-address 10.10.10.2
user@host# set group ibgp export lan_nw
user@host# set group ibgp neighbor 10.10.10.1
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

If you intend to use preshared keys instead of certificates for the authentication, make the following changes in your configuration.

- In the ike proposal, at the [`edit security ike proposal ike-proposal`] hierarchy level, replace `authentication-method rsa-signatures` with the `authentication-method pre-shared-keys`.
- In the ike policy, at the [`edit security ike policy` *`policy-name`*] hierarchy level, replace `certificate local-certificate Local1` with the `pre-shared-key ascii-text` *`key`*.
- In the ike gateway, at the [`edit security ike gateway hub-to-spoke-gw`] hierarchy level,
  - Replace `local-identity distinguished-name` with the `local-identity hostname` *`spoke1-hostname`*.
    - For example, `set local-identity hostname spoke1.juniper.net`.
  - Replace `remote-identity distinguished-name` with the `remote-identity hostname` *`hub-hostname`*.
    - For example, `set remote-identity hostname hub.juniper.net`

  Ensure your device is able to resolve the hostname. Alternatively, you can use `inet` *`ip-address`* as in `set local-identity inet 172.16.1.100` and `set remote-identity inet 192.168.1.100`.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
```

```
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile. Ignore this step, if you are using PSK.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security`

policies, and show security pki commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.2/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface fe-0/0/4.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp {
        type internal;
        local-address 10.10.10.2;
        export lan_nw;
        neighbor 10.10.10.1;
    }
```

```
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
```

```
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
```

```
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.70.70.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.3/24
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.10.10.3
set protocols bgp group ibgp export lan_nw
set protocols bgp group ibgp neighbor 10.10.10.1
set routing-options static route 10.1.1.0/30 next-hop 10.3.3.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
```

```
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.70.70.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.3/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface fe-0/0/4.0
```

```
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp type internal
user@host# set group ibgp local-address 10.10.10.3
user@host# set group ibgp export lan_nw
user@host# set group ibgp neighbor 10.10.10.1
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.3.3.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

   If you intend to use preshared keys instead of certificates for the authentication, make the following changes in your configuration.

   - In the ike proposal, at the [`edit security ike proposal ike-proposal`] hierarchy level, replace `authentication-method rsa-signatures` with the `authentication-method pre-shared-keys`.
   - In the ike policy, at the [`edit security ike policy` *`policy-name`*] hierarchy level, replace `certificate local-certificate Local1` with the `pre-shared-key ascii-text` *`key`*.
   - In the ike gateway, at the [`edit security ike gateway hub-to-spoke-gw`] hierarchy level,
     - Replace `local-identity distinguished-name` with the `local-identity hostname` *`spoke2-hostname`*.
       - For example, `set local-identity hostname spoke2.juniper.net`
     - Replace `remote-identity distinguished-name` with the `remote-identity hostname` *`hub-hostname`*.
       - For example, `set remote-identity hostname hub.juniper.net`

     Ensure your device is able to resolve the hostname. Alternatively, you can use `inet` *`ip-address`* as in `set local-identity inet 10.0.1.100` and `set remote-identity inet 192.168.1.100`.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
```

```
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile. Ignore this step, if you are using PSK.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.3.3.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.70.70.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.3/24;
            }
        }
    }
[edit]
user@host# show policy-options
```

```
policy-statement lan_nw {
    from interface fe-0/0/4.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp {
        type internal;
        local-address 10.10.10.3;
        export lan_nw;
        neighbor 10.10.10.1;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.3.3.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
```

```
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
```

```
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Phase 1 Status | **963**
- Verifying IPsec Phase 2 Status | **963**
- Verifying IPsec Next-Hop Tunnels | **964**
- Verifying BGP | **964**
- Verifying Learned Routes | **965**

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5480163 UP     a558717f387074ab  6d0135c5ecaed61d  Main           10.3.3.1
5480162 UP     7a63d16a5a723df1  c471f7ae166d3a34  Main           10.2.2.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173400 ESP:des/ md5 9bf33bc7 3567/ unlim -   root 500   10.2.2.1
  >268173400 ESP:des/ md5 aae5196b 3567/ unlim -   root 500   10.2.2.1
  <268173401 ESP:des/ md5 69c24d81 622/ unlim  -   root 500   10.3.3.1
  >268173401 ESP:des/ md5 e3fe0231 622/ unlim  -   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                  Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn               Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
10.10.10.3        st0.0       hub-to-spoke-vpn               Auto     C=IN, DC=example.net,
ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.10.10.2               10        116        119       0       0       50:25
1/1/1/0              0/0/0/0
10.10.10.3               10        114        114       0       0       50:04
1/1/1/0              0/0/0/0
```

Verifying Learned Routes

Purpose

Verify that routes to the spokes have been learned.

Action

From operational mode, enter the **show route 10.60.60.0** command.

```
user@host> show route 10.60.60.0
inet.0: 45 destinations, 45 routes (44 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.60.60.0/24      *[BGP/170] 00:50:57, localpref 100
                      AS path: I
                    > to 10.10.10.2 via st0.0
```

From operational mode, enter the **show route 10.70.70.0** command.

```
user@host> show route 10.70.70.0
inet.0: 45 destinations, 45 routes (44 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.70.70.0/24      *[BGP/170] 00:50:42, localpref 100
                      AS path: I
                    > to 10.10.10.3 via st0.0
```

SEE ALSO

Routing Protocols Overview

Example: Configuring Basic AutoVPN with iBGP for IPv6 Traffic

IN THIS SECTION

- Requirements | **966**
- Overview | **967**
- Configuration | **970**
- Verification | **1001**

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures AutoVPN for IPv6 environment using iBGP to forward packets through the VPN tunnels using the certificate based authentication. For authentication with preshared key, set up a similar configuration shown at "Example: Configuring Basic AutoVPN with iBGP" on page 929.

Requirements

This example uses the following hardware and software components:

- Three supported firewalls as AutoVPN hub and spokes.
- Supported Junos OS Release.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels. For more information about specific requirements for a dynamic routing protocol, see the Routing Protocols Overview.

Overview

IN THIS SECTION

- Topology | 969

This example shows the configuration of an AutoVPN hub and the subsequent configurations of two spokes .

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 103 on page 967 shows the options used in this example.

Table 103: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	19
Authentication algorithm	SHA-384
Encryption algorithm	AES 256 CBC

Table 103: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations *(Continued)*

Option	Value
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Lifetime Seconds	3000
Encryption algorithm	AES 256 GCM
IPsec policy:	
Perfect Forward Secrecy (PFS) group	19

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 104 on page 968 shows the options configured on the hub and on all spokes.

Table 104: AutoVPN Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	2001:db8:2000::1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string SLT in the organizational unit (OU) field	DN on the hub's certificate

Table 104: AutoVPN Configuration for Hub and All Spokes *(Continued)*

Option	Hub	All Spokes
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/0	Spoke 1: ge-0/0/0.0 Spoke 2: ge-0/0/0.0
VPN:		
Bind interface	st0.1	st0.1
Establish tunnels	(not configured)	establish-tunnels on-traffic

Table 105 on page 969 shows the configuration options that are different on each spoke.

Table 105: Comparison Between the Spoke Configurations

Option	Spoke 1	Spoke 2
st0.0 interface	2001:db8:7000::2/64	2001:db8:7000::3/64
Interface to internal network	(ge-0/0/1.0) 2001:db8:4000::1/64	(ge-0/0/1.0) 2001:db8:6000::1/64
Interface to Internet	(ge-0/0/0.0) 2001:db8:3000::2/64	(ge-0/0/0.0) 2001:db8:5000::2/64

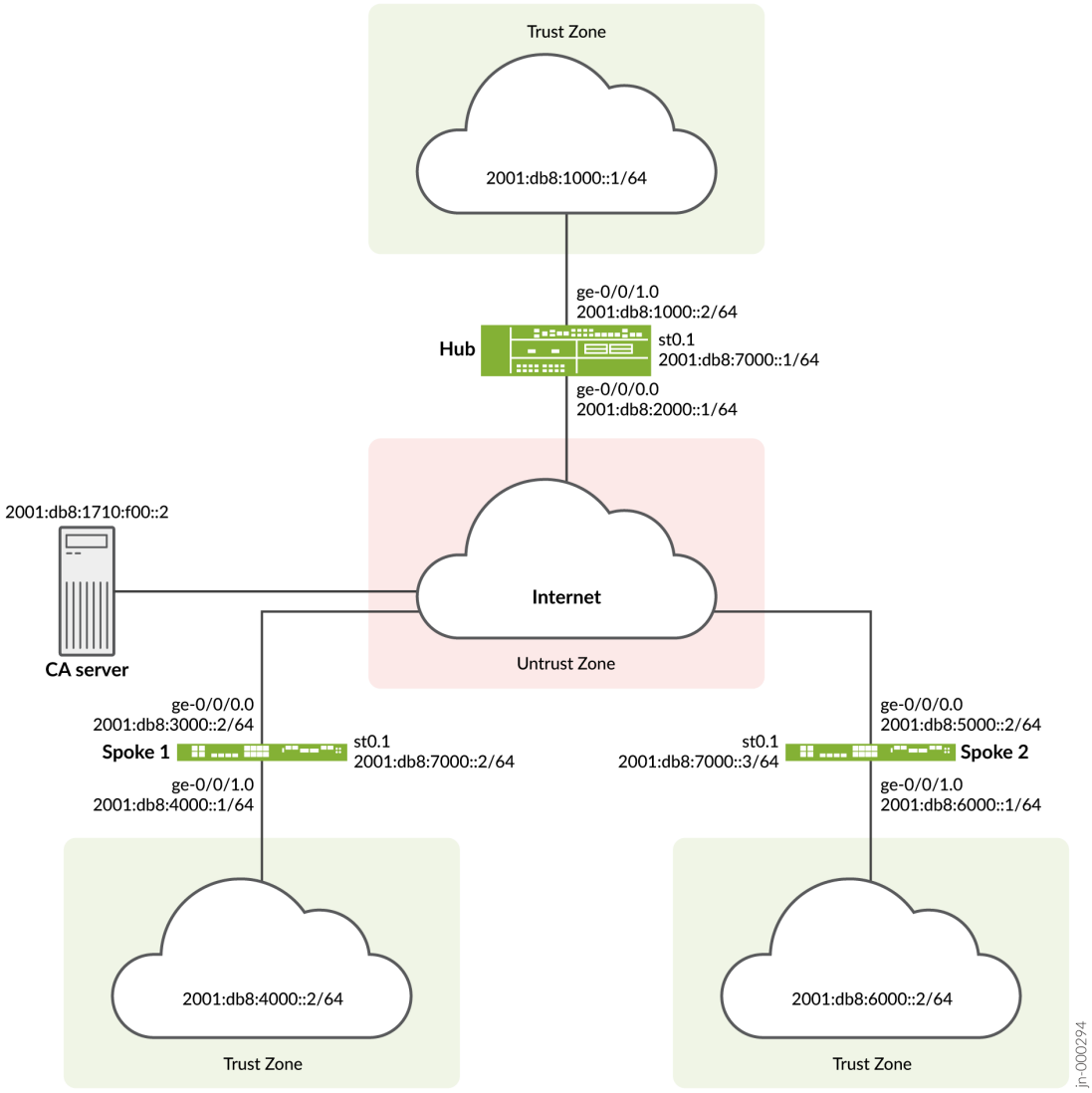
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments.

Topology

Figure 58 on page 970 shows the firewalls to be configured for AutoVPN in this example.

Figure 58: Basic AutoVPN Deployment with iBGP



Configuration

IN THIS SECTION

- Enroll Device Certificates with SCEP | **971**
- Configuring the Hub | **976**
- Configuring Spoke 1 | **985**

- Configuring Spoke 2 | 993

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
```

```
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
```

```
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
```

```
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Tumkur, Common name: spoke2, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
  Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 6-2012 10:02
    Not after: 11- 6-2013 10:12
```

```
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
    27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
    77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
    44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
    7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
    7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
    58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
    00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
```

```
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate HUB
set security ike gateway IKE_GWA_1 ike-policy IKE_POL
set security ike gateway IKE_GWA_1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway IKE_GWA_1 dead-peer-detection always-send
set security ike gateway IKE_GWA_1 dead-peer-detection interval 10
set security ike gateway IKE_GWA_1 dead-peer-detection threshold 3
set security ike gateway IKE_GWA_1 local-identity distinguished-name
set security ike gateway IKE_GWA_1 external-interface ge-0/0/0
set security ike gateway IKE_GWA_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPNA_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPNA_1 ike gateway IKE_GWA_1
set security ipsec vpn IPSEC_VPNA_1 ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::1/64
set routing-options rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::2
set routing-options rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::2
set routing-options autonomous-system 100
set routing-options forwarding-table export load_balance
set protocols bgp traceoptions file bgp
set protocols bgp traceoptions flag all
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 2001:db8:9000::1
set protocols bgp group ibgp export ibgp
set protocols bgp group ibgp cluster 10.1.3.4
set protocols bgp group ibgp peer-as 100
```

```
set protocols bgp group ibgp multipath
set protocols bgp group ibgp allow 2001:db8:9000::/64
set policy-options policy-statement ibgp from interface ge-0/0/1.0
set policy-options policy-statement ibgp then accept
set policy-options policy-statement load_balance then load-balance per-packet
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::1/64
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement ibgp from interface ge-0/0/1.0
user@host# set policy-statement ibgp then accept
user@host# set policy-statement load_balance then load-balance per-packet
[edit protocols bgp]
user@host# set traceoptions file bgp
user@host# set traceoptions flag all
user@host# set group ibgp type internal
user@host# set group ibgp local-address 2001:db8:9000::1
user@host# set group ibgp export ibgp
user@host# set group ibgp cluster 10.1.3.4
user@host# set group ibgp peer-as 100
user@host# set group ibgp multipath
user@host# set group ibgp allow 2001:db8:9000::/64
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::2
user@host# set rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::2
```

```
user@host# set autonomous-system 100
user@host# set forwarding-table export load_balance
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal ike-proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate HUB
[edit security ike gateway IKE_GWA_1]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_1]
user@host# set bind-interface st0.1
```

```
user@host# set ike gateway IKE_GWA_1
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security`

policies, and show security pki commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:2000::1/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:1000::2/64;
            }
        }
    }
    st0 {
        unit 1{
            multipoint;
            family inet6 {
                address 2001:db8:7000::1/64;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement ibgp {
    from interface ge-0/0/1.0;
    then accept;
}
policy-statement load_balance {
    then {
        load-balance per-packet;
    }
}
[edit]
user@host# show protocols
bgp {
    traceoptions {
```

```
        file bgp;
        flag all;
    }
    group ibgp {
        type internal;
        local-address 2001:db8:9000::1;
        export ibgp;
        cluster 10.1.3.4;
        peer-as 100;
        multipath;
        allow 2001:db8:9000::/64;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route route 2001:db8:3000::/64 next-hop 2001:db8:2000::2;
        route 2001:db8:5000::/64 next-hop 2001:db8:2000::2;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate HUB;
    }
}
gateway IKE_GWA_1 {
    ike-policy IKE_POL;
```

```
    dynamic {
        distinguished-name {
            wildcard OU=SLT;
        }
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface ge-0/0/0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_1;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
```

```
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE1
set security ike gateway IKE_GW_SPOKE_1 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_1 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_1 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_1 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_1 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike gateway IKE_GW_SPOKE_1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_1 establish-tunnels on-traffic
```

```
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::2/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::1
set routing-options autonomous-system 100
set protocols bgp traceoptions file bgp
set protocols bgp traceoptions flag all
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 2001:db8:9000::2
set protocols bgp group ibgp export ibgp
set protocols bgp group ibgp peer-as 100
set protocols bgp group ibgp neighbor 2001:db8:9000::1
set policy-options policy-statement ibgp from interface ge-0/0/1.0
set policy-options policy-statement ibgp then accept
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::2/64
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement ibgp from interface ge-0/0/1.0
user@host# set policy-statement ibgp then accept
```

```
[edit protocols bgp]
user@host# set traceoptions file bgp
user@host# set traceoptions flag all
user@host# set group ibgp type internal
user@host# set group ibgp local-address 2001:db8:9000::2
user@host# set group ibgp export ibgp
user@host# set group ibgp peer-as 100
user@host# set group ibgp neighbor 2001:db8:9000::1
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::1
user@host# set autonomous-system 100
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal ike-proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE1
[edit security ike gateway IKE_GW_SPOKE_1]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_SPOKE_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_SPOKE_1
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels on-traffic
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
```

```
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:3000::2/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:4000::1/64;
            }
        }
    }
    st0 {
        unit 1{
            family inet6 {
                address 2001:db8:7000::2/64;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement ibgp {
    from interface ge-0/0/1.0;
    then accept;
}
[edit]
user@host# show protocols
```

```
bgp {
    traceoptions {
        file bgp;
        flag all;
    }
    group ibgp {
        type internal;
        local-address 2001:db8:9000::2;
        export ibgp;
        peer-as 100;
        neighbor 2001:db8:9000::1;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route route  2001:db8:2000::/64 next-hop 2001:db8:3000::1;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE1;
    }
}
gateway IKE_GWA_SPOKE1 {
    ike-policy IKE_POL;
    dynamic {
```

```
        distinguished-name {
            wildcard OU=SLT;
        }
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface ge-0/0/0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_SPOKE_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_SPOKE_1;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
```

```
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE2
set security ike gateway IKE_GW_SPOKE_2 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_2 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_2 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_2 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_2 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_2 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_2 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike gateway IKE_GW_SPOKE_2
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_2 establish-tunnels on-traffic
```

```
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::3/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
set routing-options autonomous-system 100
set protocols bgp traceoptions file bgp
set protocols bgp traceoptions flag all
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 2001:db8:9000::3
set protocols bgp group ibgp export ibgp
set protocols bgp group ibgp peer-as 100
set protocols bgp group ibgp neighbor 2001:db8:9000::1
set policy-options policy-statement ibgp from interface ge-0/0/1.0
set policy-options policy-statement ibgp then accept
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::3/64
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement ibgp from interface ge-0/0/1.0
user@host# set policy-statement ibgp then accept
```

```
[edit protocols bgp]
user@host# set traceoptions file bgp
user@host# set traceoptions flag all
user@host# set group ibgp type internal
user@host# set group ibgp local-address 2001:db8:9000::3
user@host# set group ibgp export ibgp
user@host# set group ibgp peer-as 100
user@host# set group ibgp neighbor 2001:db8:9000::1
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
user@host# set autonomous-system 100
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal ike-proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE2
[edit security ike gateway IKE_GW_SPOKE_2]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_SPOKE_2]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_SPOKE_2
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels on-traffic
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
```

```
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:5000::2/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:6000::1/64;
            }
        }
    }
    st0 {
        unit 1{
            family inet6 {
                address 2001:db8:7000::3/64;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement ibgp {
    from interface ge-0/0/1.0;
    then accept;
}
[edit]
user@host# show protocols
```

```
bgp {
    traceoptions {
        file bgp;
        flag all;
    }
    group ibgp {
        type internal;
        local-address 2001:db8:9000::3;
        export ibgp;
        peer-as 100;
        neighbor 2001:db8:9000::1;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route route 2001:db8:2000::/64 next-hop 2001:db8:5000::1;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE2;
    }
}
gateway IKE_GWA_SPOKE2 {
    ike-policy IKE_POL;
    dynamic {
```

```
        distinguished-name {
            wildcard OU=SLT;
        }
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface ge-0/0/0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_SPOKE_2 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_SPOKE_2;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
```

```
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Status | **1001**
- Verifying IPsec Status | **1002**
- Verifying IPsec Next-Hop Tunnels | **1002**
- Verifying BGP | **1003**

Confirm that the configuration is working properly.

Verifying IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the **show security ike sa** command.

```
user@host> show security ike sa
Index  State Initiator cookie          Responder cookie          Mode Remote Address

493333 UP    2001:db8:88b49d915e684c93 2001:db8:fe890b1cac8522b5 Main 2001:db8:3000::2

493334 UP    2001:db8:26e40244ad3d722d 2001:db8:68b4d9f94097d32e Main 2001:db8:5000::2
```

Meaning

The `show security ike sa` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec sa** command.

```
user@host> show security ipsec sa
Total active tunnels: 2
  ID        Algorithm      SPI  Life:sec/kb    Mon    lsys Port Gateway
  >67108885 ESP:aes-gcm-256/None fdef4dab 2918/ unlim - root 500  2001:db8:3000::2
  >67108885 ESP:aes-gcm-256/None e785dadc 2918/ unlim - root 500  2001:db8:3000::2
  >67108887 ESP:aes-gcm-256/None 34a787af 2971/ unlim - root 500  2001:db8:5000::2
  >67108887 ESP:aes-gcm-256/None cf57007f 2971/ unlim - root 500  2001:db8:5000::2
```

Meaning

The `show security ipsec sa` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway             interface  IPSec VPN name  Flag  IKE-
ID                               XAUTH username

2001:db8:9000::2             st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available
```

```
2001:db8:9000::3              st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available

2001:db8::5668:ad10:fcd8:163c st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available

2001:db8::5668:ad10:fcd8:18a1 st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table       Tot Paths  Act Paths  Suppressed History Damp State    Pending
inet6.0
            2          2          0          0           0        0
Peer                AS   InPkt     OutPkt  OutQ  Flaps Last Up/Dwn State
2001:db8:9000::2    100  4          4      0     0          32     Establ
  inet6.0: 1/1/1/0
2001:db8:9000::3    100  4          4      0     0          8      Establ
  inet6.0: 1/1/1/0
```

SEE ALSO

Routing Protocols Overview

Example: Configuring AutoVPN with iBGP and ECMP

IN THIS SECTION

- Requirements | 1004
- Overview | 1005
- Configuration | 1008
- Verification | 1033

This example shows how to configure two IPsec VPN tunnels between an AutoVPN hub and spoke. This example configures iBGP with equal-cost multipath (ECMP) to forward packets through the VPN tunnels using the certificate based authentication. For authentication with preshared key, set up a similar configuration shown at "Example: Configuring Basic AutoVPN with iBGP" on page 929.

Requirements

This example uses the following hardware and software components:

- Two supported firewalls as AutoVPN hub and spoke
- Supported Junos OS Release

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

- Topology | **1007**

This example shows the configuration of an AutoVPN hub and a spoke with two IPsec VPN tunnels.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). Certificates are enrolled in the hub and in the spoke for each IPsec VPN tunnel. One of the certificates for the spoke contains the organizational unit (OU) value “SLT” in the distinguished name (DN); the hub is configured with a group IKE ID to match the value “SLT” in the OU field. The other certificate for the spoke contains the OU value “SBU” in the DN; the hub is configured with a group IKE ID to match the value “SBU” in the OU field.

The spoke establishes IPsec VPN connections to the hub, which allows it to access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and the spoke must have the same values. Table 106 on page 1005 shows the options used in this example.

Table 106: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke iBGP ECMP Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main

Table 106: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke iBGP ECMP Configurations *(Continued)*

Option	Value
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 107 on page 1006 shows the options configured on the hub and on the spoke.

Table 107: AutoVPN iBGP ECMP Configuration for Hub and Spoke 1

Option	Hub	Spoke 1
IKE gateway:		
Remote IP address	hub-to-spoke-gw-1: Dynamic hub-to-spoke-gw-2: Dynamic	spoke-to-hub-gw-1: 10.1.1.1 spoke-to-hub-gw-2: 10.1.2.1
Remote IKE ID	hub-to-spoke-gw-1: DN on the spoke's certificate with the string SLT in the OU field hub-to-spoke-gw-2: DN on the spoke's certificate with the string SBU in the OU field	spoke-to-hub-gw-1: DN on the hub's certificate spoke-to-hub-gw-2: DN on the hub's certificate

Table 107: AutoVPN iBGP ECMP Configuration for Hub and Spoke 1 *(Continued)*

Option	Hub	Spoke 1
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	hub-to-spoke-gw-1: ge-0/0/1.0 hub-to-spoke-gw-2: ge-0/0/2.0	spoke-to-hub-gw-1: fe-0/0/1.0 spoke-to-hub-gw-2: fe-0/0/2.0
VPN:		
Bind interface	hub-to-spoke-vpn-1: st0.0 hub-to-spoke-vpn-2: st0.1	spoke-to-hub-1: st0.0 spoke-to-hub-2: st0.1
Establish tunnels	(not configured)	Immediately on configuration commit

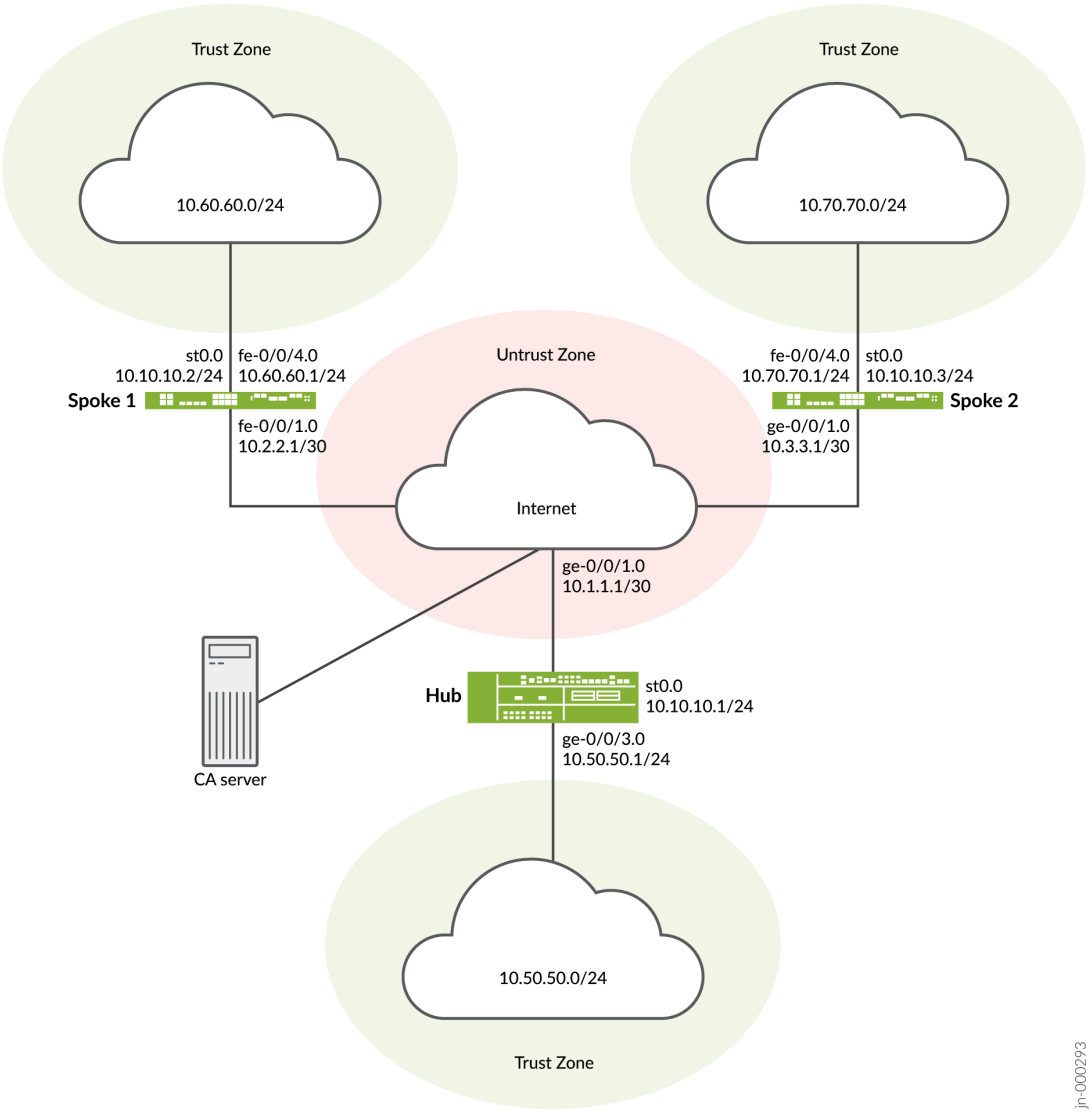
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments.

Topology

Figure 59 on page 1008 shows the firewalls to be configured for AutoVPN in this example.

Figure 59: AutoVPN Deployment with iBGP and ECMP



Configuration

IN THIS SECTION

- Enroll Device Certificates with SCEP | **1009**
- Configuring the Hub | **1014**
- Configuring Spoke 1 | **1024**

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> request security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email hub_backup@example.net ip-address 10.1.2.1 subject
```

```
DC=example.net,CN=hub_backup,OU=SBU,O=example,L=Bengaluru,ST=KA,C=IN challenge-password
<password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
```

```
    Status: Disabled
    Next trigger time: Timer not started
```

```
user@host> show security pki local-certificate certificate-id Local2 detail

Certificate identifier: Local2
  Certificate version: 3
  Serial number: 505efdf900000000259a
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SBU, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub_backup, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SBU, CN=hub_backup
  Alternate subject: "hub_backup@example.net", example.net, 10.1.2.1
  Validity:
    Not before: 11- 9-2012 10:55
    Not after: 11- 9-2013 11:05
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d5:44:08:96:f6:77:05:e6:91:50:8a:8a:2a
    4e:95:43:1e:88:ea:43:7c:c5:ac:88:d7:a0:8d:b5:d9:3f:41:db:db
    44:34:1f:56:a5:38:4b:b2:c5:85:f9:f1:bf:b2:7b:d4:b2:af:98:a0
    95:50:02:ad:f5:dd:4d:dc:67:85:dd:84:09:df:9c:68:a5:58:65:e7
    2c:72:cc:47:4b:d0:cc:4a:28:ca:09:db:ad:6e:5a:13:6c:e6:cc:f0
    29:ed:2b:2d:d1:38:38:bc:68:84:de:ae:86:39:c9:dd:06:d5:36:f0
    e6:2a:7b:46:4c:cd:a5:24:1c:e0:92:8d:ad:35:29:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    98:96:2f:ff:ca:af:33:ee:d7:4c:c8:4f:f7:71:53:c0:5d:5f:c5:59 (sha1)
    c9:87:e3:a4:5c:47:b5:aa:90:22:e3:06:b2:0b:e1:ea (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> request security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email spoke1_backup@example.net ip-address  10.3.3.1
subject DC=example.net,CN=spoke1_backup,OU=SBU,O=example,L=Mysore,ST=KA,C=IN challenge-
password <password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
```

```
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started

user@host> show security pki local-certificate certificate-id Local2 detail

Certificate identifier: Local2
  Certificate version: 3
  Serial number: 506c3d0600000000259b
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SBU, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1_backup, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
  Alternate subject: "spoke1_backup@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 9-2012 11:09
```

```
    Not after: 11- 9-2013 11:19
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:a7:02:b5:e2:cd:79:24:f8:97:a3:8d:4d:27
    8c:2b:dd:f1:57:72:4d:2b:6d:d5:95:0d:9c:1b:5c:e2:a4:b0:84:2e
    31:82:3c:91:08:a2:58:b9:30:4c:5f:a3:6b:e6:2b:9c:b1:42:dd:1c
    cd:a2:7a:84:ea:7b:a6:b7:9a:13:33:c6:27:2b:79:2a:b1:0c:fe:08
    4c:a7:35:fc:da:4f:df:1f:cf:f4:ba:bc:5a:05:06:63:92:41:b4:f2
    54:00:3f:ef:ff:41:e6:ca:74:10:56:f7:2b:5f:d3:1a:33:7e:49:74
    1c:42:cf:c2:23:ea:4b:8f:50:2c:eb:1c:a6:37:89:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    d6:7f:52:a3:b6:f8:ae:cb:70:3f:a9:79:ea:8a:da:9e:ba:83:e4:5f (sha1)
    76:0b:72:73:cf:51:ee:58:81:2d:f7:b4:e2:5c:f4:5c (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT` for Local1 and `SBU` for Local2. The IKE configurations on the hub include `OU=SLT` and `OU=SBU` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/2 unit 0 family inet address 10.1.2.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 10.20.20.1/24
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set policy-options policy-statement load_balance then load-balance per-packet
set protocols bgp group ibgp-1 type internal
```

```
set protocols bgp group ibgp-1 local-address 10.10.10.1
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 cluster 10.2.3.4
set protocols bgp group ibgp-1 multipath
set protocols bgp group ibgp-1 allow 10.10.10.0/24
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.1
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 cluster 10.2.3.5
set protocols bgp group ibgp-2 multipath
set protocols bgp group ibgp-2 allow 10.20.20.0/24
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.2.2
set routing-options autonomous-system 65010
set routing-options forwarding-table export load_balance
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway hub-to-spoke-gw-1 ike-policy ike-policy-1
set security ike gateway hub-to-spoke-gw-1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway hub-to-spoke-gw-1 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-1 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-1 external-interface ge-0/0/1.0
set security ike gateway hub-to-spoke-gw-2 ike-policy ike-policy-2
set security ike gateway hub-to-spoke-gw-2 dynamic distinguished-name wildcard OU=SBU
set security ike gateway hub-to-spoke-gw-2 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-2 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-2 external-interface ge-0/0/2.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn-1 bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn-1 ike gateway hub-to-spoke-gw-1
set security ipsec vpn hub-to-spoke-vpn-1 ike ipsec-policy vpn-policy
```

```
set security ipsec vpn hub-to-spoke-vpn-2 bind-interface st0.1
set security ipsec vpn hub-to-spoke-vpn-2 ike gateway hub-to-spoke-gw-2
set security ipsec vpn hub-to-spoke-vpn-2 ike ipsec-policy vpn-policy
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
user@host# set ge-0/0/2 unit 0 family inet address 10.1.2.1/30
user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 10.20.20.1/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface ge-0/0/3.0
user@host# set policy-statement lan_nw then accept
user@host# set policy-statement load_balance then load-balance per-packet
[edit protocols bgp]
```

```
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.1
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 cluster 10.2.3.4
user@host# set group ibgp-1 multipath
user@host# set group ibgp-1 allow 10.10.10.0/24
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.1
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 cluster 10.2.3.5
user@host# set group ibgp-2 multipath
user@host# set group ibgp-2 allow 10.20.20.0/24
[edit routing-options]
user@host# set static route 10.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 10.3.3.0/30 next-hop 10.1.2.2
user@host# set autonomous-system 65010
user@host# set forwarding-table export load_balance
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway hub-to-spoke-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
[edit security ike gateway hub-to-spoke-gw-2]
user@host# set ike-policy ike-policy-2
user@host# set dynamic distinguished-name wildcard OU=SBU
```

```
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn hub-to-spoke-vpn-1]
user@host# set bind-interface st0.0
user@host# set ike gateway hub-to-spoke-gw-1
user@host# set ike ipsec-policy vpn-policy
[edit security ipsec vpn hub-to-spoke-vpn-2]
user@host# set bind-interface st0.1
user@host# set ike gateway hub-to-spoke-gw-2
user@host# set ike ipsec-policy vpn-policy
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.0
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces ge-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 10.1.2.1/30;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
```

```
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
        unit 1 {
            multipoint;
            family inet {
                address 10.20.20.1/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface ge-0/0/3.0;
    then accept;
}
    policy-statement load_balance {
        then {
            load-balance per-packet;
        }
    }
[edit]
user@host# show protocols
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.1;
        export lan_nw;
        cluster 10.2.3.4;
        multipath;
        allow 10.10.10.0/24;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.1;
        export lan_nw;
```

```
        cluster 10.2.3.5;
        multipath;
        allow 10.20.20.0/24;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.2.2;
    }
autonomous-system 65010;
    forwarding-table {
        export load_balance;
    }
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
        }
    }
    gateway hub-to-spoke-gw-1 {
        ike-policy ike-policy-1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
```

```
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
    gateway hub-to-spoke-gw-2 {
        ike-policy ike-policy-2;
        dynamic {
            distinguished-name {
                wildcard OU=SBU;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/2.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn-1 {
        bind-interface st0.0;
        ike {
            gateway hub-to-spoke-gw-1;
            ipsec-policy vpn-policy;
        }
    }
    vpn hub-to-spoke-vpn-2 {
        bind-interface st0.1;
        ike {
            gateway hub-to-spoke-gw-2;
            ipsec-policy vpn-policy;
        }
    }
```

```
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        ge-0/0/1.0;
        ge-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
```

```
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/2 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set interfaces st0 unit 1 family inet address 10.20.20.2/24
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp-1 type internal
set protocols bgp group ibgp-1 local-address 10.10.10.2
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 neighbor 10.10.10.1
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.2
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 neighbor 10.20.20.1
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set routing-options static route 10.1.2.0/30 next-hop 10.3.3.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
```

```
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway spoke-to-hub-gw-1 ike-policy ike-policy-1
set security ike gateway spoke-to-hub-gw-1 address 10.1.1.1
set security ike gateway spoke-to-hub-gw-1 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 external-interface fe-0/0/1.0
set security ike gateway spoke-to-hub-gw-2 ike-policy ike-policy-2
set security ike gateway spoke-to-hub-gw-2 address 10.1.2.1
set security ike gateway spoke-to-hub-gw-2 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 external-interface fe-0/0/2.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn spoke-to-hub-1 bind-interface st0.0
set security ipsec vpn spoke-to-hub-1 ike gateway spoke-to-hub-gw-1
set security ipsec vpn spoke-to-hub-1 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-1 establish-tunnels immediately
set security ipsec vpn spoke-to-hub-2 bind-interface st0.1
set security ipsec vpn spoke-to-hub-2 ike gateway spoke-to-hub-gw-2
set security ipsec vpn spoke-to-hub-2 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-2 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces fe-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
user@host# set fe-0/0/2 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
user@host# set st0 unit 0 family inet address 10.10.10.2/24
user@host# set st0 unit 1 family inet address 10.20.20.2/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface fe-0/0/4.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.2
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 neighbor 10.10.10.1
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.2
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 neighbor 10.20.20.1
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
user@host# set static route 10.1.2.0/30 next-hop 10.3.3.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
user@host# set mode main
user@host# set proposals ike-proposal
```

```
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway spoke-to-hub-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
[edit security ike gateway spoke-to-hub-gw-2]
user@host# set ike-policy ike-policy-2
user@host# set address 10.1.2.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub-1]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw-1
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
[edit security ipsec vpn spoke-to-hub-2]
user@host# set bind-interface st0.1
user@host# set ike gateway spoke-to-hub-gw-2
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
user@host# set interfaces fe-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
```

```
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.3.1/30;
            }
        }
    }
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            family inet {
                address 10.10.10.2/24;
            }
        }
        unit 1 {
            family inet {
                address 10.20.20.2/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface fe-0/0/4.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.2;
        export lan_nw;
```

```
        neighbor 10.10.10.1;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.2;
        export lan_nw;
        neighbor 10.20.20.1;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
    route 10.1.2.0/30 next-hop 10.3.3.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
        }
    }
    gateway spoke-to-hub-gw-1 {
        ike-policy ike-policy-1;
        address 1o.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
```

```
        external-interface fe-0/0/1.0;
    }
    gateway spoke-to-hub-gw-2 {
        ike-policy ike-policy-2;
        address 1o.1.2.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/2.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub-1 {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw-1;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
    }
    vpn spoke-to-hub-2 {
        bind-interface st0.1;
        ike {
            gateway spoke-to-hub-gw-2;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
```

```
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
        fe-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Phase 1 Status | **1033**
- Verifying IPsec Phase 2 Status | **1034**
- Verifying IPsec Next-Hop Tunnels | **1034**
- Verifying BGP | **1035**
- Verifying Learned Routes | **1035**
- Verifying Route Installation in Forwarding Table | **1037**

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
3733049 UP     bc9686796c2e52e9  1fbe46eee168f24e  Main           10.2.2.1
3733048 UP     a88db7ed23ec5f6b  c88b81dff52617a5  Main           10.3.3.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173315 ESP:des/ md5 93cfb417 1152/ unlim -   root 500   10.2.2.1
  >268173315 ESP:des/ md5 101de6f7 1152/ unlim -   root 500   10.2.2.1
  <268173313 ESP:des/ md5 272e29c0 1320/ unlim -   root 500   10.3.3.1
  >268173313 ESP:des/ md5 a3bf8fad 1320/ unlim -   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                   Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn-1              Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
```

```
10.20.20.2        st0.1       hub-to-spoke-vpn-2               Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spoke.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.10.10.2            65010       4819       4820       0       2 1d 12:15:14
1/1/1/0              0/0/0/0
10.20.20.2            65010       4926       4928       0       0 1d 13:03:03
1/1/1/0              0/0/0/0
```

Verifying Learned Routes

Purpose

Verify that routes to the spoke have been learned.

Action

From operational mode, enter the **show route 10.60.60.0 detail** command.

```
user@host> show route 10.60.60.0 detail
inet.0: 47 destinations, 48 routes (46 active, 0 holddown, 1 hidden)
10.60.60.0/24 (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0x167407c
                Next-hop reference count: 3
                Source: 10.10.10.2
                Next hop type: Router
                Next hop: 10.10.10.2 via st0.0
                Next hop type: Router
                Next hop: 10.20.20.2 via st0.1, selected
                Protocol next hop: 10.10.10.2
                Indirect next hop: 15c8000 262142
                Protocol next hop: 10.20.20.2
                Indirect next hop: 15c80e8 262143
                State: <Act Int Ext>
                Local AS:    65010 Peer AS:    65010
                Age: 1d 12:16:25     Metric2: 0
                Task: BGP_10.10.10.10.2+53120
                Announcement bits (2): 0-KRT 3-Resolve tree 1
                AS path: I
                Accepted Multipath
                Localpref: 100
                Router ID: 10.207.36.182
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0x15b8ac0
                Next-hop reference count: 1
                Source: 10.20.20.2
                Next hop type: Router
                Next hop: 10.20.20.2 via st0.1, selected
                Protocol next hop: 10.20.20.2
                Indirect next hop: 15c80e8 262143
                State: <NotBest Int Ext>
                Inactive reason: Not Best in its group - Update source
                Local AS:    65010 Peer AS:    65010
                Age: 1d 13:04:14     Metric2: 0
```

```
                Task: BGP_10.20.20.20.2+50733
                AS path: I
                Accepted MultipathContrib
                Localpref: 100
                Router ID: 10.207.36.182
```

Verifying Route Installation in Forwarding Table

Purpose

Verify that routes to the spoke have been installed in the forwarding table.

Action

From operational mode, enter the **show route forwarding-table matching 10.60.60.0** command.

```
user@host> show route forwarding-table matching 60.60.60.0
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
10.60.60.0/24      user     0                    ulst 262144     1
                                                 indr 262142     2
                            10.10.10.2           ucst   572     3 st0.0
                                                 indr 262143     2
                            10.20.20.2           ucst   573     3 st0.1
```

Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels

IN THIS SECTION

- Requirements | 1038
- Overview | 1038
- Configuration | 1042
- Verification | 1067

This example shows how to configure active and backup IPsec VPN tunnels between an AutoVPN hub and spoke. This example configures iBGP to forward traffic through the VPN tunnels using the certificate based authentication. For authentication with preshared key, set up a similar configuration shown at "Example: Configuring Basic AutoVPN with iBGP" on page 929.

Requirements

This example uses the following hardware and software components:

- Two supported firewalls as AutoVPN hub and spoke
- Supported Junos OS Release

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

- Topology | **1041**

This example shows the configuration of an AutoVPN hub and a spoke with two IPsec VPN tunnels.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). Certificates are enrolled in the hub and in the spoke for each IPsec VPN tunnel. One of the certificates for the spoke contains the organizational unit (OU) value “SLT” in the distinguished name (DN); the hub is configured with a group IKE ID to match the value “SLT” in the OU field. The other certificate for the spoke contains the OU value “SBU” in the DN; the hub is configured with a group IKE ID to match the value “SBU” in the OU field.

The spoke establishes IPsec VPN connections to the hub, which allows it to access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and the spoke must have the same values. Table 108 on page 1039 shows the options used in this example.

Table 108: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke iBGP Active-Backup Tunnel Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 109 on page 1040 shows the options configured on the hub and on the spoke.

Table 109: AutoVPN IBGP Active-Backup Tunnel Configuration for Hub and Spoke 1

Option	Hub	Spoke 1
IKE gateway:		
Remote IP address	hub-to-spoke-gw-1: Dynamic hub-to-spoke-gw-2: Dynamic	spoke-to-hub-gw-1: 10.1.1.1 spoke-to-hub-gw-2: 10.1.2.1
Remote IKE ID	hub-to-spoke-gw-1: DN on the spoke's certificate with the string SLT in the OU field hub-to-spoke-gw-2: DN on the spoke's certificate with the string SBU in the OU field	spoke-to-hub-gw-1: DN on the hub's certificate spoke-to-hub-gw-2: DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	hub-to-spoke-gw-1: ge-0/0/1.0 hub-to-spoke-gw-2: ge-0/0/2.0	spoke-to-hub-gw-1: fe-0/0/1.0 spoke-to-hub-gw-2: fe-0/0/2.0
VPN:		
Bind interface	hub-to-spoke-vpn-1: st0.0 hub-to-spoke-vpn-2: st0.1	spoke-to-hub-1: st0.0 spoke-to-hub-2: st0.1
VPN monitor	hub-to-spoke-vpn-1: ge-0/0/1.0 (source interface) hub-to-spoke-vpn-2: ge-0/0/2.0 (source interface)	spoke-to-hub-1: 10.1.1.1 (destination IP) spoke-to-hub-2: 10.1.2.1 (destination IP)
Establish tunnels	(not configured)	Immediately on configuration commit

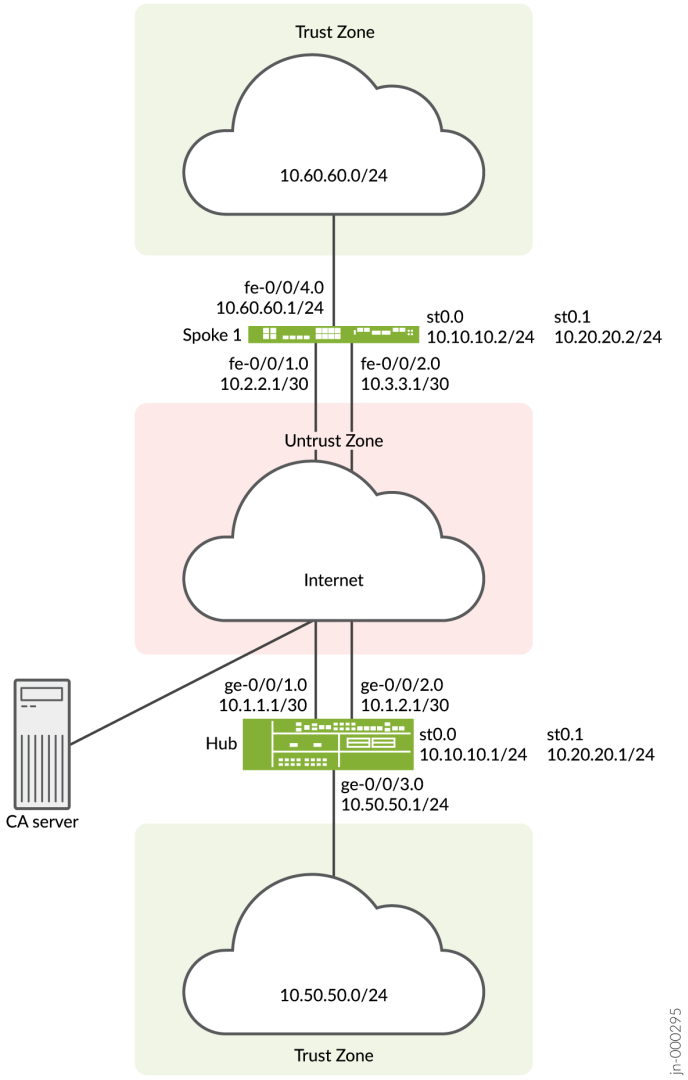
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments.

Topology

Figure 60 on page 1041 shows the firewalls to be configured for AutoVPN in this example.

Figure 60: AutoVPN Deployment with iBGP and Active-Backup Tunnels



In this example, two IPsec VPN tunnels are established between the hub and spoke 1. Routing information is exchanged through iBGP sessions in each tunnel. The longest prefix match for the route to 10.60.60.0/24 is through the st0.0 interface on the hub. Thus, the primary tunnel for the route is through the st0.0 interfaces on the hub and spoke 1. The default route is through the backup tunnel on the st0.1 interfaces on the hub and spoke 1.

VPN monitoring checks the status of the tunnels. If there is a problem with the primary tunnel (for example, the remote tunnel gateway is not reachable), the tunnel status changes to down and data destined for 10.60.60.0/24 is rerouted through the backup tunnel.

Configuration

IN THIS SECTION

- Enroll Device Certificates with SCEP | **1042**
- Configuring the Hub | **1048**
- Configuring Spoke 1 | **1057**

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> request security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email hub_backup@example.net ip-address 10.1.2.1 subject
DC=example.net,CN=hub_backup,OU=SBU,O=example,L=Bengaluru,ST=KA,C=IN challenge-password
<password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
```

```
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

```
user@host> show security pki local-certificate certificate-id Local2 detail

Certificate identifier: Local2
  Certificate version: 3
  Serial number: 505efdf900000000259a
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SBU, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub_backup, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SBU, CN=hub_backup
  Alternate subject: "hub_backup@example.net", example.net, 10.1.2.1
  Validity:
    Not before: 11- 9-2012 10:55
    Not after: 11- 9-2013 11:05
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d5:44:08:96:f6:77:05:e6:91:50:8a:8a:2a
    4e:95:43:1e:88:ea:43:7c:c5:ac:88:d7:a0:8d:b5:d9:3f:41:db:db
    44:34:1f:56:a5:38:4b:b2:c5:85:f9:f1:bf:b2:7b:d4:b2:af:98:a0
    95:50:02:ad:f5:dd:4d:dc:67:85:dd:84:09:df:9c:68:a5:58:65:e7
    2c:72:cc:47:4b:d0:cc:4a:28:ca:09:db:ad:6e:5a:13:6c:e6:cc:f0
    29:ed:2b:2d:d1:38:38:bc:68:84:de:ae:86:39:c9:dd:06:d5:36:f0
    e6:2a:7b:46:4c:cd:a5:24:1c:e0:92:8d:ad:35:29:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
```

```
    98:96:2f:ff:ca:af:33:ee:d7:4c:c8:4f:f7:71:53:c0:5d:5f:c5:59 (sha1)
    c9:87:e3:a4:5c:47:b5:aa:90:22:e3:06:b2:0b:e1:ea (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> request security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email spoke1_backup@example.net ip-address  10.3.3.1
subject DC=example.net,CN=spoke1_backup,OU=SBU,O=example,L=Mysore,ST=KA,C=IN challenge-
password <password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started

user@host> show security pki local-certificate certificate-id Local2 detail

Certificate identifier: Local2
  Certificate version: 3
  Serial number: 506c3d0600000000259b
```

```
Issuer:
  Common name: CASERVER1, Domain component: net, Domain component: internal
Subject:
  Organization: example, Organizational unit: SBU, Country: IN, State: KA,
  Locality: Mysore, Common name: spoke1_backup, Domain component: example.net
Subject string:
  C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
Alternate subject: "spoke1_backup@example.net", example.net, 10.3.3.1
Validity:
  Not before: 11- 9-2012 11:09
  Not after: 11- 9-2013 11:19
Public key algorithm: rsaEncryption(1024 bits)
  30:81:89:02:81:81:00:a7:02:b5:e2:cd:79:24:f8:97:a3:8d:4d:27
  8c:2b:dd:f1:57:72:4d:2b:6d:d5:95:0d:9c:1b:5c:e2:a4:b0:84:2e
  31:82:3c:91:08:a2:58:b9:30:4c:5f:a3:6b:e6:2b:9c:b1:42:dd:1c
  cd:a2:7a:84:ea:7b:a6:b7:9a:13:33:c6:27:2b:79:2a:b1:0c:fe:08
  4c:a7:35:fc:da:4f:df:1f:cf:f4:ba:bc:5a:05:06:63:92:41:b4:f2
  54:00:3f:ef:ff:41:e6:ca:74:10:56:f7:2b:5f:d3:1a:33:7e:49:74
  1c:42:cf:c2:23:ea:4b:8f:50:2c:eb:1c:a6:37:89:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Distribution CRL:
  http://ca-server1/CertEnroll/CASERVER1.crl
  file://\\ca-server1\CertEnroll\CASERVER1.crl
Fingerprint:
  d6:7f:52:a3:b6:f8:ae:cb:70:3f:a9:79:ea:8a:da:9e:ba:83:e4:5f (sha1)
  76:0b:72:73:cf:51:ee:58:81:2d:f7:b4:e2:5c:f4:5c (md5)
Auto-re-enrollment:
  Status: Disabled
  Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT` for Local1 and `SBU` for Local2. The IKE configurations on the hub include `OU=SLT` and `OU=SBU` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/2 unit 0 family inet address 10.1.2.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 10.20.20.1/24
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp-1 type internal
set protocols bgp group ibgp-1 local-address 10.10.10.1
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 cluster 10.2.3.4
set protocols bgp group ibgp-1 allow 10.10.10.0/24
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.1
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 cluster 10.2.3.5
set protocols bgp group ibgp-2 allow 10.20.20.0/24
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.2.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway hub-to-spoke-gw-1 ike-policy ike-policy-1
set security ike gateway hub-to-spoke-gw-1 dynamic distinguished-name wildcard OU=SLT
```

```
set security ike gateway hub-to-spoke-gw-1 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-1 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-1 external-interface ge-0/0/1.0
set security ike gateway hub-to-spoke-gw-2 ike-policy ike-policy-2
set security ike gateway hub-to-spoke-gw-2 dynamic distinguished-name wildcard OU=SBU
set security ike gateway hub-to-spoke-gw-2 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-2 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-2 external-interface ge-0/0/2.0
set security ipsec vpn-monitor-options interval 5
set security ipsec vpn-monitor-options threshold 2
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn-1 bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn-1 vpn-monitor source-interface ge-0/0/1.0
set security ipsec vpn hub-to-spoke-vpn-1 ike gateway hub-to-spoke-gw-1
set security ipsec vpn hub-to-spoke-vpn-1 ike ipsec-policy vpn-policy
set security ipsec vpn hub-to-spoke-vpn-2 bind-interface st0.1
set security ipsec vpn hub-to-spoke-vpn-2 vpn-monitor source-interface ge-0/0/2.0
set security ipsec vpn hub-to-spoke-vpn-2 ike gateway hub-to-spoke-gw-2
set security ipsec vpn hub-to-spoke-vpn-2 ike ipsec-policy vpn-policy
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
user@host# set ge-0/0/2 unit 0 family inet address 10.1.2.1/30
user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 10.20.20.1/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface ge-0/0/3.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.1
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 cluster 10.2.3.4
user@host# set group ibgp-1 allow 10.10.10.0/24
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.1
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 cluster 10.2.3.5
user@host# set group ibgp-2 allow 10.20.20.0/24
[edit routing-options]
user@host# set static route 10.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 10.3.3.0/30 next-hop 10.1.2.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
```

```
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway hub-to-spoke-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
[edit security ike gateway hub-to-spoke-gw-2]
user@host# set ike-policy ike-policy-2
user@host# set dynamic distinguished-name wildcard OU=SBU
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec vpn-monitor]
user@host# set options interval 5
user@host# set options threshold 2
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn hub-to-spoke-vpn-1]
user@host# set bind-interface st0.0
user@host# set vpn-monitor source-interface ge-0/0/1.0
user@host# set ike gateway hub-to-spoke-gw-1
user@host# set ike ipsec-policy vpn-policy
[edit security ipsec vpn hub-to-spoke-vpn-2]
user@host# set bind-interface st0.1
user@host# set vpn-monitor source-interface ge-0/0/2.0
```

```
user@host# set ike gateway hub-to-spoke-gw-2
user@host# set ike ipsec-policy vpn-policy
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.0
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces ge-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
```

```
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 10.1.2.1/30;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
        unit 1 {
            multipoint;
            family inet {
                address 10.20.20.1/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface ge-0/0/3.0;
    then accept;
}
[edit]
user@host# show protocols
```

```
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.1;
        export lan_nw;
        cluster 10.2.3.4;
        allow 10.10.10.0/24;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.1;
        export lan_nw;
        cluster 10.2.3.5;
        allow 10.20.20.0/24;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.2.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
```

```
        }
    }
    gateway hub-to-spoke-gw-1 {
        ike-policy ike-policy-1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
    gateway hub-to-spoke-gw-2 {
        ike-policy ike-policy-2;
        dynamic {
            distinguished-name {
                wildcard OU=SBU;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/2.0;
    }
[edit]
user@host# show security ipsec
vpn-monitor-options {
    interval 5;
    threshold 2;
}
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn-1 {
        bind-interface st0.0;
```

```
        vpn-monitor {
            source-interface ge-0/0/1.0;
        }
        ike {
            gateway hub-to-spoke-gw-1;
            ipsec-policy vpn-policy;
        }
    }
    vpn hub-to-spoke-vpn-2 {
        bind-interface st0.1;
        vpn-monitor {
            source-interface ge-0/0/2.0;
        }
        ike {
            gateway hub-to-spoke-gw-2;
            ipsec-policy vpn-policy;
        }
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        ge-0/0/1.0;
        ge-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
```

```
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/2 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set interfaces st0 unit 1 family inet address 10.20.20.2/24
set policy-options policy-statement default_route from protocol static
set policy-options policy-statement default_route from route-filter 0.0.0.0/0 exact
set policy-options policy-statement default_route then accept
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
```

```
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp-1 type internal
set protocols bgp group ibgp-1 local-address 10.10.10.2
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 neighbor 10.10.10.1
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.2
set protocols bgp group ibgp-2 export default_route
set protocols bgp group ibgp-2 neighbor 10.20.20.1
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set routing-options static route 10.1.2.0/30 next-hop 10.3.3.2
set routing-options static route 0.0.0.0/0 next-hop st0.1
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway spoke-to-hub-gw-1 ike-policy ike-policy-1
set security ike gateway spoke-to-hub-gw-1 address 10.1.1.1
set security ike gateway spoke-to-hub-gw-1 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 external-interface fe-0/0/1.0
set security ike gateway spoke-to-hub-gw-2 ike-policy ike-policy-2
set security ike gateway spoke-to-hub-gw-2 address 10.1.2.1
set security ike gateway spoke-to-hub-gw-2 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 external-interface fe-0/0/2.0
set security ipsec vpn-monitor-options interval 5
set security ipsec vpn-monitor-options threshold 2
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn spoke-to-hub-1 bind-interface st0.0
set security ipsec vpn spoke-to-hub-1 vpn-monitor destination-ip 10.1.1.1
set security ipsec vpn spoke-to-hub-1 ike gateway spoke-to-hub-gw-1
```

```
set security ipsec vpn spoke-to-hub-1 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-1 establish-tunnels immediately
set security ipsec vpn spoke-to-hub-2 bind-interface st0.1
set security ipsec vpn spoke-to-hub-2 vpn-monitor destination-ip 10.1.2.1
set security ipsec vpn spoke-to-hub-2 ike gateway spoke-to-hub-gw-2
set security ipsec vpn spoke-to-hub-2 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-2 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces fe-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
user@host# set fe-0/0/2 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
user@host# set st0 unit 0 family inet address 10.10.10.2/24
user@host# set st0 unit 1 family inet address 10.20.20.2/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement default_route from protocol static
user@host# set policy-statement default_route from route-filter 0.0.0.0/0 exact
```

```
user@host# set policy-statement default_route then accept
user@host# set policy-statement lan_nw from interface fe-0/0/4.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.2
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 neighbor 10.10.10.1
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.2
user@host# set group ibgp-2 export default_route
user@host# set group ibgp-2 neighbor 10.20.20.1
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
user@host# set static route 10.1.2.0/30 next-hop 10.3.3.2
user@host# set static route 0.0.0.0/0 next-hop st0.1
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway spoke-to-hub-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
[edit security ike gateway spoke-to-hub-gw-2]
user@host# set ike-policy ike-policy-2
user@host# set address 10.1.2.1
```

```
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec vpn-monitor]
user@host# set options interval 5
user@host# set options threshold 2
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub-1]
user@host# set bind-interface st0.0
user@host# set vpn-monitor destination-ip 10.1.1.1
user@host# set ike gateway spoke-to-hub-gw-1
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
[edit security ipsec vpn spoke-to-hub-2]
user@host# set bind-interface st0.1
user@host# set vpn-monitor destination-ip 10.1.2.1
user@host# set ike gateway spoke-to-hub-gw-2
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
user@host# set interfaces fe-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

```
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.3.1/30;
            }
        }
    }
    fe-0/0/4 {
```

```
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            family inet {
                address 10.10.10.2/24;
            }
        }
        unit 1 {
            family inet {
                address 10.20.20.2/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement default_route {
    from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
    }
    then accept;
}
    policy-statement lan_nw {
        from interface fe-0/0/4.0;
        then accept;
    }
[edit]
user@host# show protocols
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.2;
        export lan_nw;
        neighbor 10.10.10.1;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.2;
```

```
        export default_route;
        neighbor 10.20.20.1;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
    route 10.1.2.0/30 next-hop 10.3.3.2;
    route 0.0.0.0/0 next-hop st0.1;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
        }
    }
    gateway spoke-to-hub-gw-1 {
        ike-policy ike-policy-1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/1.0;
    }
    gateway spoke-to-hub-gw-2 {
        ike-policy ike-policy-2;
```

```
        address 10.1.2.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/2.0;
    }
[edit]
user@host# show security ipsec
vpn-monitor-options {
    interval 5;
    threshold 2;
}
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub-1 {
        bind-interface st0.0;
        vpn-monitor {
            destination-ip 10.1.1.1;
        }
        ike {
            gateway spoke-to-hub-gw-1;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
    }
    vpn spoke-to-hub-2 {
        bind-interface st0.1;
        vpn-monitor {
            destination-ip 10.1.2.1;
        }
        ike {
            gateway spoke-to-hub-gw-2;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
```

```
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
        fe-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
```

```
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Phase 1 Status (Both Tunnels Are Up) | **1067**
- Verifying IPsec Phase 2 Status (Both Tunnels Are Up) | **1068**
- Verifying IPsec Next-Hop Tunnels (Both Tunnels Are Up) | **1069**
- Verifying BGP (Both Tunnels Are Up) | **1069**
- Verifying Learned Routes (Both Tunnels Are Up) | **1070**
- Verifying IKE Phase 1 Status (Primary Tunnel Is Down) | **1071**
- Verifying IPsec Phase 2 Status (Primary Tunnel Is Down) | **1071**
- Verifying IPsec Next-Hop Tunnels (Primary Tunnel Is Down) | **1072**
- Verifying BGP (Primary Tunnel Is Down) | **1072**
- Verifying Learned Routes (Primary Tunnel Is Down) | **1073**

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status (Both Tunnels Are Up)

Purpose

Verify the IKE Phase 1 status when both IPSec VPN tunnels are up.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
3733075 UP     d4f51c28c0a82101  05b125993a864d3c  Main           10.3.3.1
3733076 UP     d53c8a0b7d4c319b  c23c5f7a26388247  Main           10.2.2.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

Verifying IPsec Phase 2 Status (Both Tunnels Are Up)

Purpose

Verify the IPsec Phase 2 status when both IPsec VPN tunnels are up.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173316 ESP:des/ md5 3cd96946 3555/ unlim U   root 500   10.2.2.1
  >268173316 ESP:des/ md5 1c09b9b 3555/ unlim  U   root 500   10.2.2.1
  <268173313 ESP:des/ md5 7c6ffca3 3340/ unlim U   root 500   10.3.3.1
  >268173313 ESP:des/ md5 33bf6f2f 3340/ unlim U   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying IPsec Next-Hop Tunnels (Both Tunnels Are Up)

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                  Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn-1              Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
10.20.20.2        st0.1       hub-to-spoke-vpn-2              Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spoke. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP (Both Tunnels Are Up)

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spoke when both IPsec VPN tunnels are up.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          2          0          0          0          0
```

```
Peer                    AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.10.10.2              65010       5          6       0       0          54
1/1/1/0              0/0/0/0
10.20.20.2              65010      13         16       0       0        4:29
1/1/1/0              0/0/0/0
```

Verifying Learned Routes (Both Tunnels Are Up)

Purpose

Verify that routes to the spoke have been learned when both tunnels are up. The route to 10.60.60.0/24 is through the st0.0 interface and the default route is through the st0.1 interface.

Action

From operational mode, enter the **show route 10.60.60.0** command.

```
user@host> show route 10.60.60.0
inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

60.60.60.0/24      *[BGP/170] 00:01:11, localpref 100
                      AS path: I
                    > to 10.10.10.2 via st0.0
```

From operational mode, enter the **show route 0.0.0.0** command.

```
user@host> show route 0.0.0.0
inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:04:55, localpref 100
                      AS path: I
                    > to 10.20.20.2 via st0.1
```

Verifying IKE Phase 1 Status (Primary Tunnel Is Down)

Purpose

Verify the IKE Phase 1 status when the primary tunnel is down.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
3733075 UP     d4f51c28c0a82101  05b125993a864d3c  Main           10.3.3.1
3733076 UP     d53c8a0b7d4c319b  c23c5f7a26388247  Main           10.2.2.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

Verifying IPsec Phase 2 Status (Primary Tunnel Is Down)

Purpose

Verify the IPsec Phase 2 status when the primary tunnel is down.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
   Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173313 ESP:des/ md5 7c6ffca3 3156/ unlim U   root 500   10.3.3.1
  >268173313 ESP:des/ md5 33bf6f2f 3156/ unlim U   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying IPsec Next-Hop Tunnels (Primary Tunnel Is Down)

Purpose

Verify the IPsec next-hop tunnel.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                   Flag     IKE-
ID                            XAUTH username
10.20.20.2        st0.1       hub-to-spoke-vpn-2              Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spoke. The next hop should be associated with the correct IPsec VPN name, in this case the backup VPN tunnel.

Verifying BGP (Primary Tunnel Is Down)

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spoke when the primary tunnel is down.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 2 Peers: 1 Down peers: 0
Unconfigured peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 1          1          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.20.20.2               10         20         24       0       0        7:24
1/1/1/0              0/0/0/0
```

Verifying Learned Routes (Primary Tunnel Is Down)

Purpose

Verify that routes to the spoke have been learned when the primary tunnel is down. Both the route to 10.60.60.0/24 and the default route are through the st0.1 interface.

Action

From operational mode, enter the **show route 10.60.60.0** command.

```
user@host> show route 60.60.60.0
inet.0: 46 destinations, 46 routes (45 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:07:41, localpref 100
                      AS path: I
                    > to 10.20.20.2 via st0.1
```

From operational mode, enter the **show route 0.0.0.0** command.

```
user@host> show route 0.0.0.0
inet.0: 46 destinations, 46 routes (45 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[BGP/170] 00:07:47, localpref 100
                      AS path: I
                    > to 10.20.20.2 via st0.1
```

Example: Configuring Basic AutoVPN with OSPF

IN THIS SECTION

- Requirements | **1074**
- Overview | **1075**
- Configuration | **1078**
- Verification | **1104**

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures OSPF to forward packets through the VPN tunnels using the certificate based authentication. For authentication with preshared key, set up a similar configuration shown at "Example: Configuring Basic AutoVPN with iBGP" on page 929.

Requirements

This example uses the following hardware and software components:

- Three supported firewalls as AutoVPN hub and spokes
- Supported Junos OS Release

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

- Topology | 1077

This example shows the configuration of an AutoVPN hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 110 on page 1075 shows the options used in this example.

Table 110: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPF Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main

Table 110: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPF Configurations *(Continued)*

Option	Value
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 111 on page 1076 shows the options configured on the hub and on all spokes.

Table 111: AutoVPN Basic OSPF Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	10.1.1.1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string SLT in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate

Table 111: AutoVPN Basic OSPF Configuration for Hub and All Spokes *(Continued)*

Option	Hub	All Spokes
External interface	ge-0/0/1.0	Spoke 1: fe-0/0/1.0 Spoke 2: ge-0/0/1.0
VPN:		
Bind interface	st0.0	st0.0
Establish tunnels	(not configured)	Immediately on configuration commit

Table 112 on page 1077 shows the configuration options that are different on each spoke.

Table 112: Comparison Between the Basic OSPF Spoke Configurations

Option	Spoke 1	Spoke 2
st0.0 interface	10.10.10.2/24	10.10.10.3/24
Interface to internal network	fe-0.0/4.0: 100.60.60.1/24	fe-0.0/4.0: 10.70.70.1/24
Interface to Internet	fe-0/0/1.0: 10.2.2.1/30	ge-0/0/1.0: 10.3.3.1/30

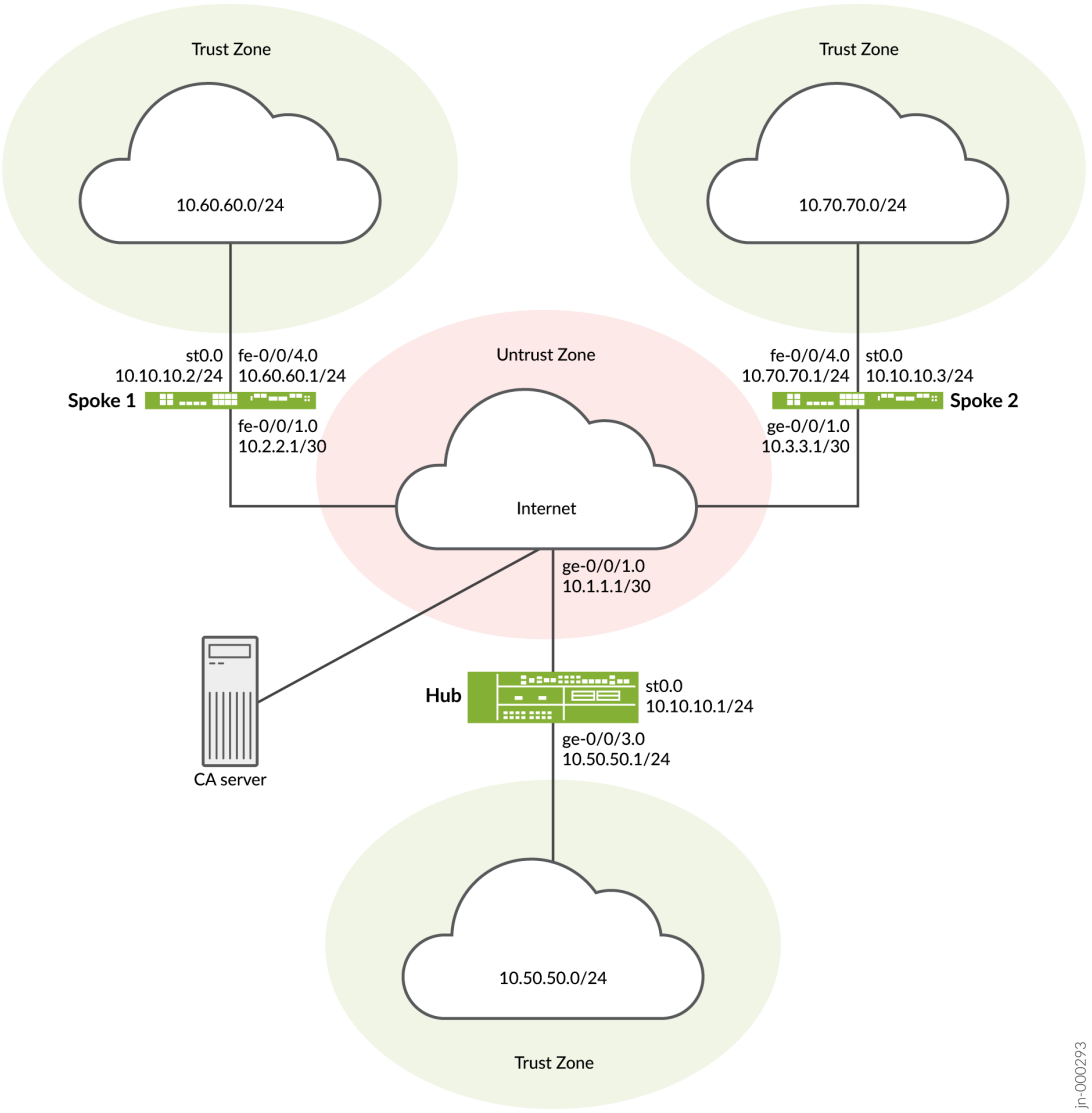
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments.

Topology

Figure 61 on page 1078 shows the firewalls to be configured for AutoVPN in this example.

Figure 61: Basic AutoVPN Deployment with OSPF



Configuration

IN THIS SECTION

- Enroll Device Certificates with SCEP | **1079**
- Configuring the Hub | **1084**
- Configuring Spoke 1 | **1091**

- Configuring Spoke 2 | 1097

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
```

```
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
```

```
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
```

```
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Tumkur, Common name: spoke2, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
  Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 6-2012 10:02
    Not after: 11- 6-2013 10:12
```

```
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
    27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
    77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
    44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
    7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
    7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
    58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
    00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
set protocols ospf area 0.0.0.0 interface st0.0 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.0 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.1.2
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
```

```
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway hub-to-spoke-gw ike-policy ike-policy1
set security ike gateway hub-to-spoke-gw dynamic distinguished-name wildcard OU=SLT
set security ike gateway hub-to-spoke-gw dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw local-identity distinguished-name
set security ike gateway hub-to-spoke-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn ike gateway hub-to-spoke-gw
set security ipsec vpn hub-to-spoke-vpn ike ipsec-policy vpn-policy1
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
```

```
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
```

2. Configure the routing protocol.

```
[edit protocols ospf]
user@host# set area 0.0.0.0 interface st0.0 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.0 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/3.0
[edit routing-options]
user@host# set static route 2.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 3.3.3.0/30 next-hop 10.1.1.2
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway hub-to-spoke-gw]
user@host# set ike-policy ike-policy1
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
```

```
[edit security ipsec vpn hub-to-spoke-vpn]
user@host# set bind-interface st0.0
user@host# set ike gateway hub-to-spoke-gw
user@host# set ike ipsec-policy vpn-policy1
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
```

```
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
    }
[edit]
user@host# show protocols
ospf {
    area 0.0.0.0 {
        interface st0.0 {
            interface-type p2mp;
            dynamic-neighbors;
        }
        interface ge-0/0/3.0;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.1.2;
}
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
```

```
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway hub-to-spoke-gw {
        ike-policy ike-policy1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
traceoptions {
    flag all;
}
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn {
        bind-interface st0.0;
        ike {
            gateway hub-to-spoke-gw;
            ipsec-policy vpn-policy1;
```

```
        }
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        ge-0/0/1.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
```

```
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set protocols ospf area 0.0.0.0 interface st0.0 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
set protocols ospf area 0.0.0.0 interface fe-0/0/4.0
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface fe-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
```

```
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.2/24
```

2. Configure the routing protocol.

```
[edit protocols ospf]
user@host# set area 0.0.0.0 interface st0.0 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
user@host# set area 0.0.0.0 interface fe-0/0/4.0
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
```

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
```

```
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.2/24;
            }
        }
    }
[edit]
user@host# show protocols
ospf {
    area 0.0.0.0 {
        interface st0.0 {
            interface-type p2mp;
            neighbor 10.10.10.1;
        }
        interface fe-0/0/4.0;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
}
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
```

```
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
```

```
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.70.70.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.3/24
```

```
set protocols ospf area 0.0.0.0 interface st0.0 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
set protocols ospf area 0.0.0.0 interface fe-0/0/4.0
set routing-options static route 10.1.1.1/32 next-hop 10.3.3.2
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.70.70.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.3/24
```

2. Configure the routing protocol.

```
[edit protocols ospf]
user@host# set area 0.0.0.0 interface st0.0 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
user@host# set area 0.0.0.0 interface fe-0/0/4.0
[edit routing-options]
user@host# set static route 10.1.1.1/32 next-hop 10.3.3.2
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
```

```
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show`

`security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.3.3.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.70.70.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.3/24;
            }
        }
    }
[edit]
user@host# show protocols
ospf {
    area 0.0.0.0 {
        interface st0.0 {
            interface-type p2mp;
            neighbor 10.10.10.1;
        }
        interface fe-0/0/4.0;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.1/32 next-hop 10.3.3.2;
```

```
}
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
```

```
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
```

```
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Phase 1 Status | **1104**
- Verifying IPsec Phase 2 Status | **1105**
- Verifying IPsec Next-Hop Tunnels | **1105**
- Verifying OSPF | **1106**
- Verifying Learned Routes | **1106**

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5480159 UP     22432fb6f7fbc389  412b751f79b45099  Main           10.2.2.1
5480161 UP     d455050707bc3eaf  b3dde111232270d2  Main           10.3.3.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external

interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173400 ESP:des/ md5 f38eea12 2954/ unlim -   root 500   10.2.2.1
  >268173400 ESP:des/ md5 bb48d228 2954/ unlim -   root 500   10.2.2.1
  <268173401 ESP:des/ md5 bcd1390b 3530/ unlim -   root 500   10.3.3.1
  >268173401 ESP:des/ md5 77fcf6e2 3530/ unlim -   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                  Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn                Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
10.10.10.3        st0.0       hub-to-spoke-vpn                Auto     C=IN, DC=example.net,
ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying OSPF

Purpose

Verify that OSPF references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show ospf neighbor** command.

```
user@host> show ospf neighbor
Address          Interface             State     ID               Pri  Dead
10.10.10.3       st0.0                 Full      10.255.226.179   128    32
10.10.10.2       st0.0                 Full      10.207.36.182    128    38
```

Verifying Learned Routes

Purpose

Verify that routes to the spokes have been learned.

Action

From operational mode, enter the **show route 60.60.60.0** command.

```
user@host> show route 10.60.60.0
 inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.60.60.0/24      *[OSPF/10] 00:51:13, metric 2
                    > to 10.10.10.2 via st0.0
```

From operational mode, enter the **show route 10.70.70.0** command.

```
user@host> show route 10.70.70.0
inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.70.70.0/24      *[OSPF/10] 00:51:48, metric 2
                    > to 10.10.10.3 via st0.0
```

Example: Configuring AutoVPN with OSPFv3 for IPv6 Traffic

IN THIS SECTION

- Requirements | **1108**
- Overview | **1108**
- Configuration | **1112**
- Verification | **1140**

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures AutoVPN for IPv6 environment using OSPFv3 to forward packets through the VPN tunnels using the certificate based authentication. For authentication with preshared key, set up a similar configuration shown at "Example: Configuring Basic AutoVPN with iBGP" on page 929.

Requirements

This example uses the following hardware and software components:

- Three supported firewalls as AutoVPN hub and spokes.
- Supported Junos OS Release.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

- Topology | **1111**

This example shows the configuration of an AutoVPN with OSPFv3 routing protocol on hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 113 on page 1108 shows the options used in this example.

Table 113: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPFv3 Configurations

Option	Value
IKE proposal:	

Table 113: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPFv3 Configurations *(Continued)*

Option	Value
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	19
Authentication algorithm	SHA-384
Encryption algorithm	AES 256 CBC
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Lifetime seconds	3000
Encryption algorithm	AES 256 GCM
IPsec policy:	
Perfect Forward Secrecy (PFS) group	19

The same certificate authority (CA) is configured on all devices.

Table 114 on page 1110 shows the options configured on the hub and on all spokes.

Table 114: AutoVPN OSPFv3 Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	2001:db8:2000::1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string SLT in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/0	Spoke 1: ge-0/0/0.0 Spoke 2: ge-0/0/0.0
VPN:		
Bind interface	st0.1	st0.1
Establish tunnels	(not configured)	Immediately on configuration commit

Table 115 on page 1110 shows the configuration options that are different on each spoke.

Table 115: Comparison Between the OSPFv3 Spoke Configurations

Option	Spoke 1	Spoke 2
st0.1 interface	2001:db8:7000::2/64	2001:db8:7000::3/64
Interface to internal network	(ge-0/0/1.0) 2001:db8:4000::1/64	(ge-0/0/1.0) 2001:db8:6000::1/64
Interface to Internet	(ge-0/0/0.0) 2001:db8:3000::2/64	(ge-0/0/0.0) 2001:db8:5000::2/64

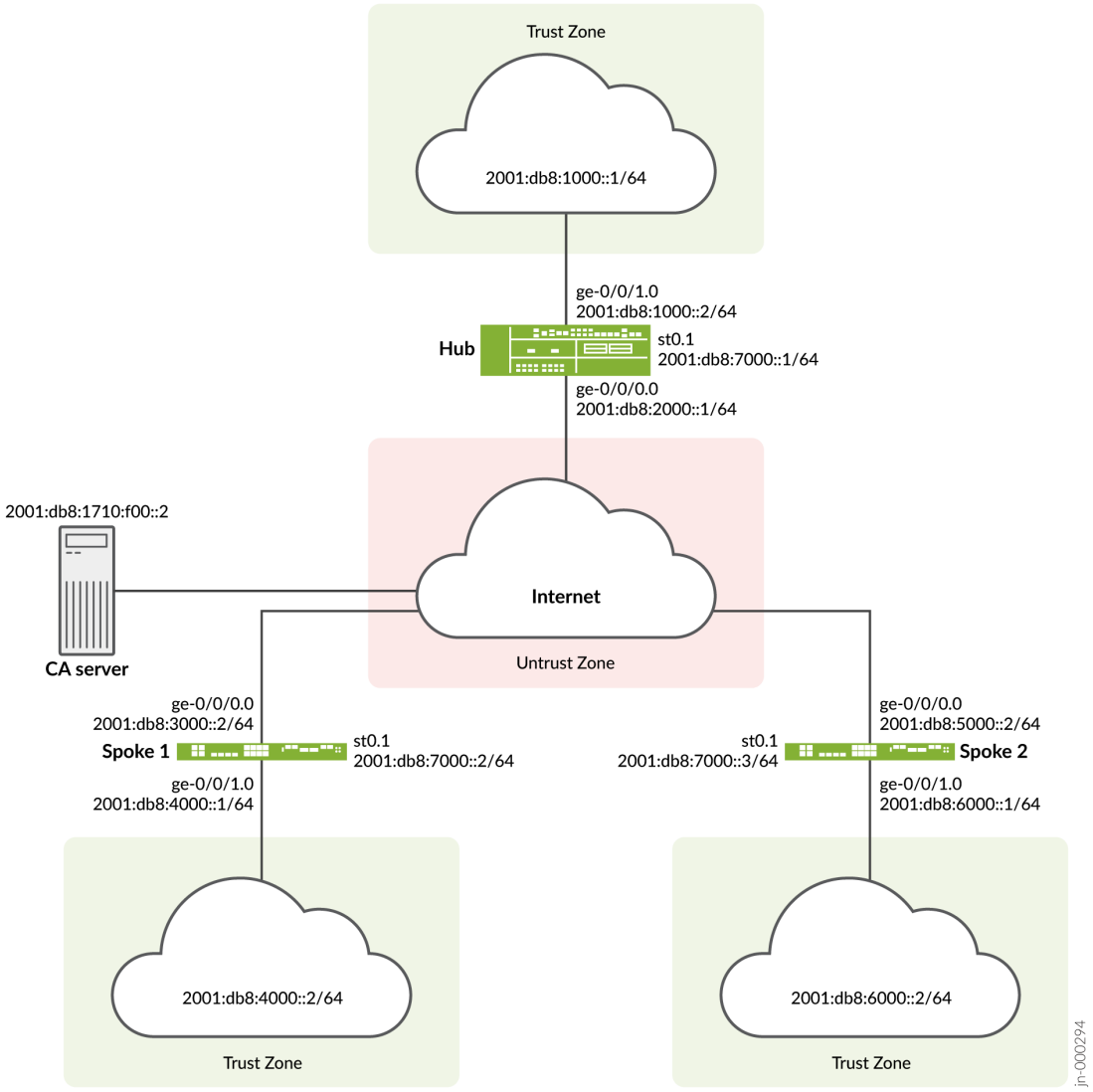
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments.

Topology

Figure 62 on page 1111 shows the firewalls to be configured for AutoVPN in this example.

Figure 62: Basic AutoVPN Deployment with OSPFv3



Configuration

IN THIS SECTION

- Enroll Device Certificates with SCEP | **1112**
- Configuring the Hub | **1117**
- Configuring Spoke 1 | **1125**
- Configuring Spoke 2 | **1133**

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2020 09:39
    Not after: 11- 6-2021 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
```

```
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2020 09:40
    Not after: 11- 6-2021 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
```

```
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Tumkur, Common name: spoke2, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
  Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 6-2020 10:02
    Not after: 11- 6-2021 10:12
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
    27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
    77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
    44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
    7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
    7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
    58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
    00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
```

```
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate HUB
set security ike gateway IKE_GWA_1 ike-policy IKE_POL
set security ike gateway IKE_GWA_1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway IKE_GWA_1 dead-peer-detection always-send
set security ike gateway IKE_GWA_1 dead-peer-detection interval 10
set security ike gateway IKE_GWA_1 dead-peer-detection threshold 3
set security ike gateway IKE_GWA_1 local-identity distinguished-name
set security ike gateway IKE_GWA_1 external-interface ge-0/0/0
set security ike gateway IKE_GWA_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPNA_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPNA_1 ike gateway IKE_GWA_1
set security ipsec vpn IPSEC_VPNA_1 ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/l..0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::1/64
set routing-options rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::1
set routing-options rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::1
set protocols ospf3 traceoptions file ospf
```

```
set protocols ospf3 traceoptions flag all
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::1/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set traceoptions file ospf
user@host# set traceoptions flag all
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/1.0
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::1
user@host# set rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::1
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
```

```
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate HUB
[edit security ike gateway IKE_GWA_1]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_1
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

```
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set pki ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:2000::1/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:1000::2/64;
            }
```

```
        }
    }
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:7000::1/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    traceoptions {
        file ospf;
        flag all;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:3000::/64 next-hop 2001:db8::1;
    route 2001:db8:5000::/64 next-hop 2001:db8::1;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
```

```
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
    policy IKE_POL {
        mode main;
        proposals IKE_PROP;
        certificate {
            local-certificate HUB;
        }
    }
    gateway IKE_GWA_1 {
        ike-policy IKE_POL;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
                }
            }
            dead-peer-detection {
                always-send;
                interval 10;
                threshold 3;
            }
        local-identity distinguished-name;
        external-interface ge-0/0/0.0;
        version v1-only;
    }
[edit]
user@host# show security ipsec
    proposal IPSEC_PROP {
        protocol esp;
        authentication-algorithm aes-256-gcm;
        set lifetime-seconds 3000;
    }
    policy IPSEC_POL {
        perfect-forward-secrecy {
            keys group19;
        }
        proposals IPSEC_PROP;
    }
    vpn IPSEC_VPNA_1 {
        bind-interface st0.1;
        ike {
            gateway IKE_GWA_1;
```

```
            ipsec-policy IPSEC_POL;
        }
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                ospf3;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
```

```
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE1
set security ike gateway IKE_GW_SPOKE_1 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_1 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_1 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_1 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_1 external-interface ge-0/0/0.0
```

```
set security ike gateway IKE_GW_SPOKE_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike gateway IKE_GW_SPOKE_1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_1 establish-tunnels immediately
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::2/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::2
set protocols ospf3 traceoptions file ospf
set protocols ospf3 traceoptions flag all
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::2/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set traceoptions file ospf
user@host# set traceoptions flag all
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/1.0
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::2
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE1
[edit security ike gateway IKE_GW_SPOKE_1]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROPl]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GW_SPOKE_1
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
```

```
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:3000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:4000::1/64;
        }
    }
}
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:7000::2/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    traceoptions {
        file ospf;
        flag all;
    }
    area 0.0.0.0 {
```

```
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE1;
    }
}
gateway IKE_GW_SPOKE_1 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
```

```
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_1;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
    security-zone trust {
```

```
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                ospf3;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE2
set security ike gateway IKE_GW_SPOKE_2 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_2 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_2 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_2 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_2 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_2 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_2 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike gateway IKE_GW_SPOKE_2
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_2 establish-tunnels on-traffic
```

```
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::3/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
set protocols ospf3 traceoptions file ospf
set protocols ospf3 traceoptions flag all
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::3/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set traceoptions file ospf
user@host# set traceoptions flag all
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/1.0
```

```
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE2
[edit security ike gateway IKE_GW_SPOKE_2]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP1]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_2]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GW_SPOKE_2
```

```
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels on-traffic
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show`

`security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:5000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:6000::1/64;
        }
    }
}
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:7000::3/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    traceoptions {
        file ospf;
        flag all;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
```

```
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE2;
    }
}
gateway IKE_GW_SPOKE_2 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
```

```
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_2 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_2;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels on-traffic;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                ospf3;
            }
        }
```

```
        interfaces {
            ge-0/0/0.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying IKE Status | **1141**
- Verifying IPsec Status | **1141**
- Verifying IPsec Next-Hop Tunnels | **1142**
- Verifying OSPFv3 | **1143**

Confirm that the configuration is working properly.

Verifying IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the **show security ike sa** command.

```
user@host> show security ike sa
Index  State Initiator cookie          Responder cookie          Mode Remote Address

493333 UP    2001:db8:88b49d915e684c93 2001:db8:fe890b1cac8522b5 Main 2001:db8:3000::2

493334 UP    2001:db8:26e40244ad3d722d 2001:db8:68b4d9f94097d32e Main 2001:db8:5000::2
```

Meaning

The `show security ike sa` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec sa** command.

```
user@host> show security ipsec sa
Total active tunnels: 2
  ID        Algorithm      SPI  Life:sec/kb    Mon    lsys Port Gateway
  >67108885 ESP:aes-gcm-256/None fdef4dab 2918/ unlim - root 500  2001:db8:3000::2
  >67108885 ESP:aes-gcm-256/None e785dadc 2918/ unlim - root 500  2001:db8:3000::2
```

```
  >67108887 ESP:aes-gcm-256/None 34a787af 2971/ unlim - root 500  2001:db8:5000::2
  >67108887 ESP:aes-gcm-256/None cf57007f 2971/ unlim - root 500  2001:db8:5000::2
```

Meaning

The `show security ipsec sa` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway              interface  IPSec VPN name  Flag  IKE-
ID                                XAUTH username

2001:db8:9000::2              st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available

2001:db8:9000::3              st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available

2001:db8::5668:ad10:fcd8:163c st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available

2001:db8::5668:ad10:fcd8:18a1 st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying OSPFv3

Purpose

Verify that OSPFv3 references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show ospf3 neighbor detail** command.

Hub:

```
user@host> show ospf3 neighbor detail
ID                        Interface   State  Pri  Dead
2001:db8:7000:2   st0.1       Full   128   -
  Neighbor-address 2001:db8::5668:ad10:fcd8:18a1
  Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
  DR-ID 0.0.0.0, BDR-ID 0.0.0.0
  Up 00:01:35, adjacent 00:01:31 Hello suppressed 00:01:31 ago
2001:db8:7000:3  st0.1                  Full      128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:163c
  Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
  DR-ID 0.0.0.0, BDR-ID 0.0.0.0
  Up 00:01:41, adjacent 00:01:37 Hello suppressed 00:01:37 ago
```

Spoke 1:

```
user@host> show ospf3 neighbor detail
ID                        Interface     State     Pri   Dead
2001:db8:7000:1   st0.1         Full      128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:1946
  Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
  DR-ID 0.0.0.0, BDR-ID 0.0.0.0
  Up 00:05:38, adjacent 00:05:38 Hello suppressed 00:05:34 ago
```

Spoke 2:

```
user@host> show ospf3 neighbor detail
ID                       Interface     State     Pri   Dead
2001:db8:7000:1 st0.1         Full      128      -
```

```
Neighbor-address 2001:db8::5668:ad10:fcd8:1946
Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
DR-ID 0.0.0.0, BDR-ID 0.0.0.0
Up 00:04:44, adjacent 00:04:44 Hello suppressed 00:04:40 ago
```

Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors

IN THIS SECTION

- Requirements | **1144**
- Overview | **1145**
- Configuration | **1148**
- Verification | **1161**

This example shows how to configure traffic selectors, instead of dynamic routing protocols, to forward packets through a VPN tunnel in an AutoVPN deployment. When traffic selectors are configured, the secure tunnel (st0) interface must be in point-to-point mode. Traffic selectors are configured on both the hub and spoke devices. The example uses certificate based authentication. For authentication with preshared key, set up a similar configuration shown at "Example: Configuring Basic AutoVPN with iBGP" on page 929.

Requirements

This example uses the following hardware and software components:

- Two firewalls connected and configured in a chassis cluster. The chassis cluster is the AutoVPN hub.
- One firewall configured as an AutoVPN spoke.
- Supported Junos OS Release.
- Digital certificates enrolled in the hub and the spoke devices that allow the devices to authenticate each other.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.
- Enroll the digital certificates in each device. See Enroll Certificate.

Overview

IN THIS SECTION

- Topology | **1147**

In this example, traffic selectors are configured on the AutoVPN hub and spoke. Only traffic that conforms to the configured traffic selector is forwarded through the tunnel. On the hub, the traffic selector is configured with the local IP address 192.0.0.0/8 and the remote IP address 172.0.0.0/8. On the spoke, the traffic selector is configured with the local IP address 172.0.0.0/8 and the remote IP address 192.0.0.0/8.

The traffic selector IP addresses configured on the spoke can be a subset of the traffic selector IP addresses configured on the hub. This is known as *traffic selector flexible match*.

Certain Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hubs and spokes must have the same values. Table 116 on page 1145 shows the values used in this example:

Table 116: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors

Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5
Authentication algorithm	sha-1
Encryption algorithm	aes-256-cbc

Table 116: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors *(Continued)*

Option	Value
IKE policy:	
Mode	main
Certificate	local-certificate
IKE gateway:	
Dynamic	distinguished name wildcard DC=Common_component
IKE user type	group IKE id
Local identity	distinguished name
Version	v1-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-192-cbc
Lifetime	3600 seconds 150,000 kilobytes
IPsec policy:	

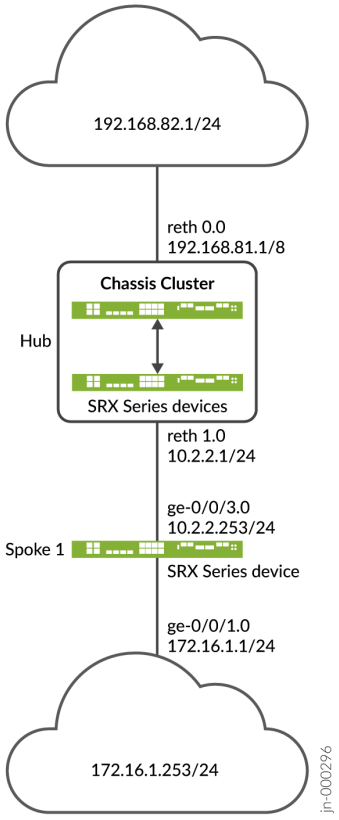
Table 116: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors *(Continued)*

Option	Value
Perfect Forward Secrecy (PFS) group	group5

Topology

Figure 63 on page 1147 shows the firewalls to be configured for this example.

Figure 63: AutoVPN with Traffic Selectors



Configuration

IN THIS SECTION

- Configuring the Hub | 1148
- Configuring the Spoke | 1155

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-8/0/2 gigether-options redundant-parent reth1
set interfaces ge-8/0/3 gigether-options redundant-parent reth0
set interfaces lo0 unit 0 family inet address 10.100.1.100/24
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.168.81.1/8
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.2.1/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
set security ike policy ikepol1 mode main
set security ike policy ikepol1 proposals prop_ike
set security ike policy ikepol1 certificate local-certificate Hub_ID
set security ike gateway HUB_GW ike-policy ikepol1
set security ike gateway HUB_GW dynamic distinguished-name wildcard DC=Domain_component
set security ike gateway HUB_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_GW local-identity distinguished-name
set security ike gateway HUB_GW external-interface reth1
```

```
set security ike gateway HUB_GW version v1-only
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-192-cbc
set security ipsec proposal prop_ipsec lifetime-seconds 3600
set security ipsec proposal prop_ipsec lifetime-kilobytes 150000
set security ipsec policy ipsecpol1 perfect-forward-secrecy keys group5
set security ipsec policy ipsecpol1 proposals prop_ipsec
set security ipsec vpn HUB_VPN bind-interface st0.1
set security ipsec vpn HUB_VPN ike gateway HUB_GW
set security ipsec vpn HUB_VPN ike ipsec-policy ipsecpol1
set security ipsec vpn HUB_VPN traffic-selector ts1 local-ip 192.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 172.0.0.0/8
set security pki ca-profile rsa ca-identity rsa
set security pki ca-profile rsa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

You can configure the CLI option `reject-duplicate-connection` at the [`edit security ike gateway` *`gateway-name`* `dynamic`] hierarchy level to retain an existing tunnel session and reject negotiation requests for a new tunnel with the same IKE ID. By default, an existing tunnel is tear down when a new tunnel with the same IKE ID is established. The `reject-duplicate-connection` option is only supported when `ike-user-type group-ike-id` or `ike-user-type shared-ike-id` is configured for the IKE gateway; the `aaa access-profile` *`profile-name`* configuration is not supported with this option.

Use the CLI option `reject-duplicate-connection` only when you are certain that reestablishment of a new tunnel with the same IKE ID should be rejected.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure the hub:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/2 gigether-options redundant-parent reth1
user@host# set ge-0/0/3 gigether-options redundant-parent reth0
user@host# set ge-8/0/2 gigether-options redundant-parent reth1
user@host# set ge-8/0/3 gigether-options redundant-parent reth0
user@host# set lo0 unit 0 family inet address 10.100.1.100/24
user@host# set lo0 redundant-pseudo-interface-options redundancy-group 1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 192.168.81.1/8
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 10.2.2.1/24
user@host# set st0 unit 1 family inet
```

2. Configure Phase 1 options.

```
[edit security ike proposal prop_ike]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ikepol1]
user@host# set mode main
user@host# set proposals prop_ike
user@host# set certificate local-certificate Hub_ID
[edit security ike gateway HUB_GW]
user@host# set ike-policy ikepol1
user@host# set dynamic distinguished-name wildcard DC=Domain_component
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface reth1
user@host# set version v1-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-192-cbc
```

```
user@host# set lifetime-seconds 3600
user@host# set lifetime-kilobytes 150000
[edit security ipsec policy ipsecpol1]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec HUB_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway HUB_GW
user@host# set ike ipsec-policy ipsecpol1
user@host# set traffic-selector ts1 local-ip 192.0.0.0/8
user@host# set traffic-selector ts1 remote-ip 172.0.0.0/8
```

4. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile rsa ca-identity rsa
user@host# set ca-profile rsa revocation-check disable
```

5. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces lo0.0
user@host# set interfaces reth1.0
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output

does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-0/0/3 {
    gigether-options {
        redundant-parent reth0;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.100.1.100/24;
        }
    }
    redundant-pseudo-interface-options {
        redundancy-group 1;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.168.81.1/8;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.2.2.1/24;
```

```
        }
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
[edit]
user@host# show security ike
proposal prop_ike {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy ikepol1 {
    mode main;
    proposals prop_ike;
    certificate {
        local-certificate Hub_ID;
    }
}
gateway HUB_GW {
    ike-policy ikepol1;
    dynamic distinguished-name wildcard DC=Domain_component;
    dynamic ike-user-type group-ike-id;
    local-identity distinguished-name;
    external-interface reth1;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal prop_ipsec {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-192-cbc;
    lifetime-seconds 3600;
    lifetime-kilobytes 150000;
}
policy ipsecpol1 {
    perfect-forward-secrecy {
        keys group5;
```

```
    }
    proposals prop_ipsec;
}
vpn HUB_VPN {
    bind-interface st0.1;
    ike {
        gateway HUB_GW;
        ipsec-policy ipsecpol1;
    }
    traffic-selector ts1 {
        local-ip 192.0.0.0/8;
        remote-ip 172.0.0.0/8;
    }
}
[edit]
user@host# show security pki
ca-profile rsa {
    ca-identity rsa;
    revocation-check {
        disable;
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
        reth0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
```

```
        protocols {
            all;
        }
    }
    interfaces {
        lo0.0;
        reth1.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Spoke

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.2.2.253/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
set security ike policy ikepol1 mode main
set security ike policy ikepol1 proposals prop_ike
set security ike policy ikepol1 certificate local-certificate Spoke1_ID
set security ike gateway SPOKE_GW ike-policy ikepol1
set security ike gateway SPOKE_GW address 10.2.2.1
set security ike gateway SPOKE_GW local-identity distinguished-name
set security ike gateway SPOKE_GW remote-identity distinguished-name container
DC=Domain_component
set security ike gateway SPOKE_GW external-interface ge-0/0/3.0
set security ike gateway SPOKE_GW version v1-only
```

```
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-192-cbc
set security ipsec proposal prop_ipsec lifetime-seconds 3600
set security ipsec proposal prop_ipsec lifetime-kilobytes 150000
set security ipsec policy ipsecpol1 perfect-forward-secrecy keys group5
set security ipsec policy ipsecpol1 proposals prop_ipsec
set security ipsec vpn SPOKE_VPN bind-interface st0.1
set security ipsec vpn SPOKE_VPN ike gateway SPOKE_GW
set security ipsec vpn SPOKE_VPN ike ipsec-policy ipsecpol1
set security ipsec vpn SPOKE_VPN traffic-selector ts1 local-ip 172.0.0.0/8
set security ipsec vpn SPOKE_VPN traffic-selector ts1 remote-ip 192.0.0.0/8
set security ipsec vpn SPOKE_VPN establish-tunnels immediately
set security pki ca-profile rsa ca-identity rsa
set security pki ca-profile rsa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure the hub:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 172.16.1.1/24
user@host# set ge-0/0/3 unit 0 family inet address 10.2.2.253/24
user@host# set st0 unit 1 family inet
```

2. Configure Phase 1 options.

```
[edit security ike proposal prop_ike]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ikepol1]
user@host# set mode main
user@host# set proposals prop_ike
user@host# set certificate local-certificate Spoke1_ID
[edit security ike gateway SPOKE_GW]
user@host# set ike-policy ikepol1
user@host# set address 10.2.2.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container DC=Domain_component
user@host# set external-interface ge-0/0/3.0
user@host# set version v1-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-192-cbc
user@host# set lifetime-seconds 3600
user@host# set lifetime-kilobytes 150000
[edit security ipsec policy ipsecpol1]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec SPOKE_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway SPOKE_GW
user@host# set ike ipsec-policy ipsecpol1
user@host# set traffic-selector ts1 local-ip 172.0.0.0/8
user@host# set traffic-selector ts1 remote-ip 192.0.0.0/8
user@host# set establish-tunnels immediately
```

4. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile rsa ca-identity rsa
user@host# set ca-profile rsa revocation-check disable
```

5. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/3.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.16.1.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
```

```
            address 10.2.2.253/24;
        }
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
[edit]
user@host# show security ike
proposal prop_ike {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy ikepol1 {
    mode main;
    proposals prop_ike;
    certificate {
        local-certificate Spoke1_ID;
    }
}
gateway SPOKE_GW {
    ike-policy ikepol1;
    address 10.2.2.1;
    local-identity distinguished-name;
    remote-identity distinguished-name container DC=Domain_component;
    external-interface ge-0/0/3.0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal prop_ipsec {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-192-cbc;
    lifetime-seconds 3600;
    lifetime-kilobytes 150000;
}
policy ipsecpol1 {
    perfect-forward-secrecy {
```

```
        keys group5;
    }
    proposals prop_ipsec;
}
vpn SPOKE_VPN {
    bind-interface st0.1;
    ike {
        gateway SPOKE_GW;
        ipsec-policy ipsecpol1;
    }
    traffic-selector ts1 {
        local-ip 172.0.0.0/8;
        remote-ip 192.0.0.0/8;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security pki
ca-profile rsa {
    ca-identity rsa;
    revocation-check {
        disable;
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
        ge-0/0/3.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
```

```
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Tunnels | **1161**
- Verifying Traffic Selectors | **1164**

Confirm that the configuration is working properly.

Verifying Tunnels

Purpose

Verify that tunnels are established between the AutoVPN hub and spoke.

Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the hub.

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
1350248074 UP  d195bce6ccfcf9af  8f1569c6592c8408  Main           10.2.2.253

user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <77594650 ESP:aes-cbc-192/sha1 ac97cb1 2799/  150000 - root 500 10.2.2.253
  >77594650 ESP:aes-cbc-192/sha1 828dc013 2798/  150000 - root 500 10.2.2.253

user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------

ID: 77594650 Virtual-system: root, VPN Name: HUB_VPN
  Local Gateway: 10.2.2.1, Remote Gateway: 10.2.2.253
  Traffic Selector Name: ts1
  Local Identity: ipv4(192.0.0.0-192.255.255.255)
  Remote Identity: ipv4(172.0.0.0-172.255.255.255)
  Version: IKEv1
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x24608b29
  Tunnel events:
    Tue Dec 30 2014 11:30:21 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Dec 30 2014 11:30:20 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Dec 30 2014 11:30:20 -0800: IKE SA negotiation successfully completed (3 times)
  Location: FPC 5, PIC 0, KMD-Instance 1
  Direction: inbound, SPI: ac97cb1, AUX-SPI: 0
    Hard lifetime: Expires in 2796 seconds
    Lifesize Remaining:  150000 kilobytes
    Soft lifetime: Expires in 2211 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
```

```
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Location: FPC 5, PIC 0, KMD-Instance 1
  Direction: outbound, SPI: 828dc013, AUX-SPI: 0
    Hard lifetime: Expires in 2796 seconds
    Lifesize Remaining:  150000 kilobytes
    Soft lifetime: Expires in 2211 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the spoke.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
276505646 UP   d195bce6ccfcf9af  8f1569c6592c8408  Main           10.2.2.1

user@host> show security ipsec security-associations
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <69206018 ESP:aes-cbc-192/sha1 828dc013 2993/  150000 - root 500 10.2.2.1
  >69206018 ESP:aes-cbc-192/sha1 ac97cb1 2993/  150000 - root 500 10.2.2.1

user@host> show security ipsec security-associations detail
ID: 69206018 Virtual-system: root, VPN Name: SPOKE_VPN
  Local Gateway: 10.2.2.253, Remote Gateway: 10.2.2.1
  Traffic Selector Name: ts1
  Local Identity: ipv4(172.0.0.0-172.255.255.255)
  Remote Identity: ipv4(192.0.0.0-192.255.255.255)
  Version: IKEv1
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x2c608b29
  Tunnel events:
    Tue Dec 30 2014 11:30:20 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Dec 30 2014 11:30:20 -0800: IKE SA negotiation successfully completed (1 times)
    Tue Dec 30 2014 11:26:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
  Location: FPC 1, PIC 0, KMD-Instance 1
  Direction: inbound, SPI: 828dc013, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  150000 kilobytes
```

```
    Soft lifetime: Expires in 2369 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Location: FPC 1, PIC 0, KMD-Instance 1
  Direction: outbound, SPI: ac97cb1, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  150000 kilobytes
    Soft lifetime: Expires in 2369 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. The hub shows one active tunnel to the spoke while the spoke shows one active tunnel to the hub.

If no SAs are listed for IKE Phase 1, then there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

If no SAs are listed for IKE Phase 2, then there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying Traffic Selectors

Purpose

Verify the traffic selectors.

Action

From operational mode, enter the `show security ipsec traffic-selector interface-name st0.1` command on the hub.

```
user@host> show security ipsec traffic-selector interface-name st0.1
node0:
--------------------------------------------------------------------------
```

```
 Source IP                        Destination IP                  Interface   Tunnel-id    IKE-
ID
 192.0.0.0-192.255.255.255          172.0.0.0-172.255.255.255          st0.1       77594650
DC=Domain_component, CN=Spoke1_ID, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
```

From operational mode, enter the `show security ipsec traffic-selector interface-name st0.1` command on the spoke.

```
user@host> show security ipsec traffic-selector interface-name st0.1
 Source IP                        Destination IP                  Interface   Tunnel-id    IKE-
ID
 172.0.0.0-172.255.255.255          192.0.0.0-192.255.255.255          st0.1       69206018
DC=Domain_component, CN=Hub_ID, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
```

Meaning

A traffic selector is an agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified pair of local and remote addresses. Only traffic that conforms to a traffic selector is permitted through an SA. Traffic selectors are negotiated between the initiator and the responder (the hub).

Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors

IN THIS SECTION

- Requirements | **1166**
- Overview | **1166**
- Configuration | **1169**
- Verification | **1190**

Georedundancy is the deployment of multiple geographically distant sites so that traffic can continue to flow over a provider network even if there is a power outage, a natural disaster, or other catastrophic event that affects a site. In a mobile provider network, multiple Evolved Node B (eNodeB) devices can be

connected to the core network through georedundant IPsec VPN gateways on the firewalls. The alternate routes to the eNodeB devices are distributed to the core network using a dynamic routing protocol.

This example configures AutoVPN hubs with multiple traffic selectors on the firewalls to ensure that there are georedundant IPsec VPN gateways to eNodeB devices. Auto route insertion (ARI) is used to automatically insert routes toward the eNodeB devices in the routing tables on the hubs. ARI routes are then distributed to the provider's core network through BGP. The example is using the certificate based authentication. For authentication with preshared key, set up a similar configuration shown at "Example: Configuring Basic AutoVPN with iBGP" on page 929.

Requirements

This example uses the following hardware and software components:

- Two firewalls connected and configured in a chassis cluster. The chassis cluster is AutoVPN hub A.
- One firewall configured as AutoVPN hub B.
- Supported Junos OS Release.
- eNodeB devices that can establish IPsec VPN tunnels with AutoVPN hubs. eNodeB devices are third-party network equipment providers that initiate a VPN tunnel with AutoVPN hubs.
- Digital certificates enrolled in the hubs and the eNodeB devices that allow the devices to authenticate each other.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.
- Enroll the digital certificates in each device. See Enroll Certificate.

This example uses the BGP dynamic routing protocol to advertise routes toward the eNodeB devices to the core network.

Overview

IN THIS SECTION

- Topology | **1168**

In this example, two AutoVPN hubs are configured with multiple traffic selectors on the firewalls to provide georedundant IPsec VPN gateways to eNodeB devices. ARI automatically inserts routes to the eNodeB devices in the routing tables on the hubs. ARI routes are then distributed to the provider's core network through BGP.

Certain Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hubs and eNodeB devices must have the same values. Table 117 on page 1167 shows the values used in this example:

Table 117: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs

Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5
Authentication algorithm	sha-1
Encryption algorithm	aes-256-cbc
IKE policy:	
Certificate	local-certificate
IKE gateway:	
Dynamic	distinguished name wildcard DC=Common_component
IKE user type	group IKE id
Dead peer detection	probe-idle-tunnel
Local identity	distinguished name

Table 117: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs *(Continued)*

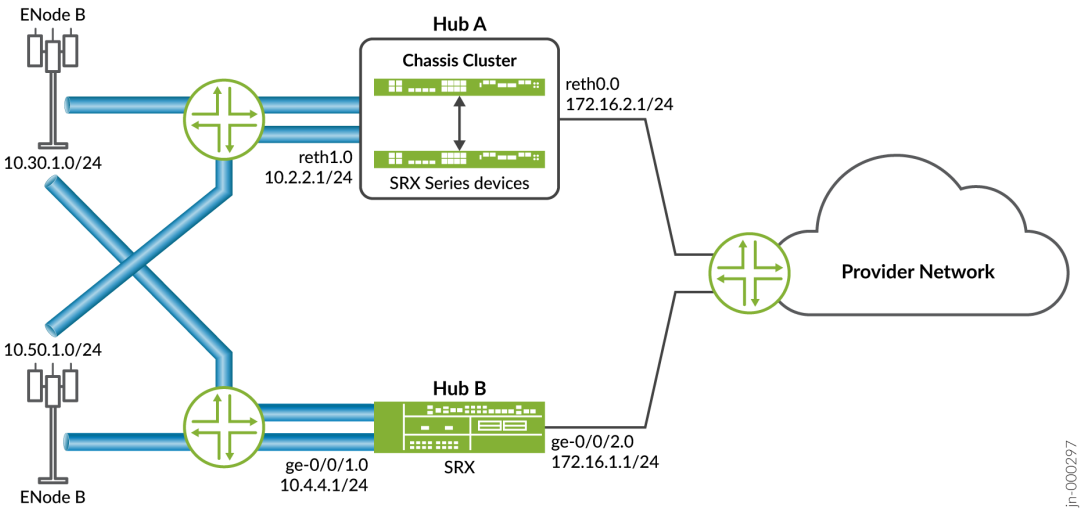
Option	Value
Version	v2-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-256-cbc
IPsec policy:	
Perfect Forward Secrecy (PFS) group	group5

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. For simplicity, the configuration on the firewalls allows all types of inbound traffic; this configuration is not recommended for production deployments.

Topology

Figure 64 on page 1169 shows the firewalls to be configured for this example.

Figure 64: Georedundant IPsec VPN Gateways to eNodeB Devices



Configuration

IN THIS SECTION

- Configuring Hub A | **1169**
- Configuring Hub B | **1179**
- Configuring the eNodeB (Sample Configuration) | **1188**

Configuring Hub A

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-8/0/2 gigether-options redundant-parent reth1
```

```
set interfaces ge-8/0/3 gigether-options redundant-parent reth0
set interfaces lo0 unit 0 family inet address 10.100.1.100/24
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 172.16.2.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.2.1/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
set security ike policy ph1_ike_policy proposals prop_ike
set security ike policy ph1_ike_policy certificate local-certificate HubA_certificate
set security ike gateway HUB_GW ike-policy ph1_ike_policy
set security ike gateway HUB_GW dynamic distinguished-name wildcard DC=Common_component
set security ike gateway HUB_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_GW dead-peer-detection probe-idle-tunnel
set security ike gateway HUB_GW local-identity distinguished-name
set security ike gateway HUB_GW external-interface reth1
set security ike gateway HUB_GW version v2-only
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-256-cbc
set security ipsec policy ph2_ipsec_policy perfect-forward-secrecy keys group5
set security ipsec policy ph2_ipsec_policy proposals prop_ipsec
set security ipsec vpn HUB_VPN bind-interface st0.1
set security ipsec vpn HUB_VPN ike gateway HUB_GW
set security ipsec vpn HUB_VPN ike ipsec-policy ph2_ipsec_policy
set security ipsec vpn HUB_VPN traffic-selector ts1 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 10.50.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 remote-ip 10.30.0.0/16
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 172.16.2.1
set protocols bgp group internal-peers export inject_ts1_routes
set protocols bgp group internal-peers export inject_ts2_routes
set protocols bgp group internal-peers export inject_up_routes
set protocols bgp group internal-peers neighbor 172.16.2.4
set policy-options policy-statement inject_ts1_routes term cp_allow from protocol static
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
10.30.1.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
```

```
10.30.1.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow then next-hop self
set policy-options policy-statement inject_ts1_routes term cp_allow then accept
set policy-options policy-statement inject_ts2_routes term mp_allow from protocol static
set policy-options policy-statement inject_ts2_routes term mp_allow from route-filter
10.50.1.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow from route-filter
10.50.2.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow then next-hop self
set policy-options policy-statement inject_ts2_routes term mp_net_allow then accept
set policy-options policy-statement inject_up_routes term up_allow from protocol static
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.1.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.2.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow then next-hop self
set policy-options policy-statement inject_up_routes term up_allow then accept
set security pki ca-profile csa ca-identity csa
set security pki ca-profile csa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure hub A:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/2 gigether-options redundant-parent reth1
user@host# set ge-0/0/3 gigether-options redundant-parent reth0
user@host# set ge-8/0/2 gigether-options redundant-parent reth1
```

```
user@host# set ge-8/0/3 gigether-options redundant-parent reth0
user@host# set lo0 unit 0 family inet address 10.100.1.100/24
user@host# set lo0 redundant-pseudo-interface-options redundancy-group 1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 172.16.2.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 10.2.2.1/24
user@host# set st0 unit 1 family inet
```

2. Configure Phase 1 options.

```
[edit security ike proposal prop_ike]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ph1_ike_policy]
user@host# set proposals prop_ike
user@host# set certificate local-certificate HubA_certificate
[edit security ike gateway HUB_GW]
user@host# set ike-policy ph1_ike_policy
user@host# set dynamic distinguished-name wildcard DC=Common_component
user@host# set dynamic ike-user-type group-ike-id
user@host# set dead-peer-detection probe-idle-tunnel
user@host# set local-identity distinguished-name
user@host# set external-interface reth1
user@host# set version v2-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy ph2_ipsec_policy]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec vpn HUB_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway HUB_GW
user@host# set ike ipsec-policy ph2_ipsec_policy
```

```
user@host# set traffic-selector ts1 local-ip 172.16.0.0/16
user@host# set traffic-selector ts1 remote-ip 10.50.0.0/16
user@host# set traffic-selector ts2 local-ip 172.16.0.0/16
user@host# set traffic-selector ts2 remote-ip 10.30.0.0/16
```

4. Configure the BGP routing protocol.

```
[edit protocols bgp group internal-peers]
user@host# set type internal
user@host# set local-address 172.16.2.1
user@host# set export inject_ts1_routes
user@host# set export inject_ts2_routes
user@host# set export inject_up_routes
user@host# set neighbor 172.16.2.4
```

5. Configure routing options.

```
[edit policy-options policy-statement inject_ts1_routes]
user@host# set term cp_allow from protocol  static
user@host# set term cp_allow from route-filter 10.30.2.0/24 orlonger
user@host# set term cp_allow from route-filter 10.30.1.0/24 orlonger
user@host# set term cp_allow then next-hop  self
user@host# set term cp_allow then accept
[edit policy-options policy-statement inject_ts2_routes]
user@host# set term mp_allow from protocol  static
user@host# set term mp_allow from route-filter 10.50.1.0/24 orlonger
user@host# set term mp_allow from  route-filter 10.50.2.0/24 orlonger
user@host# set term mp_allow then next-hop  self
user@host# set term mp_allow then accept
[edit policy-options policy-statement inject_up_routes]
user@host# set term up_allow from protocol  static
user@host# set term up_allow from route-filter 172.16.1.0/24 orlonger
user@host# set term up_allow from  route-filter 172.16.2.0/24 orlonger
user@host# set term up_allow then next-hop  self
user@host# set term up_allow then accept
```

6. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile csa ca-identity csa
user@host# set ca-profile csa revocation-check disable
```

7. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces lo0.0
user@host# set interfaces reth1.0
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` `show security ike`, `show security ipsec`, `show protocols bgp`, `show policy-options`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/2 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-0/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
```

```
    }
    ge-8/0/2 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-8/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.100.1.100/24;
            }
        }
        redundant-pseudo-interface-options {
            redundancy-group 1;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 172.16.2.1/16;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.2.2.1/24;
            }
        }
    }
    st0 {
        unit 1 {
```

```
            family inet;
        }
    }
[edit]
user@host# show security ike
    proposal prop_ike {
        authentication-method rsa-signatures;
        dh-group group5;
        authentication-algorithm sha1;
        encryption-algorithm aes-256-cbc;
    }
    policy ph1_ike_policy {
        proposals prop_ike;
        certificate {
            local-certificate HubA_certificate;
        }
    }
    gateway HUB_GW {
        ike-policy ph1_ike_policy;
        dynamic {
            distinguished-name {
                wildcard DC=Common_component;
            }
            ike-user-type group-ike-id;
        }
        dead-peer-detection {
            probe-idle-tunnel;
        }
        local-identity distinguished-name;
        external-interface reth1;
        version v2-only;
    }
[edit]
user@host# show security ipsec
    proposal prop_ipsec {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm aes-256-cbc;
    }
    policy ph2_ipsec_policy {
        perfect-forward-secrecy {
            keys group5;
        }
```

```
        proposals prop_ipsec;
    }
    vpn HUB_VPN {
        bind-interface st0.1;
        ike {
            gateway HUB_GW;
            ipsec-policy ph2_ipsec_policy;
        }
        traffic-selector ts1 {
            local-ip 172.16.0.0/16;
            remote-ip 10.50.0.0/16;
        }
        traffic-selector ts2 {
            local-ip 172.16.0.0/16;
            remote-ip 10.30.0.0/16;
        }
    }
[edit]
user@host# show protocols bgp
    group internal-peers {
        type internal;
        local-address 172.16.2.1;
            export [ inject_ts1_routes inject_ts2_routes inject_up_routes ];
        neighbor 172.16.2.4;
    }
[edit]
user@host# show policy-options
policy-statement inject_ts1_routes {
    term cp_allow {
        from {
            protocol static;
            route-filter 10.30.2.0/24 orlonger;
            route-filter 10.30.1.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_ts2_routes {
    term mp_allow {
        from {
```

```
            protocol static;
            route-filter 10.50.1.0/24 orlonger;
            route-filter 10.50.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_up_routes {
    term up_allow {
        from {
            protocol static;
            route-filter 172.16.1.0/24 orlonger;
            route-filter 172.16.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
[edit]
user@host# show security pki
ca-profile csa {
    ca-identity csa;
    revocation-check {
        disable;
    }
}
[edit]
user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
```

```
            st0.1;
            reth0.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            lo0.0;
            reth1.0;
        }
    }
[edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Hub B

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.4.4.1/24
set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/16
set interfaces lo0 unit 0 family inet address 10.100.1.101/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
```

```
set security ike policy ph1_ike_policy proposals prop_ike
set security ike policy ph1_ike_policy certificate local-certificate HubB_certificate
set security ike gateway HUB_GW ike-policy ph1_ike_policy
set security ike gateway HUB_GW dynamic distinguished-name wildcard DC=Common_component
set security ike gateway HUB_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_GW dead-peer-detection probe-idle-tunnel
set security ike gateway HUB_GW local-identity distinguished-name
set security ike gateway HUB_GW external-interface ge-0/0/1
set security ike gateway HUB_GW version v2-only
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-256-cbc
set security ipsec policy ph2_ipsec_policy perfect-forward-secrecy keys group5
set security ipsec policy ph2_ipsec_policy proposals prop_ipsec
set security ipsec vpn HUB_VPN bind-interface st0.1
set security ipsec vpn HUB_VPN ike gateway HUB_GW
set security ipsec vpn HUB_VPN ike ipsec-policy ph2_ipsec_policy
set security ipsec vpn HUB_VPN traffic-selector ts1 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 10.50.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 remote-ip 10.30.0.0/8
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 172.16.1.1
set protocols bgp group internal-peers export inject_ts1_routes
set protocols bgp group internal-peers export inject_ts2_routes
set protocols bgp group internal-peers export inject_up_routes
set policy-options policy-statement inject_ts1_routes term cp_allow from protocol static
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
10.30.2.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
10.30.1.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow then next-hop self
set policy-options policy-statement inject_ts1_routes term cp_allow then accept
set policy-options policy-statement inject_ts2_routes term mp_allow from protocol static
set policy-options policy-statement inject_ts2_routes term mp_allow from route-filter
10.50.1.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow from route-filter
10.50.2.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow then next-hop self
set policy-options policy-statement inject_ts2_routes term mp_net_allow then accept
set policy-options policy-statement inject_up_routes term up_allow from protocol static
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.1.0/24 orlonger
```

```
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.2.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow then next-hop self
set policy-options policy-statement inject_up_routes term up_allow then accept
set security pki ca-profile csa ca-identity csa
set security pki ca-profile csa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces ge-0/0/2.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure hub B:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.4.4.1/24
user@host# set ge-0/0/2 unit 0 family inet address 172.16.1.1/16
user@host# set lo0 unit 0 family inet address 10.100.1.101/24
user@host# set st0 unit 1 family inet
```

2. Configure Phase 1 options.

```
[edit security ike proposal prop_ike]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ph1_ike_policy]
user@host# set proposals prop_ike
user@host# set certificate local-certificate HubB_certificate
```

```
[edit security ike gateway HUB_GW]
user@host# set ike-policy ph1_ike_policy
user@host# set dynamic distinguished-name wildcard DC=Common_component
user@host# set dynamic ike-user-type group-ike-id
user@host# set dead-peer-detection probe-idle-tunnel
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1
user@host# set version v2-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy ph2_ipsec_policy]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec vpn HUB_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway HUB_GW
user@host# set ike ipsec-policy ph2_ipsec_policy
user@host# set traffic-selector ts1 local-ip 172.16.0.0/16
user@host# set traffic-selector ts1 remote-ip 10.50.0.0/16
user@host# set traffic-selector ts2 local-ip 172.16.0.0/16
user@host# set traffic-selector ts2 remote-ip 10.30.0.0/16
```

4. Configure the BGP routing protocol.

```
[edit protocols bgp group internal-peers]
user@host# set type internal
user@host# set local-address 172.16.1.1
user@host# set export inject_ts1_routes
user@host# set export inject_ts2_routes
user@host# set export inject_up_routes
user@host# set neighbor 172.16.1.2
```

5. Configure routing options.

```
[edit policy-options policy-statement inject_ts1_routes]
user@host# set term cp_allow from protocol  static
user@host# set term cp_allow from route-filter 10.30.2.0/24 orlonger
user@host# set term cp_allow from route-filter 10.30.1.0/24 orlonger
user@host# set term cp_allow then next-hop  self
user@host# set term cp_allow then accept
[edit policy-options policy-statement inject_ts2_routes]
user@host# set term mp_allow from protocol  static
user@host# set term mp_allow from route-filter 10.50.1.0/24 orlonger
user@host# set term mp_allow from route-filter 10.50.2.0/24 orlonger
user@host# set term mp_allow then next-hop  self
user@host# set term mp_allow then accept
[edit policy-options policy-statement inject_up_routes]
user@host# set term up_allow from protocol  static
user@host# set term up_allow from route-filter 172.16.1.0/24 orlonger
user@host# set term up_allow from  route-filter 172.16.2.0/24 orlonger
user@host# set term up_allow then next-hop  self
user@host# set term up_allow then accept
```

6. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile csa ca-identity csa
user@host# set ca-profile csa revocation-check disable
```

7. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/2.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces lo0.0
user@host# set interfaces ge-0/0/1.0
```

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` `show security ike`, `show security ipsec`, `show protocols bgp`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.4.4.1/24;
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 172.16.1.1/16;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.100.1.101/24;
            }
        }
    }
    st0 {
        unit 1 {
            family inet;
        }
    }
[edit]
user@host# show security ike
    proposal prop_ike {
```

```
        authentication-method rsa-signatures;
        dh-group group5;
        authentication-algorithm sha1;
        encryption-algorithm aes-256-cbc;
    }
    policy ph1_ike_policy {
        proposals prop_ike;
        certificate {
            local-certificate HubB_certificate;
        }
    }
    gateway HUB_GW {
        ike-policy ph1_ike_policy;
        dynamic {
            distinguished-name {
                wildcard DC=Common_component;
            }
            ike-user-type group-ike-id;
        }
        dead-peer-detection {
            probe-idle-tunnel;
        }
        local-identity distinguished-name;
        external-interface reth1;
        version v2-only;
    }
[edit]
user@host# show security ipsec
    proposal prop_ipsec {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm aes-256-cbc;
    }
    policy ph2_ipsec_policy {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals prop_ipsec;
    }
    vpn HUB_VPN {
        bind-interface st0.1;
        ike {
            gateway HUB_GW;
```

```
            ipsec-policy ph2_ipsec_policy;
        }
        traffic-selector ts1 {
            local-ip 172.16.0.0/16;
            remote-ip 10.50.0.0/16;
        }
        traffic-selector ts2 {
            local-ip 172.16.0.0/16;
            remote-ip 10.30.0.0/16;
        }
    }
[edit]
user@host# show protocols bgp
    group internal-peers {
        type internal;
        local-address 172.16.1.1;
            export [ inject_ts1_routes inject_ts2_routes inject_up_routes ];
        neighbor 172.16.1.2;
    }
user@host# show policy-options
policy-statement inject_ts1_routes {
    term cp_allow {
        from {
            protocol static;
            route-filter 10.30.2.0/24 orlonger;
            route-filter 10.30.1.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_ts2_routes {
    term mp_allow {
        from {
            protocol static;
            route-filter 10.50.1.0/24 orlonger;
            route-filter 10.50.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
```

```
        }
    }
}
policy-statement inject_up_routes {
    term up_allow {
        from {
            protocol static;
            route-filter 172.16.1.0/24 orlonger;
            route-filter 172.16.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
[edit]
user@host# show security pki
ca-profile csa {
    ca-identity csa;
    revocation-check {
        disable;
    }
}
[edit]
user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            st0.1;
            ge-0/0/2.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
```

```
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/1.0;
            lo0.0;
        }
    }
[edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the eNodeB (Sample Configuration)

Step-by-Step Procedure

1. The eNodeB configuration in this example is provided for reference. Detailed eNodeB configuration information is beyond the scope of this document. The eNodeB configuration must include the following information:

   - Local certificate (X.509v3) and IKE identity information
   - The firewall's IKE identity information and public IP address
   - Phase 1 and Phase 2 proposals that match the configurations on the hubs

Results

The eNodeB devices in this example use strongSwan open source software for IPsec-based VPN connections:

```
config setup
        plutostart=yes
        plutodebug=all
        charondebug="ike 4, cfg 4, chd 4, enc 1"
```

```
        charonstart=yes  #ikev2 deamon"
        nat_traversal=yes  #<======= need to enable even no nat_t

conn %default
        ikelifetime=60m
        keylife=45m
        rekeymargin=2m
        keyingtries=4
        mobike=no

conn Hub_A
        keyexchange=ikev2
        authby=pubkey
        ike=aes256-sha-modp1536
        esp=aes256-sha1-modp1536
        leftcert=/usr/local/etc/ipsec.d/certs/fight02Req.pem.Email.crt
        left=10.5.5.1 # self if
        leftsubnet=10.1.1.0/24 # left subnet
        leftid="CN=fight02, DC=Common_component, OU=Dept, O=Company, L=City, ST=CA, C=US " #
self id
        right=10.2.2.1 # peer if
        rightsubnet=10.1.1.0/24 # peer net for proxy id
        rightid="DC=Domain_component, CN=HubA_certificate, OU=Dept, O=Company, L=City, ST=CA,
C=US " # peer id
        auto=add
        leftfirewall=yes
        dpdaction=restart
        dpddelay=10
        dpdtimeout=120
        rekeyfuzz=10%
        reauth=no

conn Hub_B
        keyexchange=ikev2
        authby=pubkey
        ike=aes256-sha-modp1536
        esp=aes192-sha1-modp1536
        leftcert=/usr/local/etc/ipsec.d/certs/fight02Req.pem.Email.crt
        left=10.5.5.1 # self if
        leftsubnet=10.1.1.0/24 # self net for proxy id
        leftid="CN=fight02, DC=Common_component, OU=Dept, O=Company, L=City, ST=CA, C=US " #
self id
        right=10.4.4.1 # peer if
```

```
        rightsubnet=10.1.1.0/24 # peer net for proxy id
        rightid="DC=Domain_component, CN=HubB_certificate, OU=Dept, O=Company, L=City, ST=CA,
C=US " # peer id
        auto=add
        leftfirewall=yes
        dpdaction=restart
        dpddelay=10
        dpdtimeout=120
        rekeyfuzz=10%
        reauth=no
```

Verification

IN THIS SECTION

- Verifying Tunnels on the AutoVPN Hubs | **1190**
- Verifying Traffic Selectors | **1191**
- Verifying ARI Routes | **1192**

Confirm that the configuration is working properly.

Verifying Tunnels on the AutoVPN Hubs

Purpose

Verify that tunnels are established between the AutoVPN hub and eNodeB devices.

Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the hub.

```
user@host> show security ike security-associations
node0:
------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
276505706 UP   16d6e53f0866b5cc  ccd8ca944da7b63e  IKEv2          10.5.5.1
```

```
1350247532 UP  d5f0cb3a3b18cb92  91269f05527217a0  IKEv2          10.1.1.1

user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <77594626 ESP:aes-cbc-192/sha1 a82bbc3 3600/  64 - root 500  10.1.1.1
  >77594626 ESP:aes-cbc-192/sha1 c930a858 3600/  64 - root 500 10.1.1.1
  <69206018 ESP:aes-cbc-192/sha1 2b437fc 3600/  64 - root 500  10.5.5.1
  >69206018 ESP:aes-cbc-192/sha1 c6e02755 3600/  64 - root 500 10.5.5.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. The hub shows two active tunnels, one to each eNodeB device.

If no SAs are listed for IKE Phase 1, then there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and eNodeB devices.

If no SAs are listed for IKE Phase 2, then there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and eNodeB devices.

Verifying Traffic Selectors

Purpose

Verify the traffic selectors.

Action

From operational mode, enter the `show security ipsec traffic-selector interface-name st0.1` command.

```
user@host> show security ipsec traffic-selector interface-name st0.1
node0:
--------------------------------------------------------------------------
 Source IP                       Destination IP                  Interface   Tunnel-id    IKE-
ID
```

```
 10.1.1.0-10.1.1.255              10.1.1.0-10.1.1.255             st0.1       69206018
DC=Common_component, CN=enodebA, OU=Dept, O=Company, L=City, ST=CA, C=US
 10.1.1.0-10.1.1.255              10.1.1.0-10.1.1.255             st0.1       77594626
DC=Common_component, CN=enodebB, OU=Dept, O=Company, L=City, ST=CA, C=US
```

Meaning

A traffic selector is an agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified pair of local and remote addresses. Only traffic that conforms to a traffic selector is permitted through an SA. Traffic selectors are negotiated between the initiator and the responder (the hub).

Verifying ARI Routes

Purpose

Verify that the ARI routes are added to the routing table.

Action

From operational mode, enter the `show route` command.

```
user@host> show route
inet.0: 23 destinations, 23 routes (22 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.0.0/16         *[Static/5] 02:57:57
                    > to 2.2.2.253 via reth1.0
10.2.2.0/24         *[Direct/0] 02:58:43
                    > via reth1.0
10.2.2.1/32         *[Local/0] 02:59:25
                      Local via reth1.0
10.5.0.0/16         *[Static/5] 02:57:57
                    > to 2.2.2.253 via reth1.0
10.157.64.0/19     *[Direct/0] 21:54:52
                    > via fxp0.0
10.157.75.117/32   *[Local/0] 21:54:52
                      Local via fxp0.0
10.254.75.117/32   *[Direct/0] 21:54:52
                    > via lo0.0
10.30.1.0/24        *[ARI-TS/5] 02:28:10        [ARI route added based on TSi]
```

```
                    > via st0.1
10.50.1.0/24        *[ARI-TS/5] 02:28:26
                    > via st0.1
10.80.0.0/16         *[Direct/0] 02:57:57
                    > via reth0.0
10.80.1.1/32        *[Local/0] 02:57:57
                      Local via reth0.0
10.100.1.0/24     *[Direct/0] 02:57:57
                    > via lo0.0
10.100.1.100/32   *[Local/0] 02:57:57
                       Local via lo0.0
10.102.1.0/24     *[Static/5] 02:57:57
                    > to 10.2.2.253 via reth1.0
10.104.1.0/24     *[Static/5] 02:57:57
                    > to 10.2.2.253 via reth1.0
172.16.0.0/12      *[Static/5] 21:54:52
```

Meaning

Auto route insertion (ARI) automatically inserts a static route for the remote network and hosts protected by a remote tunnel endpoint. A route is created based on the remote IP address configured in the traffic selector. In the case of traffic selectors, the configured remote address is inserted as a route in the routing instance associated with the st0 interface that is bound to the VPN.

Static routes to the eNodeB destinations 10.30.1.0/24 and 10.50.1.0/24 are added to the routing table on the hub. These routes are reachable through the st0.1 interface.

Example: Configuring AutoVPN with Pre-Shared Key

IN THIS SECTION

- Configure different IKE preshared key | **1194**
- Configure same IKE preshared key | **1196**

This example shows how to configure different IKE preshared key used by the VPN gateway to authenticate the remote peer. Similarly, to configure same IKE preshared key used by the VPN gateway to authenticate the remote peer.

Refer other examples in this topic for end-to-end configuration of AutoVPN.

Configure different IKE preshared key

To configure different IKE preshared key that the VPN gateway uses to authenticate the remote peer, perform these tasks.

1. Configure the seeded preshared for IKE policy in the device with AutoVPN hub.

```
 [edit]
user@host# set security ike policy IKE_POL seeded-pre-shared-key ascii-text ascii-text
```

or

```
user@host# set security ike policy IKE_POL seeded-pre-shared-key hexadecimal hexadecimal
```

For example:

```
user@host# set security ike policy IKE_POL seeded-pre-shared-key ascii-text
ThisIsMySecretPreSharedkey
```

or

```
user@host# set security ike policy IKE_POL seeded-pre-shared-key hexadecimal
5468697349734d795365637265637450726553686172656 46b6579
```

2. Display the `pre-shared key` for remote peer using gateway name and user-id.

```
 [edit]
user@host> show security ike pre-shared-key gateway gateway-name user-id user-id
```

For example:

```
user@host> show security ike pre-shared-key gateway-name HUB_GW user-id user1@juniper.net
```

```
Pre-shared key: 79e4ea39f5c06834a3c4c031e37c6de24d46798a
```

3. Configure the generated PSK ("79e4ea39f5c06834a3c4c031e37c6de24d46798a" in "step 2" on page 1194) in the ike policy on the remote peer device.

```
[edit]
user@peer# set security ike policy IKE_POL pre-shared-key ascii-text generated-psk
```

For example:

```
user@peer# set security ike policy IKE_POL pre-shared-key ascii-text
79e4ea39f5c06834a3c4c031e37c6de24d46798a
```

4. (Optional) To bypass the IKE ID validation and allow all IKE ID types, configure `general-ikeid` configuration statement under the [edit security ike gateway *gateway_name* dynamic] hierarchy level in the gateway.

```
[edit]
user@host# set security ike gateway HUB_GW dynamic general-ikeid
```

Result

From the configuration mode, confirm your configuration by entering the show security command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host> show security
    ike {
        proposal IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 750;
        }
        policy IKE_POL {
          proposals IKE_PROP;
          seeded-pre-shared-key ascii-text "$9$zoDln9pIEyWLN0BLNdboaFn/C0BRhSeM8"; ##SECRET-DATA
        }
```

```
        gateway HUB_GW {
            ike-policy IKE_POL;
            dynamic {
                general-ikeid;
                ike-user-type group-ike-id;
            }
            local-identity hostname hub.juniper.net;
            external-interface lo0.0;
            local-address 11.0.0.1;
            version v2-only;
        }
    }
```

Configure same IKE preshared key

To configure same IKE preshared key that the VPN gateway uses to authenticate the remote peer, perform these tasks.

1. Configure the common `pre-shared-key` for ike policy in the device with AutoVPN hub.

```
 [edit]
user@host# set security ike policy IKE_POL pre-shared-key ascii-text ascii text
```

For example:

```
user@host# # set security ike policy IKE_POL pre-shared-key ascii-text
ThisIsMySecretPreSharedkey
```

2. Configure the common `pre-shared-key` on the ike policy for remote peer device.

```
 [edit]
user@peer# set security ike policy IKE_POL pre-shared-key ascii-text ascii text
```

For example:

```
user@peer# set security ike policy IKE_POL pre-shared-key ascii-text
ThisIsMySecretPreSharedkey
```

3. (Optional) To bypass the IKE ID validation and allow all IKE ID types, configure `general-ikeid` configuration statement under the [edit security ike gateway *gateway_name* dynamic] hierarchy level in the gateway.

```
[edit]
user@host# set security ike gateway HUB_GW dynamic general-ikeid
```

Result

From the configuration mode, confirm your configuration by entering the show security command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host> show security
    ike {
        proposal IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 750;
        }
        policy IKE_POL {
            proposals IKE_PROP;
            pre-shared-key ascii-text "$9$wo2oGk.569pDi9p0BSys24"; ## SECRET-DATA
        }
        gateway HUB_GW {
            ike-policy IKE_POL;
            dynamic {
                general-ikeid;
                ike-user-type group-ike-id;
            }
            local-identity user-at-hostname user1@juniper.net;
            external-interface lo0;
            local-address 11.0.0.1;
            version v2-only;
        }
    }
```

Configure Multicast Support on P2MP Infrastructure

In this topic, you'll learn how to enable multicast support on P2MP infrastructure.

Before enabling multicast support, ensure that you meet the considerations listed in "Multicast Support Using PIM" on page 920.

See the following sections to configure and verify multicast support.

Configure Multicast Interface

- To enable PIM on the st0.0 interface, use the `set protocols pim interface` *`interface-name`* `command`:

```
[edit]
user@host# set protocols pim interface st0.0
```

  Here, `st0.0` is the secure tunnel interface.

- To enable multipoint on the st0.0 interface for P2MP mode use `set interfaces` *`interface-name`* `unit` *`unit-number`* `multipoint` command:

```
[edit]
user@host# set interfaces st0.0 unit 0 multipoint
```

- To set the IPv4 address for the st0.0 interface, use the `set interfaces` *`interface-name`* `unit` *`unit-number`* `family inet address` *`IPv4 address`* command:

```
[edit]
user@host# set interfaces st0.0 unit 0 family inet address 192.168.1.3/24
```

  Here, 192.168.1.3/24 is the IP address of the interface.

- To disable PIM on the st0.0 interface, use the option `disable`:

```
[edit]
user@host# set protocols pim interface st0.0 disable
```

CLI Commands to Verify the Multicast Configuration

You can verify multicast configuration using the following commands.

- To list the PIM interfaces, use the `show pim interfaces` command.
- To list the neighbors that joined the multicast groups, use the `show pim join extensive` command.
- To view the entries in the IP multicast forwarding table, use the `show multicast route` command.
- To view the multicast next hop details, use the `show multicast next-hops detail` command.
- To view the IP multicast statistics, use the `show multicast statistics` command.
- To view the forwarding table entries, use the `show route forwarding-table extensive` command.

SEE ALSO

PIM Overview

interface (Protocols PIM)

show pim interfaces

show pim join

show multicast route

show multicast next-hops

show multicast statistics

Platform-Specific AutoVPN Behavior

IN THIS SECTION

- Platform-Specific Authentication Behavior | **1200**
- Platform-Specific Multicast Behavior | **1200**

Use Feature Explorer to confirm platform and release support for specific features.

Use the following tables to review platform-specific behaviors for your platform.

Platform-Specific Authentication Behavior

Table 118: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX5000 line with SPC3 and vSRX Virtual Firewalls that support PSK-based authentication, you must run IPsec VPN service with the iked process for AutoVPN with seeded PSK authentication. The firewall supports AutoVPN with PSK only if you install the junos-ike package.

Platform-Specific Multicast Behavior

Table 119: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX300, SRX320, SRX340, SRX345, SRX1500, and vSRX 3.0 devices that support multicast, to enable PIM on st0 `p2mp` interface, you must run IPsec VPN service with the kmd process. • On SRX1500, SRX1600, SRX2300, SRX4100, SRX4120, SRX4200, SRX4300, SRX4600, SRX4700, and vSRX 3.0 devices that support multicast, you can enable PIM on st0 `p2mp` interface when the IPsec VPN service runs with the iked process. • SRX400, SRX440, and SRX440-2AC do not support multicast.

Additional Platform Information

Use Feature Explorer to confirm platform and release support for specific features. Additional Platforms may be supported.

Table 120: AutoVPN PSK Support with IKED Modes

IKE Mode	**SRX5000 Line with SPC3 vSRX 3.0 (iked process)**	
	Shared PSK	Seeded-PSK
IKEv2	Yes	Yes
IKEv2 with any-remote-id	Yes	Yes
IKEv1 Aggressive Mode	Yes	Yes
IKEv1 Aggressive Mode with any-remote-id/general-ikeid	Yes	Yes
IKEv1 main mode	Yes	No
IKEv1 main mode with any-remote-id/general-ikeid	Yes	No

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.2R1	Support for multicast traffic (IPv4 address) with AutoVPN for firewalls running the iked process is added in Junos OS Release 24.2R1.
19.2R1	Support for multicast traffic with AutoVPN for firewalls running the kmd process is added in Junos OS Release 19.2R1.
17.4R1	Starting with Junos OS Release 17.4R1, IPv6 address is supported on AutoVPN.
17.4R1	Starting with Junos OS Release 17.4R1, AutoVPN networks that use secure tunnel interfaces in point-to-point mode support IPv6 addresses for traffic selectors and for IKE peers.

15.1X49-D120	Starting with Junos OS Release 15.1X49-D120, you can configure the CLI option `reject-duplicate-connection` at the [`edit security ike gateway` *`gateway-name`* `dynamic`] hierarchy level to retain an existing tunnel session and reject negotiation requests for a new tunnel with the same IKE ID.

RELATED DOCUMENTATION

- VPN Monitoring Overview

10

CHAPTER

Remote Access VPN

IN THIS CHAPTER

- Juniper Secure Connect | **1204**

Juniper Secure Connect

SUMMARY

Read this topic to get an overview about Juniper Secure Connect solution.

IN THIS SECTION

- Benefits of Juniper Secure Connect | **1206**
- What's Next | **1206**

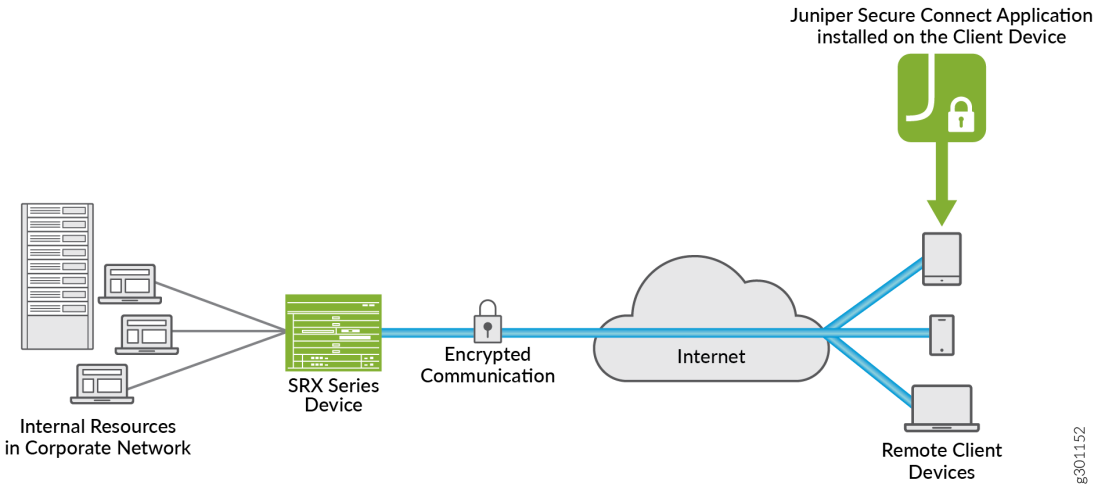
Juniper Secure Connect is a client-based SSL-VPN application that allows you to securely connect and access protected resources on your network. This application when combined with the firewall helps organizations quickly achieve dynamic, flexible, and adaptable connectivity from devices anywhere across the globe. Juniper Secure Connect extends visibility and enforcement from client to cloud using secure VPN connections.

Juniper Secure Connect application includes:

- Firewall—Serves as an entry and exit point for communication between users with Juniper Secure Connect and the protected resources on the corporate network or in cloud.
- Juniper Secure Connect application—Secures connectivity between the host clients running Microsoft Windows, Apple macOS, Google Android, and iOS operating systems and the protected resources. Juniper Secure Connect application connects through a VPN tunnel to the firewall to gain access to the protected resources in the network.

Figure 65 on page 1205 illustrates the Juniper Secure Connect remote access solution for establishing secure VPN connectivity for remote users at different locations.

Figure 65: Juniper Secure Connect Remote Access Solution



To work with Juniper Secure Connect, see System Requirements.

Table 121: Features Support for Juniper Secure Connect

Feature	Description
Multi-Platform support	Supports Windows, macOS, Android, and iOS platforms.
Windows Pre-domain logon	Allows users to logon to the local Windows system through an already established VPN tunnel (using Windows Pre-Logon), so that it is authenticated to the central Windows domain or Active Directory.
Configuration support	Validates automatically that the most current policy is available before establishing the connection.
Biometric user authentication	Allows the user to protect their credentials using the operating system's built-in biometric authentication support.
Multi-Factor Authentication (MFA)	Allows you to use multi-factor authentication to extend the authentication.
Juniper Secure Connect license	Licenses are available in 1 year and 3 year subscription models.

Benefits of Juniper Secure Connect

- Secure remote access from anywhere with VPN
- Simple user experience
- Easy management of remote clients, policies, and VPN events from a single console (using J-Web)

What's Next

- To configure Juniper Secure Connect, see Juniper Secure Connect User Guide.
- See the CLI configuration statements related to Juniper Secure Connect at:

  default-profile, *windows-logon*, *certificate*, traceoptions, *profile*, *global-options*, *client-config*, and *remote-access*.



NOTE: The Junos-FIPS devices do not support `web-management` statement at [`edit system services`] hierarchy level. For detailed list of Junos-FIPS configuration restrictions on the FIPS compliant firewalls, see platform specific Junos-FIPS configuration restrictions on the Juniper Tech Library. Search for the specific firewall and navigate to **System Admin Guides** > **FIPS Evaluated Configuration Guide**.

RELATED DOCUMENTATION

Overview

Migrating from Junos OS Dynamic VPN to Juniper Secure Connect

Preparing Juniper Secure Connect Configuration

11

CHAPTER

Monitoring VPN

IN THIS CHAPTER

- VPN Monitoring Overview | **1208**
- VPN Monitoring Methods | **1209**
- VPN Alarms, Audits, and Events | **1219**

VPN Monitoring Overview

SUMMARY

Read this topic to know why it's important to monitor VPNs and learn about what Junos OS offers to monitor your VPNs.

IN THIS SECTION

- Ways of Monitoring a VPN | **1208**

VPN monitoring is an important feature in terms of having an uninterrupted channel for secure communication. You monitor the VPN to ensure seamless functioning of all the elements involved in secure channel establishment—the security associations, the endpoints, the tunnel etc. focusing on tracking the overall health of the VPN.

Let's say, you have a VPN established between your firewalls. You typically assume that the VPN works seamlessly without any issues. However, that's not the case in a real scenario. You may encounter the following issues related to the VPN tunnel between the two firewalls:

- You stopped receiving traffic from the remote peer—How do you know whether there're no clients trying to use the VPN tunnel or whether another firewall in the data path is blocking the traffic?
- Your tunnel is successfully established (IKE phase 1 and phase 2 are complete), but the remote VPN endpoint becomes unreachable—Does the firewall detect this problem and ensure that the tunnel state is updated if the peer becomes unreachable?
- What if you know that the remote VPN endpoint is reachable, but you also want to verify that a specific host on the remote network is also reachable?
- What if suddenly your VPN peers become unsynchronized?

The VPN monitoring techniques discussed in this topic can detect these problems.

Ways of Monitoring a VPN

Junos OS offers multiple ways of monitoring a VPN. You can:

- Monitor the IPsec datapath before you configure the VPN tunnel.
- Enable the Dead Peer Detection (DPD) protocol for checking the availability of an IKE peer.
- Enable the VPN monitoring feature to check the liveness of a VPN tunnel.
- Check the Security Parameter Index (SPI) to uniquely identify the security association.

- Monitor VPN alarms and tunnel events.

RELATED DOCUMENTATION

VPN Monitoring Methods | 1209

VPN Alarms, Audits, and Events | 1219

VPN Monitoring Methods

SUMMARY

Read this topic to understand multiple ways in which you can monitor the VPN tunnel in your firewall.

IN THIS SECTION

- IPsec Datapath Verification | 1210
- Dead Peer Detection | 1212
- VPN Tunnel Monitoring | 1215
- Configure Dead Peer Detection | 1216
- Configure VPN Tunnel Monitoring | 1217
- Platform-Specific VPN Monitoring Behavior | 1219

We would expect the VPN tunnel to function optimally all the time. But that's hardly the case in a real world scenario. We know that the VPN tunnel can be down because of multiple reasons.

Junos OS offers following methods to monitor a VPN:

- IPsec datapath verification using Internet Control Message Protocol (ICMP) to check the datapath.
- Dead Peer Detection (DPD) protocol configuration to check the liveness of the IKE peer.
- VPN tunnel monitoring configuration to check the liveness of IPsec security association (SA).

Additionally, you can use the following global VPN features for monitoring:

- VPN peers in a SA can become unsynchronized when one of the peers doesn't respond. For example, if one of the peers reboots, it might send an incorrect security parameter index (SPI). You can enable the device to detect such an event and resynchronize the peers by configuring the bad SPI response feature. For more information about the `respond-bad-spi max-responses` option, see ike (Security).

- You can periodically send ICMP requests to the peer to determine whether the peer is reachable. For more information about the `vpn-monitor-options` option, see ipsec (Security).

You can choose to configure any of the methods explained in this topic to monitor your VPN.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific VPN Monitoring Behavior" on page 1219 section for notes related to your platform.

IPsec Datapath Verification

IN THIS SECTION

- Why Do You Need IPsec Datapath Verification? | **1210**
- How Does IPsec Datapath Verification Work? | **1211**

IPsec datapath verification is a process of validating the datapath between the tunnel endpoints to check that the path is clear and not blocked by any transit firewall.

Why Do You Need IPsec Datapath Verification?

The state of the secure tunnel (st0) interface in point-to-point mode for route-based VPNs is typically based on the state of the VPN tunnel. After the device establishes the IPsec security association (SA), the Junos OS adds routes associated with the st0 interface to the forwarding table. If your network has a transit firewall between the VPN tunnel endpoints, the firewall might block IPsec data traffic that uses active routes on the st0 interface. As a result, you might encounter traffic loss.

To avoid such traffic loss, you must enable IPsec datapath verification. When you enable this feature, the Junos OS device doesn't bring up the st0 interface until it verifies the datapath. You can configure the datapath verification with the following options:

- You can configure with the statement `[set security ipsec vpn vpn-name vpn-monitor verify-path]` for route-based and site-to-site VPN tunnels.
- If there is a NAT device in front of the peer tunnel endpoint, the firewall translates the IP address of the peer tunnel endpoint to the IP address of the NAT device. For the VPN monitor ICMP request to reach the peer tunnel endpoint, you need to explicitly specify the original, untranslated IP address of

the peer tunnel endpoint behind the NAT device. You can configure this with the `[set security ipsec vpn vpn-name vpn-monitor verify-path destination-ip]` configuration statement.

- You can configure the size of the packet that is used to verify an IPsec datapath before the st0 interface is brought up. Use the `[set security ipsec vpn vpn-name vpn-monitor verify-path packet-size]` configuration statement. The configurable packet size ranges from 64 to 1350 bytes; the default is 64 bytes.

Consider the following points when using IPsec datapath verification:

- The source interface and destination IP addresses that you configure for VPN monitoring operation have no effect on the IPsec datapath verification. The source for the ICMP requests in the IPsec datapath verification is the local tunnel endpoint.
- When you enable IPsec datapath verification, Junos OS automatically activates VPN monitoring only after the st0 interface is up. We recommend that you configure the `vpn-monitor optimized` option at the `[edit security ipsec vpn vpn-name]` hierarchy level when you enable IPsec datapath verification.
- If a chassis cluster failover occurs during the IPsec datapath verification, the new active node starts the verification again. Junos OS doesn't active the st0 interface until the verification succeeds.
- For IPsec SA rekeys, Junos OS doesn't perform IPsec datapath verification, because the st0 interface state does not change for rekeys.
- Junos OS does not support IPsec datapath verification on st0 interfaces in a point-to-multipoint mode that are used with AutoVPN, Auto Discovery VPN (ADVPN), and multiple traffic selectors.
- VPN monitoring and IPsec datapath verification do not support IPv6 addresses. Therefore, you cannot use IPsec datapath verification with IPv6 tunnels.

How Does IPsec Datapath Verification Work?

When you configure IPsec datapath verification, the following events occur:

1. After the device establishes the VPN tunnel, it sends an ICMP request to the peer tunnel endpoint to verify the IPsec datapath.
   The peer tunnel endpoint must be reachable by VPN monitor ICMP requests and must be able to respond to the ICMP request. While the datapath verification is in progress, the **VPN Monitoring** field in the `show security ipsec security-association detail` command output displays the letter **V**.
2. Junos OS activates st0 interface only when it receives a response from the peer. The `show interface st0.x` command output shows the st0 interface status during and after the datapath verification: **Link-Layer-Down** before the verification finishes and **Up** after the verification finishes successfully.
3. If the peer doesn't send an ICMP response, the device sends another ICMP request at the configured VPN monitor interval until it reaches the configured VPN monitor threshold value.

Note that the default VPN monitor interval is 10 seconds and the default VPN monitor threshold value is 10 times. If the verification does not succeed, the **KMD_VPN_DOWN_ALARM_USER** system log entry indicates the reason as a VPN monitoring verify-path error. The device logs an error under tunnel events in the `show security ipsec security-association detail` command output. The `show security ipsec tunnel-events-statistics` command displays the number of times the error occurred.
You can configure the VPN monitor interval and the VPN monitor threshold value using the `vpn-monitor-options` configuration option at the [`edit security ipsec`] hierarchy level.

4. If the peer doesn't send an ICMP response even after it reaches the VPN monitor threshold value, Junos OS brings down the VPN tunnel and renegotiates the VPN tunnel.

SEE ALSO

verify-path

Dead Peer Detection

IN THIS SECTION

- How Does DPD Work? | **1212**
- Configurable DPD Parameters | **1213**

Dead Peer Detection (DPD) is a standards-based protocol that uses the network traffic to detect the liveness of an IKE peer in an IPsec connection.

How Does DPD Work?

During IPsec tunnel creation, VPN peers negotiate to decide whether to use the dead peer detection (DPD) method or not. If the peers agree to use the DPD method, when there is no active traffic, the DPD protocol sends periodic messages to the peer and waits for a response. If the peer does not respond to the messages, the DPD protocol assumes that the peer is no longer available. The behavior of DPD is the same for both IKEv1 and IKEv2 protocols. DPD timers are active as soon as the IKE establishes Phase 1 Security Association (SA).

A firewall uses the DPD protocol to detect the liveness in an IPsec VPN connection.

Figure 66: Message Exchange in DPD Protocol

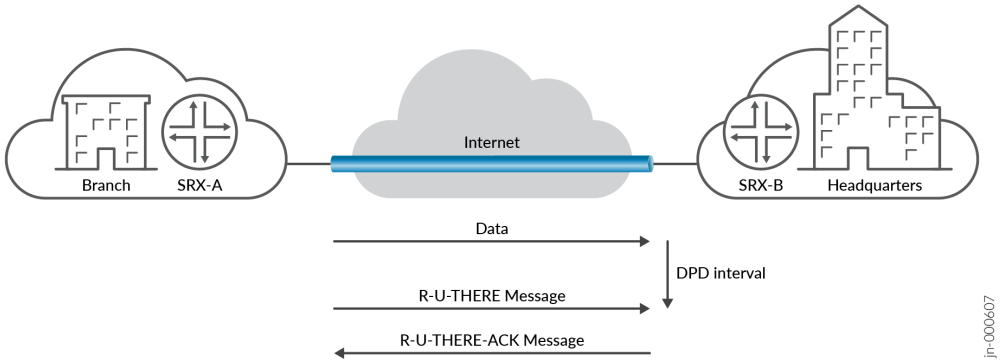


Figure 66 on page 1213 shows the exchange of DPD messages between the IKE peers in an IPsec VPN tunnel. The following events occur when the firewall device performs DPD:

1. The firewall SRX-A waits until the specified DPD interval to check whether it has received any traffic from the peer, SRX-B.
2. If SRX-A does not receive any traffic from SRX-B during the specified DPD interval, it sends an encrypted IKE Phase 1 notification payload—an R-U-THERE message—to SRX-B.
3. SRX-A waits for the DPD acknowledgment—an R-U-THERE-ACK message—from SRX-B.
    a. If SRX-A receives an R-U-THERE-ACK message from SRX-B during this interval, it considers the peer alive. Then SRX-A resets its R-U-THERE message counter for that tunnel, and starts a new interval.
    b. If SRX-A does not receive an R-U-THERE-ACK message during the interval, it considers the peer, SRX-B, down. SRX-A then removes the Phase 1 SA and all Phase 2 SA for that peer.

Configurable DPD Parameters

Here's a list of DPD parameters you'll need to configure:

- Mode—Based on the traffic activity, you can configure DPD in one of the following modes:
    - Optimized—In the *optimized* mode, when the initiating device sends outgoing packets to the peer, if there is no incoming IKE or IPsec traffic from the peer within the configured interval, the initiating device triggers R-U-THERE messages. DPD operates in this default mode unless you configure another mode.
    - Probe idle tunnel—In the *probe idle tunnel* mode, the device triggers R-U-THERE messages if there is no incoming or outgoing IKE or IPsec traffic within a configured interval. The device sends

R-U-THERE messages periodically to the peer until there is traffic activity. This mode helps in early detection of a peer that is down, ensuring tunnel availability during the active traffic flow.

> **NOTE**: In this scenario, when you configure probe idle tunnel mode, the device triggers R-U-THERE messages if a tunnel becomes idle regardless of the traffic in another tunnel for the same IKE SA.

  - Always-send—In the *always-send* mode, the device sends R-U-THERE messages at a configured interval regardless of traffic activity between the peers.
    We recommend that you use probe idle tunnel mode instead of always-send mode.

- Interval—Use the *interval* parameter to specify the amount of time (in seconds) that the device waits for traffic from its peer before sending an R-U-THERE message. The default interval is 10 seconds. The permissible interval parameter range at which R-U-THERE messages are sent to the peer device is 2 seconds through 60 seconds. We recommend that you set the minimum threshold parameter to 3, when the DPD interval parameter is set to less than 10 seconds.

- Threshold—Use the *threshold* parameter to specify the maximum number of times the device sends the R-U-THERE message without receiving a response from the peer before it considers the peer down. The default number of transmissions is five, with a permissible range of 1 through 5 retries.

Note the following considerations before configuring DPD:

- After you add the DPD configuration to an existing gateway with active tunnels, the device starts triggering R-U-THERE messages without clearing Phase 1 or Phase 2 SA.

- When you delete the DPD configuration from an existing gateway with active tunnels, the device stops triggering R-U-THERE messages for the tunnels. But this doesn't affect IKE and IPsec SA.

- When you modify DPD configuration parameters such as the mode, interval, or threshold values, IKE updates the DPD operation without clearing Phase 1 or Phase 2 SAs.

- If you configure the IKE gateway with DPD and VPN monitoring without specifying the option to establish tunnels immediately, IKE does not initiate Phase 1 negotiation. When you configure DPD, you must also configure the `establish-tunnels immediately` option at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level to tear down the st0 interface when no phase 1 and phase 2 SAs are available. See vpn (Security) for `establish-tunnels` option .

- If you configure the IKE gateway with multiple peer IP addresses and DPD, but fail to establish Phase 1 SA with the first peer IP address, IKE attempts to establish with the next peer IP address. DPD is active only after the IKE establishes Phase 1 SA. See dead-peer-detection.

- If you configure the IKE gateway with multiple peers and DPD, but the connection fails with the current peer's IP address, IKE clears Phase 1 and Phase 2 SAs and DPD does a failover to the next peer IP address. See gateway (Security IKE).

- More than one Phase 1 or Phase 2 SA can exist with the same peer because of simultaneous negotiations. In this case, DPD sends R-U-THERE messages to all Phase 1 SA. If the gateway fails to receive DPD responses for the configured number of consecutive times, it clears the Phase 1 SA and the associated Phase 2 SA (for IKEv2 only).



NOTE: For more details about DPD implementation, see RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*.

If the IKE peer is alive, does it mean that the underlying VPN is up?



TIP: Think whether DPD ensures IPsec SA liveness. See "VPN Tunnel Monitoring" on page 1215 .

SEE ALSO

dead-peer-detection

gateway (Security IKE)

VPN Tunnel Monitoring

IN THIS SECTION

- How Does VPN Tunnel Monitoring Work? | **1216**

VPN monitoring is a Junos OS proprietary feature of monitoring a VPN tunnel.

While the Dead Peer Detection (DPD) protocol checks the liveness of an IKE peer, it does not guarantee the liveness of an underlying VPN. We've no standards-based method to check whether the underlying VPN is up. VPN monitoring is a Junos OS proprietary mechanism of checking the liveness of an IPsec security association.

How Does VPN Tunnel Monitoring Work?

VPN monitoring uses Internet Control Message Protocol (ICMP) echo requests (or pings) and signature data, such as tunnel ID, in the ICMP packet to determine whether the VPN tunnel is up.

When you enable VPN monitoring, the device sends ICMP echo requests through the VPN tunnel to the peer gateway or to a specified destination at the other end of the tunnel. The device sends the requests by default at intervals of 10 seconds for up to 10 consecutive times. If the device doesn't receive any reply after 10 consecutive pings, it considers the VPN down and clears the IPsec security association.

Use the following operating modes to monitor VPN tunnels:

- Always-send mode—In this mode, the device sends a VPN monitoring packet once every configured interval irrespective of the traffic in the tunnel. After you enable VPN monitoring, Junos OS uses always-send mode as the default mode if you don't specify one.
- Optimized mode—In this mode, the device sends a VPN monitoring packet once every configured interval only if there is outgoing traffic and no incoming traffic through the tunnel during the interval. If there is incoming traffic through the VPN tunnel, the device considers the tunnel to be active and stops sending pings to the peer. You can use optimized mode to save resources on the device because in this mode the device sends pings only when it needs to determine peer liveness. Sending pings can also activate costly backup links that would otherwise not be used.

  The device operates in the default always-send mode if you don't configure optimized mode explicitly.

SEE ALSO

> vpn-monitor
>
> vpn (Security)

Configure Dead Peer Detection

Before you begin, ensure that you have the IKE gateway configured. See gateway (Security IKE) for more details.

In this topic, you'll learn how to configure the Dead Peer Detection (DPD) protocol and its parameters on your firewall. To enable the device with DPD:

1. Specify the DPD mode `probe-idle-tunnel` at the `[edit security ike gateway gateway-name]` hierarchy level. See "Configurable DPD Parameters" on page 1212 and dead-peer-detection for more details.

```
[edit]
set security ike gateway vpngw1 dead-peer-detection probe-idle-tunnel
```

   Here we use vpngw1 as the *gateway-name*.

2. Configure the interval at the `[edit security ike gateway gateway-name]` hierarchy level. See "Configurable DPD Parameters" on page 1212 and dead-peer-detection for more details.

```
[edit]
set security ike gateway vpngw1 dead-peer-detection interval 40
```

   Here we set an interval of 40 seconds.

3. Specify the threshold at the `[edit security ike gateway gateway-name]` hierarchy level. See "Configurable DPD Parameters" on page 1212 and dead-peer-detection for more details.

```
[edit]
set security ike gateway vpngw1 dead-peer-detection threshold 3
```

   Here we set a threshold of 3, which means the device will ping the peer 3 times before it considers the peer down.



NOTE: With the firewall running the **iked** process for the IPsec VPN service, you can use DPD with multiple peer addresses per gateway. See gateway (Security IKE) for more details.

Configure VPN Tunnel Monitoring

Before you begin, you must have an existing VPN tunnel.

In this topic, you'll learn how to enable VPN tunnel monitoring, and set the interval and threshold parameters for the ping packets used for VPN monitoring on your firewall.

1. Enable VPN monitoring for a specific VPN tunnel using the `vpn-monitor` option at the `[edit security ipsec vpn vpn-name]` hierarchy level. See vpn-monitor.

```
[edit]
set security ipsec vpn vpn1 vpn-monitor
```

   Here we use vpn1 as the *vpn-name*.

2. Configure VPN monitoring mode as `optimized`.

```
[edit]
set security ipsec vpn vpn1 vpn-monitor optimized
```

3. Specify the destination IP address. The peer gateway's IP address is the default destination; however, you can specify a different destination IP address (such as that of a server) that is at the other end of the tunnel.

```
[edit]
set security ipsec vpn vpn1 vpn-monitor destination-ip 192.168.10.11
```

4. Specify the `source-interface` address. The local tunnel endpoint is the default source interface, but you can specify a different interface name.

```
[edit]
set security ipsec vpn vpn1 vpn-monitor source-interface ge-0/0/5
```

5. Configure the interval at which the device sends the pings and the number of consecutive pings that it sends with the `interval` and `threshold` options, respectively, at the `[edit security ipsec vpn-monitor-options]` hierarchy level. If you don't configure these options, the device sends pings at the default interval of 10 seconds up to 10 consecutive times. If the device doesn't receive a reply, it considers the VPN down. The device then clears the IPsec security association. See ipsec (Security).

```
[edit]
set security ipsec vpn-monitor-options interval 20
set security ipsec vpn-monitor-options threshold 20
```

CAUTION: VPN monitoring can cause tunnel flapping in some environments if ping packets are not accepted by the peer based on the packet's source or destination IP address.

Platform-Specific VPN Monitoring Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform.

Table 122: Platform-Specific Behavior

Platform	Difference
SRX Series	• On SRX5400, SRX5600, and SRX5800 devices that support VPN monitoring, the destination for `vpn-monitor` must be a local interface. The firewall cannot perform VPN monitoring of an externally connected device such as a PC.

VPN Alarms, Audits, and Events

SUMMARY

Read this topic to understand various types of Junos OS alarms, logs, and events.

IN THIS SECTION

- VPN Alarms and Audits | **1220**
- VPN Tunnel Events | **1222**
- Configure VPN Alarms | **1223**
- Example: Configure an Audible Alert Notification | **1224**
- Example: Configure Security Alarms Generation | **1225**

Platform-Specific VPN Alarms Behavior | 1230

Junos OS records various events, maintains logs, and triggers alarms pertaining to the operation you perform on a device. While the VPN monitoring methods provide an active monitoring technique for your VPN, the alarms, the events, and the audits provide recorded information to analyze the cause of a failure. You might notice these failures before and after the tunnel creation.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific VPN Alarms Behavior" on page 1230 section for notes related to your platform.

VPN Alarms and Audits

You can analyze and understand the cause of VPN related failures using the alarms that Junos OS generates.

A VPN alarm is an indication that the configured VPN is in a state that may require user intervention to resolve. You'll see an alarm when the device reports a failure while monitoring the audited events. While an event is an occurrence that happens at a specific point of time, an alarm is an indication of failure state.

Note the following points when monitoring the alarms and events:

- Make sure you enable security event logging during the initial setup of the device using the `[set security log cache]` command. See cache (Security Log) for more details.
- Every administrator has a unique set of privileges based on the roles such as audit-administrator, cryptographic-administrator, IDS-administrator, and security-administrator. The other administrators cannot modify security event logging after it is enabled by the authorized administrator.

A VPN failure triggers an alarm when the system monitors any of the following audited events:

- *Authentication failures*—You can configure the device to generate a system alarm when the number of packet authentication failures reaches a specified threshold.
- *Encryption and decryption failures*—You can configure the device to generate a system alarm when the number of encryption or decryption failures exceeds a specified threshold.
- *IKE Phase 1 and IKE Phase 2 failures*—Junos OS establishes Internet Key Exchange (IKE) security associations (SAs) during IKE Phase 1 negotiations. The security associations protect the IKE Phase 2

negotiations. You can configure the device to generate a system alarm when the number of IKE Phase 1 or IKE Phase 2 failures exceeds a specified limit.

- *Self-test failures*—After a device powers on or reboots, Junos OS runs a few tests on the device to verify the implementation of the security software.
  Self-tests ensure the correctness of cryptographic algorithms. The Junos-FIPS image performs self-tests automatically after the device powers on and runs the test continuously for key-pair generation. In either the domestic or FIPS images, you can configure self-tests to run according to a defined schedule, on demand, or immediately after key generation.
  You can also configure the device to generate a system alarm when a self-test fails.

- *IDP flow policy attacks*—You use an intrusion detection and prevention (IDP) policy to enforce various attack detection and prevention techniques on the network traffic. You can configure the device to generate a system alarm when an IDP flow policy violation occurs.

- *Replay attacks*—A replay attack is a network attack in which an attacker maliciously or fraudulently repeats or delays a valid data transmission event. You can configure the device to generate a system alarm when a replay attack occurs.

Junos OS generates system log messages (also called *syslog messages*) to record the events that occur on the device. You'll notice syslog messages in the following cases:

- Failed symmetric key generation
- Failed asymmetric key generation
- Failed manual key distribution
- Failed automated key distribution
- Failed key destruction
- Failed key handling and storage
- Failed data encryption or decryption
- Failed signature
- Failed key agreement
- Failed cryptographic hashing
- IKE failure
- Failed authentication of the received packets
- Decryption error due to invalid padding content

- Mismatch in the length specified in the alternative subject field of the certificate received from a remote VPN peer device

Junos OS triggers alarms based on the syslog messages. While Junos OS logs every failure, it generates an alarm only when the threshold is reached. To view the alarm information, run the `show security alarms` command. The violation count and the alarm do not persist across system reboots. After a reboot, the violation count resets to zero, and the alarm is cleared from the alarm queue. The alarm remains in the queue until you reboot the device or until you clear the alarm after taking the appropriate actions. To clear the alarm, run the `clear security alarms` command.

SEE ALSO

cache (Security Log)

show security alarms

clear security alarms

VPN Tunnel Events

After a VPN tunnel comes up, Junos OS tracks the tunnel status related to tunnel down issues and negotiation failures. Junos OS also tracks successful events such as successful IPsec security association negotiations, IPsec rekey, and IKE security association rekeys. We use the term *tunnel events* to refer to these failure and success events.

For Phase 1 and Phase 2, Junos OS tracks the negotiation events for a given tunnel with the iked or the kmd process along with the events that occur in external processes such as the authd or the pkid. When a tunnel event occurs multiple times, the device maintains only one entry with the updated time and the number of times that event occurred.

Overall, Junos OS tracks 16 events–eight events for Phase 1 and eight events for Phase 2.

Some events can reoccur and fill up the event memory, resulting in important events being removed. To avoid overwriting, the device doesn't store an event unless the tunnel is down. Few of these events are listed below:

- Lifetime in kilobytes expired for IPsec security association.
- Hard lifetime of IPsec security association expired.
- IPsec security associations cleared as the corresponding delete payload received from the peer.
- Cleared unused redundant backup IPsec security association pairs.

- IPsec security associations cleared as corresponding IKE security association deleted.

Junos OS creates and removes AutoVPN tunnels dynamically. As a result, tunnel events corresponding to these tunnels are short lived. Not all tunnel events are associated with a tunnel, nevertheless, you can use them for debugging.

Configure VPN Alarms

Before you begin, you must ensure that the security event logging is enabled using the following command during the initial setup of the device:

```
[edit]
set security log cache
```

See cache (Security Log) for more details.

In this task, you'll see how to view and clear the alarms on your firewalls.

1. To view the alarm information, run the following command in operational mode.

   ```
   show security alarms
   ```

   See show security alarms for details.
2. To clear the alarm information, run the following command in operational mode.

   ```
   clear security alarms
   ```

   See clear security alarms for details.

Example: Configure an Audible Alert Notification

IN THIS SECTION

- Requirements | 1224
- Overview | 1224
- Configuration | 1224
- Verification | 1225

This example shows how to configure a device to generate a system alert beep when a new security event occurs. By default, alarms are not audible.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you set an audible beep to be generated in response to a security alarm.

Configuration

IN THIS SECTION

- Procedure | 1224

Procedure

Step-by-Step Procedure

To set an audible alarm:

1. Enable security alarms.

```
[edit]
user@host# edit security alarms
```

2. Specify that you want to be notified of security alarms with an audible beep.

```
[edit security alarms]
user@host# set audible
```

3. If you are done configuring the device, commit the configuration.

```
[edit security alarms]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the `show security alarms detail` command.

SEE ALSO

IPsec Overview | 12

Example: Configure Security Alarms Generation

IN THIS SECTION

- Requirements | 1226
- Overview | 1226
- Configuration | 1226
- Verification | 1229

This example shows how to configure the device to generate a system alarm when a potential violation occurs. By default, no alarm is raised when a potential violation occurs.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you configure an alarm to be raised when:

- The number of authentication failures exceeds 6.
- The cryptographic self-test fails.
- The non-cryptographic self-test fails.
- The key generation self-test fails.
- The number of encryption failures exceeds 10.
- The number of decryption failures exceeds 1.
- The number of IKE Phase 1 failures exceeds 10.
- The number of IKE Phase 2 failure exceeds 1.
- A replay attack occurs.

Configuration

IN THIS SECTION

- Procedure | 1227

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security alarms potential-violation authentication 6
set security alarms potential-violation cryptographic-self-test
set security alarms potential-violation non-cryptographic-self-test
set security alarms potential-violation key-generation-self-test
set security alarms potential-violation encryption-failures threshold 10
set security alarms potential-violation decryption-failures threshold 1
set security alarms potential-violation ike-phase1-failures threshold 10
set security alarms potential-violation ike-phase2-failures threshold 1
set security alarms potential-violation replay-attacks
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure alarms in response to potential violations:

1. Enable security alarms.

```
[edit]
user@host# edit security alarms
```

2. Specify that an alarm should be raised when an authentication failure occurs.

```
[edit security alarms potential-violation]
user@host# set authentication 6
```

3. Specify that an alarm should be raised when a cryptographic self-test failure occurs.

```
[edit security alarms potential-violation]
user@host# set cryptographic-self-test
```

4. Specify that an alarm should be raised when a non-cryptographic self-test failure occurs.

```
[edit security alarms potential-violation]
user@host# set non-cryptographic-self-test
```

5. Specify that an alarm should be raised when a key generation self-test failure occurs.

```
[edit security alarms potential-violation]
user@host# set key-generation-self-test
```

6. Specify that an alarm should be raised when an encryption failure occurs.

```
[edit security alarms potential-violation]
user@host# set encryption-failures threshold 10
```

7. Specify that an alarm should be raised when a decryption failure occurs.

```
[edit security alarms potential-violation]
user@host# set decryption-failures threshold 1
```

8. Specify that an alarm should be raised when an IKE Phase 1 failure occurs.

```
[edit security alarms potential-violation]
user@host# set ike-phase1-failures threshold 10
```

9. Specify that an alarm should be raised when an IKE Phase 2 failure occurs.

```
[edit security alarms potential-violation]
user@host# set ike-phase2-failures threshold 1
```

10. Specify that an alarm should be raised when a replay attack occurs.

```
[edit security alarms potential-violation]
user@host# set replay-attacks
```

Results

From configuration mode, confirm your configuration by entering the `show security alarms` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
potential-violation {
    authentication 6;
    cryptographic-self-test;
    decryption-failures {
        threshold 1;
    }
    encryption-failures {
        threshold 10;
    }
    ike-phase1-failures {
        threshold 10;
    }
    ike-phase2-failures {
        threshold 1;
    }
    key-generation-self-test;
    non-cryptographic-self-test;
    replay-attacks;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

To confirm that the configuration is working properly, from operational mode, enter the `show security alarms` command.

SEE ALSO

VPN Support for Inserting Services Processing Cards | 86

Platform-Specific VPN Alarms Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform.

Table 123: Platform-Specific Behavior

Platform	Difference
SRX Series	On SRX300, SRX320, SRX340, SRX345, SRX1500, and vSRX instances that support this feature: • You can configure the devices to generate a system alert beep when a new security event occurs. By default, alarms aren't audible. • You can configure the devices to generate a system alarm when a potential violation occurs. By default, the device doesn't raise an alarm when a potential violation occurs.

12

CHAPTER

Performance Tuning

IN THIS CHAPTER

- VPN Session Affinity | **1232**
- PowerMode IPsec | **1241**

VPN Session Affinity

SUMMARY

Learn how to improve the performance of IPsec VPN using VPN session affinity.

IN THIS SECTION

- Understanding VPN Session Affinity | **1232**
- Enabling VPN Session Affinity | **1234**
- Accelerating the IPsec VPN Traffic Performance | **1236**
- IPsec Distribution Profile | **1238**
- Understanding the Loopback Interface for a High Availability VPN | **1239**
- Platform-Specific High Availability VPN Loopback Interface Behavior | **1239**

The performance of IPsec VPN traffic to minimize packet forwarding overhead can be optimized by enabling VPN session affinity and performance acceleration.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific High Availability VPN Loopback Interface Behavior" on page 1239 section for notes related to your platform.

Understanding VPN Session Affinity

VPN session affinity occurs when a cleartext session is located in a Services Processing Unit (SPU) that is different from the SPU where the IPsec tunnel session is located. The goal of VPN session affinity is to locate the cleartext and IPsec tunnel session in the same SPU.

Without VPN session affinity, a cleartext session created by a flow might be located in one SPU and the tunnel session created by IPsec might be located in another SPU. An SPU to SPU forward or hop is needed to route cleartext packets to the IPsec tunnel.

By default, VPN session affinity is disabled on the firewalls. When VPN session affinity is enabled, a new cleartext session is placed on the same SPU as the IPsec tunnel session. Existing cleartext sessions are not affected.

The firewalls support VPN session affinity through improved flow module and session cache. With IOCs, the flow module creates sessions for IPsec tunnel-based traffic before encryption and after decryption on its tunnel-anchored SPU and installs the session cache for the sessions so that the IOC can redirect the packets to the same SPU to minimize packet forwarding overhead. Express Path (previously known as services offloading) traffic and NP cache traffic share the same session cache table on the IOCs.

To display active tunnel sessions on SPUs, use the `show security ipsec security-association` command and specify the Flexible PIC Concentrator (FPC) and *Physical Interface Card* (PIC) slots that contain the SPU. For example:

```
user@host> show security ipsec security-association fpc 3 pic 0
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <131073 ESP:aes-128/sha1 18c4fd00 491/  128000 - root 500   203.0.113.11
  >131073 ESP:aes-128/sha1 188c0750 491/  128000 - root 500   203.0.113.11
```

You need to evaluate the tunnel distribution and traffic patterns in your network to determine if VPN session affinity should be enabled.

If VPN session affinity is enable on the firewall, the tunnel overhead is calculated according to the negotiated encryption and authentication algorithms on the anchor Services Processing Unit (SPU). If the configured encryption or authentication changes, the tunnel overhead is updated on the anchor SPU when a new IPsec security association is established.

The VPN session affinity limitations are as follows:

- Traffic across logical systems is not supported.
- If there is a route change, established cleartext sessions remain on an SPU and traffic is rerouted if possible. Sessions created after the route change can be set up on a different SPU.
- VPN session affinity only affects self traffic that terminates on the device (also known as host-inbound traffic); self traffic that originates from the device (also known as host-outbound traffic) is not affected.
- Multicast replication and forwarding performance is not affected.

SEE ALSO

Understanding Traffic Processing on SRX5000 Line Devices

Understanding Session Cache

Express Path Overview

Example: Enabling Express Path in Security Policies

Express Path

Enabling VPN Session Affinity

By default, VPN session affinity is disabled on the firewalls. Enabling VPN session affinity can improve VPN throughput under certain conditions. This section describes how to use the CLI to enable VPN session affinity.

Determine if clear-text sessions are being forwarded to IPsec tunnel sessions on a different SPU. Use the `show security flow session` command to display session information about clear-text sessions.

```
user@host> show security flow session
Flow Sessions on FPC3 PIC0:

Session ID: 60000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/6204 --> 203.0.113.6/41264;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000003, Policy name: self-traffic-policy/1, Timeout: 58, Valid
  In: 203.0.113.6/500 --> 203.0.113.11/500;udp, If: .local..0, Pkts: 105386, Bytes: 12026528
  Out: 203.0.113.11/500 --> 203.0.113.6/500;udp, If: ge-0/0/2.0, Pkts: 106462, Bytes: 12105912

Session ID: 60017354, Policy name: N/A, Timeout: 1784, Valid
  In: 0.0.0.0/0 --> 0.0.0.0/0;0, If: N/A, Pkts: 0, Bytes: 0
  Out: 198.51.100.156/23 --> 192.0.2.155/53051;tcp, If: N/A, Pkts: 0, Bytes: 0
Total sessions: 4

Flow Sessions on FPC6 PIC0:

Session ID: 120000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 120000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 120031730, Policy name: default-policy-00/2, Timeout: 1764, Valid
  In: 192.0.2.155/53051 --> 198.51.100.156/23;tcp, If: ge-0/0/1.0, Pkts: 44, Bytes: 2399
```

```
  Out: 198.51.100.156/23 --> 192.0.2.155/53051;tcp, If: st0.0, Pkts: 35, Bytes: 2449
Total sessions: 3
```

In the example, there is a tunnel session on FPC 3, PIC 0 and a clear-text session on FPC 6, PIC 0. A forwarding session (session ID 60017354) is set up on FPC 3, PIC 0.

You can enable session affinity for the IPsec tunnel session on the IOC FPCs. To enable IPsec VPN affinity, you must also enable the session cache on IOCs by using the `set chassis fpc` *`fpc-slot`* `np-cache` command.

To enable VPN session affinity:

1. In configuration mode, use the `set` command to enable VPN session affinity.

```
[edit]
user@host# set security flow load-distribution session-affinity ipsec
```

2. Check your changes to the configuration before committing.

```
[edit]
user@host# commit check
```

3. Commit the configuration.

```
[edit]
user@host# commit
```

After enabling VPN session affinity, use the `show security flow session` command to display session information about clear-text sessions.

```
user@host> show security flow session
Flow Sessions on FPC3 PIC0:

Session ID: 60000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/6352 --> 203.0.113.6/7927;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000003, Policy name: self-traffic-policy/1, Timeout: 56, Valid
  In: 203.0.113.6/500 --> 203.0.113.11/500;udp, If: .local..0, Pkts: 105425, Bytes: 12031144
```

```
  Out: 203.0.113.11/500 --> 203.0.113.6/500;udp, If: ge-0/0/2.0, Pkts: 106503, Bytes: 12110680

Session ID: 60017387, Policy name: default-policy-00/2, Timeout: 1796, Valid
  In: 192.0.2.155/53053 --> 198.51.100.156/23;tcp, If: ge-0/0/1.0, Pkts: 10, Bytes: 610
  Out: 198.51.100.156/23 --> 192.0.2.155/53053;tcp, If: st0.0, Pkts: 9, Bytes: 602
Total sessions: 4

Flow Sessions on FPC6 PIC0:

Session ID: 120000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 120000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0
Total sessions: 2
```

After VPN session affinity is enabled, the clear-text session is always located on FPC 3, PIC 0.

SEE ALSO

- Understanding Session Cache
- Express Path Overview

Accelerating the IPsec VPN Traffic Performance

You can accelerate IPsec VPN performance by configuring the performance acceleration parameter. By default, VPN performance acceleration is disabled on the firewalls. Enabling the VPN performance acceleration can improve the VPN throughput with VPN session affinity enabled.

This topic describes how to use the CLI to enable VPN performance acceleration.

To enable performance acceleration, you must ensure that cleartext sessions and IPsec tunnel sessions are established on the same Services Processing Unit (SPU). IPsec VPN performance is optimized when the VPN session affinity and performance acceleration features are enabled. For more information on enabling session affinity, see "Understanding VPN Session Affinity" on page 1232.

To enable IPsec VPN performance acceleration:

1. Enable VPN session affinity.

```
[edit]
user@host# set security flow load-distribution session-affinity ipsec
```

2. Enable IPsec performance acceleration.

```
[edit]
user@host# set security flow ipsec-performance-acceleration
```

3. Check your changes to the configuration before committing.

```
[edit]
user@host# commit check
```

4. Commit the configuration.

```
[edit]
user@host# commit
```

After enabling VPN performance acceleration, use the `show security flow status` command to display flow status.

```
Flow forwarding mode:
   Inet forwarding mode: flow based
   Inet6 forwarding mode: drop
   MPLS forwarding mode: drop
   ISO forwarding mode: drop
 Flow trace status
   Flow tracing status: off
 Flow session distribution
   Distribution mode: Hash-based
       Flow packet ordering
   Ordering mode: Hardware
 Flow ipsec performance acceleration: on
```

SEE ALSO

ipsec-performance-acceleration (Security Flow)

show security flow status

IPsec Distribution Profile

You can configure one or more IPsec distribution profiles for IPsec security associations (SAs). Tunnels are distributed evenly across all resources (SPCs) specified in the configured distribution profile. It is supported in SPC3 only and mixed-mode (SPC3 + SPC2), it is not supported on SPC1 and SPC2 systems. With the IPsec distribution profile, use the `set security ipsec vpn` *`vpn-name`* `distribution-profile` *`distribution-profile-name`* command to associate tunnels to a specified:

- Slot
- PIC

Alternatively, you can use the default IPsec distribution profiles:

- `default-spc2-profile` —Use this predefined default profile to associate IPsec tunnels to all available SPC2 cards.
- `default-spc3-profile` —Use this predefined default profile to associate IPsec tunnels to all available SPC3 cards.

You can now assign a profile to a specific VPN object, where all associated tunnels will be distributed based on this profile. If no profile is assigned to the VPN object, the firewall automatically distributes these tunnels evenly across all resources.

You can associate a VPN object with either a user-defined profile or a predefined (default) profile.

In the following example, all tunnels associated with profile ABC will be distributed on FPC 0, PIC 0.

```
userhost# show security {
    distribution-profile ABC {
        fpc 0 {
            pic 0;
        }
    }
}
```

Understanding the Loopback Interface for a High Availability VPN

In an IPsec VPN tunnel configuration, an external interface must be specified to communicate with the peer IKE gateway. Specifying a loopback interface for the external interface of a VPN is a good practice when there are multiple physical interfaces that can be used to reach a peer gateway. Anchoring a VPN tunnel on the loopback interface removes the dependency on a physical interface for successful routing.

Using a loopback interface for VPN tunnels is supported on standalone firewalls as well as on firewalls in chassis clusters. In a chassis cluster active-passive deployment, you can create a logical loopback interface and make it a member of a redundancy group so that it can be used to anchor VPN tunnels. The loopback interface can be configured in any redundancy group and is assigned as the external interface for the IKE gateway. VPN packets are processed on the node where the redundancy group is active.

In a chassis cluster setup, the node on which the external interface is active selects an SPU to anchor the VPN tunnel. IKE and IPsec packets are processed on that SPU. Thus an active external interface determines the anchor SPU.

You can use the `show chassis cluster interfaces` command to view information on the redundant pseudointerface.

SEE ALSO

> *show chassis cluster interfaces*

Platform-Specific High Availability VPN Loopback Interface Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platforms.

Table 124: Platform-Specific Behavior

Platform	Difference

SRX Series	• On SRX5400, SRX5600, and SRX5800 devices that support loopback interface for high availability VPNs:
	  • For SPC2-based devices that run the kmd process, if you use the loopback interface as the IKE gateway external interface, configure the interface binding in a redundancy group other than RG0.
	  • For SPC3 or SPC3+SPC2-based devices that run the iked process, you do not need to bind the loopback interface to a redundancy group.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.2R2	Starting in Junos OS Release 20.2R2, the invalid thread IDs configured to the distribution profile are ignored with no commit-check error message. The IPsec tunnel gets anchored as per the configured distribution profile ignoring invalid thread IDs if any for that profile.
19.2R1	Starting in Junos OS Release 19.2R1, you can configure one or more IPsec distribution profiles for IPsec security associations (SAs).
17.4R1	Starting with Junos OS Release 17.4R1, IPsec VPN performance is optimized when the VPN session affinity and performance acceleration features are enabled.
12.3X48-D50	Starting with Junos OS Release 12.3X48-D50, Junos OS Release 15.1X49-D90, and Junos OS Release 17.3R1, if VPN session affinity is enabled on SRX5000 line, the tunnel overhead is calculated according to the negotiated encryption and authentication algorithms on the anchor Services Processing Unit (SPU).

RELATED DOCUMENTATION

VPN Support for Inserting Services Processing Cards | 86

IPsec VPN Configuration Overview | 109

PowerMode IPsec

SUMMARY

Read this topic to learn about PowerMode IPsec (PMI) VPNs.

IN THIS SECTION

- Improving IPsec Performance with PowerMode IPsec | **1241**
- Example: Configuring Behavior Aggregate Classifier in PMI | **1247**
- Example: Configuring Behavior Aggregate Classifier in PMI for vSRX Virtual Firewall Instances | **1252**
- Example: Configuring and Applying a Firewall Filter for a Multifield Classifier in PMI | **1259**
- Example: Configuring and Applying Rewrite Rules on a Security Device in PMI | **1266**
- Configure IPsec ESP Authentication-only Mode in PMI | **1271**
- Platform-Specific PMI Behavior | **1272**
- Additional Platform Information | **1273**

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific PMI Behavior" on page 1272 section for notes related to your platform.

See the "Additional Platform Information" on page 1273 section for more information.

Improving IPsec Performance with PowerMode IPsec

IN THIS SECTION

- PMI Processing | **1242**
- PMI Statistics | **1242**

- Advanced Encryption Standard New Instructions (AES-NI) and Inline Field-Programmable Gate Array (FPGA) | **1242**
- Supported and Non-Supported Features for PMI | **1243**
- Benefits of PMI | **1245**
- Configuring Security Flow PMI | **1245**
- Understanding Symmetric Fat Tunnel | **1246**

PowerMode IPsec (PMI) is a mode of operation that provides IPsec performance improvements using Vector Packet Processing and Intel Advanced Encryption Standard New Instructions (AES-NI). PMI utilizes a small software block inside the Packet Forwarding Engine that bypasses flow processing and utilizes the AES-NI instruction set for optimized performance of IPsec processing that gets activated when PMI is enabled.

PMI Processing

You can enable or disable PMI processing:

- Enable PMI processing by using the `set security flow power-mode-ipsec` configuration mode command.
- Disable PMI processing by using the `delete security flow power-mode-ipsec` configuration mode command. Executing this command deletes the statement from the configuration.

PMI Statistics

You can verify the PMI statistics by using the `show security flow pmi statistics` operational mode command.

You can verify the PMI and fat tunnel status by using the `show security flow status` operational mode command.

Advanced Encryption Standard New Instructions (AES-NI) and Inline Field-Programmable Gate Array (FPGA)

. AES-NI in PMI mode helps in balancing the load in SPUs and supports the symmetric fat tunnel in SPC3 cards. This results in accelerated traffic-handling performance and higher throughput for IPsec VPN. PMI uses AES-NI for encryption and FPGA for decryption of cryptographic operation.

To enable PMI processing with AES-NI, include the `power-mode-ipsec` statement at the `[edit security flow]` hierarchy level.

To enable or disable inline FPGA, include the `inline-fpga-crypto (disabled | enabled)` statement at the `[edit security forwarding-process application-services]` hierarchy level.

Supported and Non-Supported Features for PMI

A tunnel session can either be PMI or non-PMI. Table 125 on page 1243 summarizes the supported and non-supported PMI features.

Review the Table 128 on page 1272 section for notes related to your platform.

If a session is configured with any non-supported features listed in Table 125 on page 1243, the session is marked as non-PMI and the tunnel goes into non-PMI mode. Once the tunnel goes into the non-PMI mode, the tunnel does not return to the PMI mode.

Table 125: Summary of Supported and Non-supported Features in PMI

Supported Features in PMI	Non-Supported Features in PMI
Internet Key Exchange (IKE) functionality	Layer 4 - 7 applications: application firewall and AppSecure
AutoVPN with traffic selectors	Multicast
High availability	Nested tunnels
IPv6	Screen options
Stateful firewall	Application Layer Gateway (ALG)
st0 interface	
Traffic selectors	
NAT (In PMI mode, you cannot use NAT64. NAT64 works properly in normal mode, when PMI is enabled.)	

Table 125: Summary of Supported and Non-supported Features in PMI *(Continued)*

Supported Features in PMI	Non-Supported Features in PMI
AES-GCM-128 and AES-GCM-256 encryption algorithm. We recommend you to use AES-GCM encryption algorithm for optimal performance.	
AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA1 encryption algorithm with HMAC-SHA1-96 authentication algorithm	
AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA2 encryption algorithm with HMAC-SHA-256-128 authentication algorithm	
NULL encryption algorithm	
ChaCha20-Poly1305 authenticated encryption algorithm	
HMAC-SHA-384 and HMAC-SHA-512 authentication algorithm	

Note the following usage considerations with PMI:

- **Antireplay window size**
  - Antireplay window size is 64 packets by default. If you configure fat-tunnel, then it is recommended to increase the Antireplay window size to greater than or equal to 512 packets.
- **Class of Service (CoS)**
  - Class of Service(CoS) supports configuration of behavior aggregate (BA) classifier, multifield (MF) classifier, and rewrite-rule functions in PMI.
  - If you enable PMI for a flow session, then the CoS is performed based on a per-flow basis. This means, the first packet of a new flow caches the CoS information in the flow session. Then the subsequent packets of the flow reuse the CoS information cached in the session.
- **Encryption algorithm**
  - PMI supports the options aes-128-cbc, aes-192-cbc, and aes-256-cbc to improve IPsec performance, along with the existing support in normal mode.
- **GTP-U**
  - PMI supports GTP-U scenario with TEID distribution and asymmetric fat tunnel solution.

    - PMI supports Software Receive Side Scaling feature.

- **LAG and redundant (reth) interfaces**
    - PMI is supported on link aggregation group (LAG) and redundant Ethernet (reth) interfaces.
- **PMI fragmentation check**
    - PMI does a pre-fragmentation and post-fragmentation check. If the PMI detects pre-fragmentation and post-fragmentation packets, packets are not allowed through the PMI mode. The packets will return to non-PMI mode.
    - Any fragments received on an interface does not go through PMI.
- **PMI for NAT-T**
    - PMI for NAT-T is supported only on SRX5K-SPC3 Services Processing Card (SPC), or with vSRX Virtual Firewall.
- **PMI support (vSRX)**
    - vSRX Virtual Firewall instances support:
        - Per-flow CoS functions for GTP-U traffic in PMI mode.
        - CoS features in PMI mode. The following CoS features are supported in PMI mode:
            - Classifier
            - Rewrite-rule functions
            - Queuing
            - Shaping
            - Scheduling

Benefits of PMI

- Enhances the performance of IPsec.

Configuring Security Flow PMI

The below section describes you how to configure security flow PMI.

To configure security flow PMI, you must enable session cache on IOCs and session affinity:

1. Enable the session cache on IOCs (IOC2 and IOC3)

```
user@host# set chassis fpc <fpc-slot> np-cache
```

2. Enable VPN session affinity

```
user@host# set security flow load-distribution session-affinity ipsec
```

3. Create security flow in PMI.

```
user@host#set security flow power-mode-ipsec
```

4. Confirm your configuration by entering the `show security` command.

```
user@host# show security
flow {
    power-mode-ipsec;
}
```

Understanding Symmetric Fat Tunnel

To improve the throughput of IPsec tunnel, you can use fat tunnel technology.

A new CLI command is introduced to enable the fat IPsec tunnel. The fat IPsec tunnel feature is disabled by default. The new CLI command introduced is `fat-core` in the `set security distribution-profile` hierarchy. When you enable the fat-core, the below configuration is displayed:

```
security {
    distribution-profile {
        fat-core;
    }
}
```

Before configuring the fat IPsec tunnel, make sure the following are configured.

- For fast path forwarding, configure the IOC cache for the session information using the `set chassis fpc` *`FPC slot`* `np-cache` command.

- To enable session affinity, use the `set security flow load-distribution session-affinity ipsec` command.
- To enable Power mode, use the `set security flow power-mode-ipsec` command.

SEE ALSO

IPsec VPN Overview | 79

flow (Security Flow)

PMI Flow-Based CoS Functions for GTP-U

show security flow pmi statistics

Example: Configuring Behavior Aggregate Classifier in PMI

IN THIS SECTION

- Requirements | 1247
- Overview | 1248
- Configuration | 1248
- Verification | 1251

This example shows how to configure behavior aggregate(BA) classifiers for a SRX Series Firewall to determine forwarding treatment of packets in PMI.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall.
- Junos OS Release 19.1R1 and later releases.

Before you begin:

- Determine the forwarding class and PLP that are assigned by default to each well-known DSCP that you want to configure for the behavior aggregate classifier.

Overview

Configure behavior aggregate classifiers to classify the packets that contain valid DSCPs to appropriate queues. Once configured, you apply the behavior aggregate classifier to the correct interfaces. You override the default IP precedence classifier by defining a classifier and applying it to a logical interface. To define new classifiers for all code point types, include the `classifiers` statement at the `[edit class-of-service]` hierarchy level.

In this example, set the DSCP behavior aggregate classifier to `ba-classifier` as the default DSCP map. Set a best-effort forwarding class as `be-class`, an expedited forwarding class as `ef-class`, an assured forwarding class as `af-class`, and a network control forwarding class as `nc-class`. Finally, apply the behavior aggregate classifier to the interface ge-0/0/0.

Table 2 shows how the behavior aggregate classifier assigns loss priorities, to incoming packets in the four forwarding classes.

Table 126: Sample ba-classifier Loss Priority Assignments

mf-classifier Forwarding Class	For CoS Traffic Type	ba-classifier Assignments
`be-class`	Best-effort traffic	High-priority code point: 000001
`ef-class`	Expedited forwarding traffic	High-priority code point: 101111
`af-class`	Assured forwarding traffic	High-priority code point: 001100
`nc-class`	Network control traffic	High-priority code point: 110001

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1249**
- Procedure | **1249**
- Results | **1250**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set class-of-service classifiers dscp ba-classifier import default
set class-of-service classifiers dscp ba-classifier forwarding-class be-class loss-priority high
code-points 000001
set class-of-service classifiers dscp ba-classifier forwarding-class ef-class loss-priority high
code-points 101111
set class-of-service classifiers dscp ba-classifier forwarding-class af-class loss-priority high
code-points 001100
set class-of-service classifiers dscp ba-classifier forwarding-class nc-class loss-priority high
code-points 110001
set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp ba-classifier
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure Behavior Aggregate Classifiers for a device in PMI:

1. Configure the class of service.

```
[edit]
user@host# edit class-of-service
```

2. Configure behavior aggregate classifiers for Differentiated Services (DiffServ) CoS.

```
[edit class-of-service]
user@host# edit classifiers dscp ba-classifier
user@host# set import default
```

3. Configure a best-effort forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class be-class loss-priority high code-points 000001
```

4. Configure an expedited forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class ef-class loss-priority high code-points 101111
```

5. Configure an assured forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class af-class loss-priority high code-points 001100
```

6. Configure a network control forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class nc-class loss-priority high code-points 110001
```

7. Apply the behavior aggregate classifier to an interface.

```
[edit]
user@host# set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp ba-classifier
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
classifiers {
    dscp ba-classifier {
        import default;
        forwarding-class be-class {
```

```
            loss-priority high code-points 000001;
        }
        forwarding-class ef-class {
            loss-priority high code-points 101111;
        }
        forwarding-class af-class {
            loss-priority high code-points 001100;
        }
        forwarding-class nc-class {
            loss-priority high code-points 110001;
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            classifiers {
                dscp ba-classifier;
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying the Classifier is applied to the Interfaces | 1251

To confirm that the configuration is working properly, perform these tasks:

Verifying the Classifier is applied to the Interfaces

Purpose

Make sure that the classifier is applied to the correct interfaces.

Action

From the operational mode, enter the `show class-of-service interface ge-0/0/0` command.

```
user@host> show class-of-service interface ge-0/0/0
Physical interface: ge-0/0/0, Index: 144
 Queues supported: 8, Queues in use: 4
Scheduled map: <default>, Index:2
Congestion-notification: Disabled

LOgical interface: ge-1/0/3, Index: 333
Object      Name              Type    Index
Classifier   v4-ba-classifier  dscp    10755
```

Meaning

The interfaces are configured as expected.

Example: Configuring Behavior Aggregate Classifier in PMI for vSRX Virtual Firewall Instances

IN THIS SECTION

- Requirements | **1252**
- Overview | **1253**
- Configuration | **1254**
- Verification | **1258**

This example shows how to configure behavior aggregate (BA) classifiers for a vSRX Virtual Firewall instance to determine forwarding treatment of packets in PMI.

Requirements

This example uses the following hardware and software components:

- A vSRX Virtual Firewall instance.
- Junos OS Release 19.4R1 and later releases.

Before you begin:

- Determine the forwarding class and PLP that are assigned by default to each well-known DSCP that you want to configure for the BA classifier.

Overview

Configure BA classifiers to classify the packets that contain valid DSCPs to appropriate queues. Once configured, you apply the BA classifier to the correct interfaces. You override the default IP precedence classifier by defining a classifier and applying it to a logical interface. To define new classifiers for all code point types, include the `classifiers` statement at the `[edit class-of-service]` hierarchy level.

In this example, set the DSCP BA classifier to `ba-classifier` as the default DSCP map. Set a best-effort (BE) forwarding class as `be-class`, an expedited forwarding (EF) class as `ef-class`, an assured forwarding (AF) class as `af-class`, and a network control forwarding class as `nc-class`. Finally, apply the BA classifier to the interface ge-0/0/0.

Table 2 shows how the BA classifier assigns loss priorities, to incoming packets in the four forwarding classes.

Table 127: Sample ba-classifier Loss Priority Assignments

Multifield-Classifier Forwarding Class	For CoS Traffic Type	BA Classifier Assignments
`be-class`	BE traffic	High-priority code point: 000001
`ef-class`	EF traffic	High-priority code point: 101111
`af-class`	AF traffic	High-priority code point: 001100
`nc-class`	Network control traffic	High-priority code point: 110001

Configuration

IN THIS SECTION

- CLI Quick Configuration | 1254
- Procedure | 1255
- Results | 1256

CLI Quick Configuration

To quickly configure the example, copy the following commands and paste the commands into a text file. Next, remove line breaks and adjust details to fit your network configuration. Copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set class-of-service classifiers dscp ba-classifier forwarding-class be loss-priority low code-points be
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority low code-points ef
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority high code-points af41
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority high code-points af11
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority high code-points af31
set class-of-service classifiers dscp ba-classifier forwarding-class low_delay loss-priority low code-points af21
set class-of-service classifiers dscp ba-classifier forwarding-class low_loss loss-priority low code-points cs6
set class-of-service drop-profiles drop_profile fill-level 20 drop-probability 50
set class-of-service drop-profiles drop_profile fill-level 50 drop-probability 100
set class-of-service forwarding-classes queue 0 be
set class-of-service forwarding-classes queue 1 ef
set class-of-service forwarding-classes queue 2 low_delay
set class-of-service forwarding-classes queue 3 low_loss
set class-of-service interfaces ge-0/0/1 unit 0 classifiers dscp ba-classifier
set class-of-service interfaces ge-0/0/3 unit 0 scheduler-map SCHEDULER-MAP
set class-of-service interfaces ge-0/0/3 unit 0 shaping-rate 2k
set class-of-service scheduler-maps SCHEDULER-MAP forwarding-class ef scheduler voice
```

```
set class-of-service schedulers voice buffer-size temporal 5k
set class-of-service schedulers voice drop-profile-map loss-priority any protocol any drop-
profile drop_profile
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure BA Classifiers for a device in PMI:

1. Configure the CoS.

```
[edit]
user@host# edit class-of-service
```

2. Configure BA classifiers for Differentiated Services (DiffServ) CoS.

```
[edit class-of-service]
user@host# edit classifiers dscp ba-classifier
```

3. Configure a BE forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class be loss-priority low code-points be
```

4. Configure an EF class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class ef-class loss-priority low code-points ef
user@host# set forwarding-class ef-class loss-priority high code-points af41
user@host# set forwarding-class ef-class loss-priority high code-points af11
user@host# set forwarding-class ef-class loss-priority high code-points af31
user@host# set forwarding-class low_delay loss-priority low code-points af21
user@host# set forwarding-class low_loss loss-priority low code-points cs6
```

5. Configure drop profiles.

```
[edit class-of-service drop-profiles]
user@host# set drop_profile fill-level 20 drop-probability 50
user@host# set drop_profile fill-level 50 drop-probability 100
```

6. Configure the forwarding classes queues.

```
[edit class-of-service forwarding-classes ]
user@host# set queue 0 be
user@host# set queue 1 ef
user@host# set queue 2 low_delay
user@host# set 3 low_loss
```

7. Apply the classifier to the interfaces.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/1 unit 0 classifiers dscp ba-classifier
user@host# set interfaces ge-0/0/3 unit 0 scheduler-map SCHEDULER-MAP
user@host# set interfaces ge-0/0/3 unit 0 shaping-rate 2k
```

8. Configure the schedulers.

```
[edit class-of-service]
user@host# set scheduler-maps SCHEDULER-MAP forwarding-class ef scheduler voice
user@host# set schedulers voice buffer-size temporal 5k
user@host# set schedulers voice drop-profile-map loss-priority any protocol any drop-profile
drop_profile
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
classifiers {
```

```
    dscp ba-classifier {
        forwarding-class be {
            loss-priority low code-points be;
        }
        forwarding-class ef {
            loss-priority low code-points ef;
            loss-priority high code-points [ af41 af11 af31 ];
        }
        forwarding-class low_delay {
            loss-priority low code-points af21;
        }
        forwarding-class low_loss {
            loss-priority low code-points cs6;
        }
    }
}
drop-profiles {
    drop_profile {
        fill-level 20 drop-probability 50;
        fill-level 50 drop-probability 100;
    }
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 low_delay;
    queue 3 low_loss;
}
interfaces {
    ge-0/0/1 {
        unit 0 {
            classifiers {
                dscp ba-classifier;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            scheduler-map SCHEDULER-MAP;
            shaping-rate 2k;
        }
    }
}
```

```
scheduler-maps {
    SCHEDULER-MAP {
        forwarding-class ef scheduler voice;
    }
}
schedulers {
    voice {
        buffer-size temporal 5k;
        drop-profile-map loss-priority any protocol any drop-profile drop_profile;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying the Classifier Application to the Interfaces | 1258

To confirm that the configuration is working properly, perform these tasks:

Verifying the Classifier Application to the Interfaces

Purpose

Verify that you've properly configured the classifier, and confirm the forwarding classes configuration.

Action

From the operational mode, enter the `show class-of-service forwarding-class` command.

```
user@host> show class-of-service forwarding-class
Forwarding class                          ID      Queue  Restricted queue  Fabric priority
Policing priority   SPU priority
  be                                       0       0          0             low
normal            low
  ef                                       1       1          1             low
```

```
normal           low
  low_delay                        2       2          2            low
normal           low
  low_loss                         3       3          3            low
normal           low
```

Meaning

The output shows the configured custom classifier settings.

Example: Configuring and Applying a Firewall Filter for a Multifield Classifier in PMI

IN THIS SECTION

- Requirements | **1259**
- Overview | **1260**
- Configuration | **1260**
- Verification | **1265**

This example shows how to configure a firewall filter to classify traffic to different forwarding class by using DSCP value and multifield (MF) classifier in PMI.

The classifier detects packets of interest to class of service (CoS) as they arrive on an interface. MF classifiers are used when a simple behavior aggregate (BA) classifier is insufficient to classify a packet, when peering routers do not have CoS bits marked, or the peering router's marking is untrusted.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall.
- Junos OS Release 19.1R1 and later releases.

Before you begin:

- Determine the forwarding class that are assigned by default to each well-known DSCP that you want to configure for the MF classifier. See "Improving IPsec Performance with PowerMode IPsec" on page 1241.

Overview

This example explain how to configure the firewall filter `mf-classifier`. To configure the MF classifier, create and name the assured forwarding traffic class, set the match condition, and then specify the destination address as 192.168.44.55. Create the forwarding class for assured forwarding DiffServ traffic as `af-class` and set the loss priority to low.

In this example, create and name the expedited forwarding traffic class and set the match condition for the expedited forwarding traffic class. Specify the destination address as 192.168.66.77. Create the forwarding class for expedited forwarding DiffServ traffic as `ef-class` and set the policer to `ef-policer`. Create and name the network-control traffic class and set the match condition.

In this example, create and name the forwarding class for the network control traffic class as `nc-class` and name the forwarding class for the best-effort traffic class as `be-class`. Finally, apply the multifield classifier firewall filter as an input and output filter on each customer-facing or host-facing that needs the filter. In this example, the interface for input filter is ge-0/0/2 and interface for output filter is ge-0/0/4.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1260**
- Procedure | **1261**
- Results | **1263**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set firewall filter mf-classifier interface-specific
set firewall filter mf-classifier term assured-forwarding from destination-address 192.168.44.55
set firewall filter mf-classifier term assured-forwarding then forwarding-class af-class
set firewall filter mf-classifier term assured-forwarding then loss-priority low
```

```
set firewall filter mf-classifier term expedited-forwarding from destination-address
192.168.66.77
set firewall filter mf-classifier term expedited-forwarding then forwarding-class ef-class
set firewall filter mf-classifier term expedited-forwarding then policer ef-policer
set firewall filter mf-classifier term network-control from precedence net-control
set firewall filter mf-classifier term network-control then forwarding-class nc-class
set firewall filter mf-classifier term best-effort then forwarding-class be-class
set interfaces ge-0/0/2 unit 0 family inet filter input mf-classifier
set interfaces ge-0/0/4 unit 0 family inet filter output mf-classifier
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure a Firewall Filter for a Multifield Classifier for a device in PMI:

1. Create and name the multifield classifier filter.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# set interface-specific
```

2. Create and name the term for the assured forwarding traffic class.

```
[edit firewall filter mf-classifier]
user@host# edit term assured-forwarding
```

3. Specify the destination address for assured forwarding traffic.

```
[edit firewall filter mf-classifier term assured-forwarding]
user@host# set from destination-address 192.168.44.55
```

4. Create the forwarding class and set the loss priority for the assured forwarding traffic class.

```
[edit firewall filter mf-classifier term assured-forwarding]
user@host# set then forwarding-class af-class
user@host# set then loss-priority low
```

5. Create and name the term for the expedited forwarding traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term expedited-forwarding
```

6. Specify the destination address for the expedited forwarding traffic.

```
[edit firewall filter mf-classifier term expedited-forwarding]
user@host# set from destination-address 192.168.66.77
```

7. Create the forwarding class and apply the policer for the expedited forwarding traffic class.

```
[edit firewall filter mf-classifier term expedited-forwarding]
user@host# set then forwarding-class ef-class
user@host# set then policer ef-policer
```

8. Create and name the term for the network control traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term network-control
```

9. Create the match condition for the network control traffic class.

```
[edit firewall filter mf-classifier term network-control]
user@host# set from precedence net-control
```

10. Create and name the forwarding class for the network control traffic class.

```
[edit firewall filter mf-classifier term network-control]
user@host# set then forwarding-class nc-class
```

11. Create and name the term for the best-effort traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term best-effort
```

12. Create and name the forwarding class for the best-effort traffic class.

```
[edit firewall filter mf-classifier term best-effort]
user@host# set then forwarding-class be-class
```

13. Apply the multifield classifier firewall filter as an input filter.

```
[edit]
user@host# set interfaces ge-0/0/2 unit 0 family inet filter input mf-classifier
```

14. Apply the multifield classifier firewall filter as an output filter.

```
[edit]
user@host# set interfaces ge-0/0/4 unit 0 family inet filter output mf-classifier
```

Results

From configuration mode, confirm your configuration by entering the `show firewall filter mf-classifier` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show firewall filter mf-classifier
interface-specific;
    term assured-forwarding {
    from {
```

```
        destination-address {
            192.168.44.55/32;
        }
    }
    then {
        loss-priority low;
        forwarding-class af-class;
    }
}
term expedited-forwarding {
    from {
        destination-address {
            192.168.66.77/32;
        }
    }
    then {
        policer ef-policer;
        forwarding-class ef-class;
    }
}
term network-control {
    from {
        precedence net-control;
    }
    then forwarding-class nc-class;
}
term best-effort {
    then forwarding-class be-class;
}
```

From configuration mode, confirm your configuration by entering the `show interfaces` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/2 {
    unit 0 {
        family inet {
            filter {
                input mf-classifier;
            }
```

```
        }
    }
}
ge-0/0/4 {
    unit 0 {
        family inet {
            filter {
                output mf-classifier;
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying a Firewall Filter for a Multifield Classifier Configuration | **1265**

To confirm that the configuration is working properly, perform these tasks:

Verifying a Firewall Filter for a Multifield Classifier Configuration

Purpose

Verify that a firewall filter for a multifield classifier is configured properly on a device and confirm that the forwarding classes are configured correctly.

Action

From configuration mode, enter the `show class-of-service forwarding-class` command.

```
user@host> show class-of-service forwarding-class
Forwarding class                      ID      Queue  Restricted queue  Fabric priority
Policing priority   SPU priority
  BE-data                              0       0          0             low
```

```
normal           low
  Premium-data                    1       1          1            low
normal           low
  Voice                           2       2          2            low
normal           low
  NC                              3       3          3            low
normal           low
```

Meaning

The output shows the configured custom classifier settings.

Example: Configuring and Applying Rewrite Rules on a Security Device in PMI

IN THIS SECTION

- Requirements | 1266
- Overview | 1267
- Configuration | 1267
- Verification | 1270

This example shows how to configure and apply rewrite rules for a device in PMI.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall.
- Junos OS Release 19.1R1 and later releases.

Before you begin:

- Create and configure the forwarding classes. See "Improving IPsec Performance with PowerMode IPsec" on page 1241.

Overview

This example explains how to configure rewrite rules to replace CoS values on packets received from the customer or host with the values expected by other SRX Series Firewalls. You do not have to configure rewrite rules if the received packets already contain valid CoS values. Rewrite rules apply the forwarding class information and packet loss priority used internally by the device to establish the CoS value on outbound packets. After you configure the rewrite rules, apply them to the correct interfaces.

In this example, configure the rewrite rule for DiffServ CoS as `rewrite-dscps`. Specify the best-effort forwarding class as `be-class`, expedited forwarding class as `ef-class`, an assured forwarding class as `af-class`, and a network control class as `nc-class`. Finally, apply the rewrite rule to the ge-0/0/0 interface.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1267**
- Procedure | **1268**
- Results | **1269**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class loss-priority
low code-point 000000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class loss-priority
high code-point 000001
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
low code-point 101110
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
high code-point 101111
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
low code-point 001010
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
high code-point 001100
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
```

```
low code-point 110000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
high code-point 110001
set class-of-service interfaces ge-0/0/0 unit 0 rewrite-rules dscp rewrite-dscps
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure and apply Rewrite Rules for a device in PMI:

1. Configure rewrite rules for DiffServ CoS.

```
[edit]
user@host# edit class-of-service
user@host# edit rewrite-rules dscp rewrite-dscps
```

2. Configure best-effort forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class be-class loss-priority low code-point 000000
user@host# set forwarding-class be-class loss-priority high code-point 000001
```

3. Configure expedited forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class ef-class loss-priority low code-point 101110
user@host# set forwarding-class ef-class loss-priority high code-point 101111
```

4. Configure an assured forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class af-class loss-priority low code-point 001010
user@host# set forwarding-class af-class loss-priority high code-point 001100
```

5. Configure a network control class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class nc-class loss-priority low code-point 110000
user@host# set forwarding-class nc-class loss-priority high code-point 110001
```

6. Apply rewrite rules to an interface.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/0  unit 0 rewrite-rules dscp rewrite-dscps
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
interfaces {
    ge-0/0/0 {
        unit 0 {
            rewrite-rules {
                dscp rewrite-dscps;
            }
        }
    }
}
rewrite-rules {
    dscp rewrite-dscps {
        forwarding-class be-class {
            loss-priority low code-point 000000;
            loss-priority high code-point 000001;
        }
        forwarding-class ef-class {
            loss-priority low code-point 101110;
            loss-priority high code-point 101111;
        }
        forwarding-class af-class {
            loss-priority low code-point 001010;
```

```
            loss-priority high code-point 001100;
        }
        forwarding-class nc-class {
            loss-priority low code-point 110000;
            loss-priority high code-point 110001;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Rewrite Rules Configuration | 1270

To confirm that the configuration is working properly, perform these tasks:

Verifying Rewrite Rules Configuration

Purpose

Verify that rewrite rules are configured properly.

Action

From the operational mode, enter the `show class-of-service` command.

```
user@host> show class-of-service
Physical interface: ge-0/0/0, Index: 130
 Maximum usable queues: 8, Queues in use: 4
Scheduled map: <default>, Index:2
Congestion-notification: Disabled

LOgical interface: ge0/0/0, Index: 71
Object      Name                   Type    Index
Classifier   ipprec-compatibility  ip    13
```

Meaning

Rewrite rules are configured on ge-0/0/0 interface as expected.

Configure IPsec ESP Authentication-only Mode in PMI

The PMI introduced a new data path for achieving a high IPsec throughput performance. You can use Encapsulating Security Payload (ESP) authentication-only mode in PMI mode, which provides authentication, integrity checking, and replay protection without encrypting the data packets.

Before you begin:

- Make sure that the session is PMI capable. See "VPN Session Affinity " on page 1232.

To configure ESP authentication-only mode:

1. Configure IPsec proposal and policy.

```
user@host# set security ipsec proposal IPSEC_PROP protocol esp
user@host# set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@host# set security ipsec policy IPSEC_POL proposals IPSEC_PROP
```

2. Confirm your configuration by entering the `show security ipsec` command.

```
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
}
policy IPSEC_POL {
    proposals IPSEC_PROP;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

SEE ALSO

proposal

Platform-Specific PMI Behavior

Use Feature Explorer to confirm platform and release support for specific features.

See the "Additional Platform Information" on page 1273 section for more information.

Use the following table to review platform-specific behaviors for your platforms.

Table 128: Platform-Specific Behavior

Platform	Difference

MX Series with MX-SPC3	• On MX Series that support PMI, here's a list of supported and non-supported features in addition to the features in "Supported and Non-Supported Features for PMI" on page 1241 :
	  • List of supported features:
	  • ADVPN
	  • DPD
	  • Antireplay check
	  • Post/Pre-Fragment
	  • Incoming cleartext fragments and ESP fragment
	  • List of non-supported features:
	  • np-cache and IPsec session-affinity

Table 128: Platform-Specific Behavior *(Continued)*

Platform	Difference
SRX Series	• On SRX Series that support PMI, here's a list of supported and non-supported features in addition to the features in "Supported and Non-Supported Features for PMI" on page 1241: • List of supported features: • NAT-T (SRX5K-SPC3 and vSRX Virtual Firewall) • GTP-U scenario with TEID distribution and asymmetric fat tunnel solution • QoS • First path and fast path processing for fragment handling and unified encryption. • List of non-supported features: • IPsec-in-IPsec tunnels • GPRS tunneling protocol (GTP) and Stream Control Transmission Protocol (SCTP) firewalls • Host traffic • DES-CBC encryption algorithm • 3DES-CBC encryption algorithm

Additional Platform Information

Use Feature Explorer to confirm platform and release support for specific features. Additional Platforms may be supported. Review the "Platform-Specific PMI Behavior" on page 1272 section for notes related to your platform.

Table 129: Additional Platform Information for PMI Enable/Disable

Feature	SRX4100 SRX4200 SRX4600	SRX5000 Line with SPC3	vSRX Virtual Firewalls
Reboot required after PMI enabled or disabled	Yes	No	No for Junos OS Release 19.2R1 or later Yes for Junos OS Release 18.3R1

Table 130: Additional Platform Information for PMI Support

PMI Support	SRX1500	SRX1600 SRX2300	SRX4100 SRX4120	SRX4200 SRX4600	SRX4300	SRX4700	SRX5000 Line with SPC3	vSRX 3.0
Default	NA	23.4R1 and later	SRX4100: 21.3R1 or later SRX4120: 25.2R1 and later	21.3R1 or later	24.2R1 and later	24.4R1-S2 and later	21.3R1 or later	21.3R1 or later
Optional	NA	NA	SRX4100: 19.1R1 and later	SRX4600: 20.4R1 and later SRX4200: 19.1R1 and later	NA	NA	19.1R1 and later	20.1R1 and later

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.4R2	Support for PMI processing added for HMAC-SHA-384 and HMAC-SHA-512 authentication algorithm.
24.2R1	Support for PMI processing added for ChaCha20-Poly1305 authenticated encryption algorithm.
22.1R3	Support for PMI express path processing for passthrough ESP traffic added in SRX Series Firewalls.
21.1R1	Support for fat IPsec tunnel added in MX-SPC3 services card.
20.4R1	Support for AES-NI added to PMI.
20.4R1	Support for PMI added in SRX4600.
19.4R1	Support for fat IPsec tunnel introduced in SRX5K-SPC3 and vSRX Virtual Firewall.
19.4R1	Support added for per-flow CoS functions for GTP-U in PMI mode on vSRX Virtual Firewalls.
19.4R1	Support for Encapsulating Security Payload (ESP) authentication only mode in PMI added in SRX5K-SPC3.
19.3R1	Support added for aes-128-cbc, aes-192-cbc, and aes-256-cbc in PMI mode on SRX4100, SRX4200, and vSRX Virtual Firewall.
19.3R1	PMI support added for Network Address Translation (NAT).
19.2R1	Support added for per-flow CoS support for GTP-U in PMI mode on SRX5K-SPC3.
19.2R1	Support added for PMI for NAT-T on SRX5K-SPC3 and vSRX Virtual Firewalls.
19.1R1	Support for CoS classifier and rewrite functions in PMI introduced in SRX5K-SPC3.
19.1R1	Support for PMI added in SRX5K-SPC3.
18.4R1	Support for PMI added in SRX4100 and SRX4200.
18.3R1	Support for PMI added in vSRX Virtual Firewalls.

13

CHAPTER

Troubleshooting

IN THIS CHAPTER

- Troubleshoot a Flapping VPN Tunnel | **1277**
- Troubleshoot a VPN That Is Up But Not Passing Traffic | **1280**
- Troubleshoot a VPN Tunnel That is Down | **1284**
- Troubleshoot Issues in Quantum Safe IPsec VPN | **1286**
- How to Analyze IKE Phase 2 VPN Status Messages | **1300**

Troubleshoot a Flapping VPN Tunnel

IN THIS SECTION

- Problem | 1277
- Diagnosis | 1277

Problem

Description

Site-to-site VPN tunnel or remote IPsec VPN tunnel is flapping (that is, going up and down in quick succession).

Diagnosis

1. Does the issue affect only one VPN?

   - Yes: Check the system logs and proceed to Step 2. Use the `show log messages` command to view the logs. You must enable information-level logging for messages to be reported correctly.

     ```
     user@host # set system syslog file messages any info
     ```

     Here are examples of system logs reporting a flapping VPN tunnel:

     VPN up/down events:

     ```
     Jul 9 21:07:58 kmd[1496]: KMD_VPN_DOWN_ALARM_USER: VPN to_hub from 3.3.3.2 is down. Local-
     ip: 4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if:
     st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH
     username: Not-Applicable, VR id: 4
     Jul 9 21:08:10 kmd[1496]: KMD_VPN_UP_ALARM_USER: VPN to_hub from 3.3.3.2 is up. Local-ip:
     4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if:
     st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH
     username: Not-Applicable, VR id: 4
     Jul 9 21:09:58 kmd[1496]: KMD_VPN_DOWN_ALARM_USER: VPN to_hub from 3.3.3.2 is down. Local-
     ip: 4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if:
     ```

```
st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH
username: Not-Applicable, VR id: 4
Jul 9 21:10:10 kmd[1496]: KMD_VPN_UP_ALARM_USER: VPN to_hub from 3.3.3.2 is up. Local-ip:
4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if:
st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH
username: Not-Applicable, VR id: 4
```

Unstable VPN behavior (VPN constantly rebuilding):

```
Jul 9 20:43:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: inbound, SPI: 0xfd91b643, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:43:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: outbound, SPI: 0xbdec9669, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:44:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: inbound, SPI: 0x69b34ae4, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:44:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: outbound, SPI: 0x6f55d8ea, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:45:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: inbound, SPI: 0x6fa6b0b3, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:45:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: outbound, SPI: 0xa66ac906, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
```

- No: If the issue is on all configured VPNs, investigate the errors associated with the Internet connection, and on the firewall and switch interfaces. To check for errors on the firewall interface, run the `show interfaces extensive` command.

2. Verify that VPN Monitor is enabled for this VPN by using the `show configuration security ipsec vpn` *vpn-name* command.

   Is VPN Monitor enabled?

- Yes: Proceed to Step 3.
- No: Proceed to Step 5.

3. Disable VPN Monitor and check the VPN.

```
user@host# deactivate security ipsec vpn vpn-name vpn-monitor
user@host# commit
```

   Is the VPN stable?

   - Yes: The instability is related to the VPN Monitor configuration. Proceed to Step 4.
   - No: Proceed to Step 5.

4. Is the remote VPN connection configured to block ICMP echo requests?

   - Yes: Reenable and reconfigure VPN Monitor to use the source interface and destination IP options. See KB10119.
   - No: Proceed to Step 5.

5. Is the remote device that is connected to the firewall a non-Juniper device?

   - Yes: Verify the *proxy-id* value on the firewall and the peer VPN device.
   - No: Proceed to Step 6.

6. Was the VPN stable for a period of time and then started going up and down?

   - Yes: Investigate for network or device changes or whether any new network equipment has been added to the environment.
   - No: Collect site-to-site logs from the VPN devices at both ends and open a case with your technical support representative. See Data Collection for Customer Support.

Troubleshoot a VPN That Is Up But Not Passing Traffic

IN THIS SECTION

- Problem | 1280
- Solution | 1280

Problem

Description

The VPN is up, but there is no passing traffic in one or both directions.

This topic helps troubleshoot the issues that could prevent traffic passing through an active VPN tunnel.

Environment

VPN

Solution

1. Check whether the VPN security association (SA) is active: **show security ipsec security-associations**

```
user@CORPORATE> show security ipsec security-associations
  total configured sa: 1
  ID     Gateway         Port  Algorithm       SPI      Life:sec/kb  Mon vsys
  <32785 2.2.2.2         1398  ESP:3des/sha1   29e26eba 28735/unlim   -   0
  >32785 2.2.2.2         1398  ESP:3des/sha1   6d4e790b 28735/unlim   -   0
```

   If the VPN gateway is listed, the tunnel is established and is up. The output displays two lines for each VPN tunnel displaying the SPI information for each direction of traffic.

   The `MON` field is used by VPN monitoring to show the status of the tunnel and has one of the following values:

- **-** (hyphen): The VPN tunnel is active, and the VPN monitor optional feature is not configured.
- **U** (up): The VPN tunnel is active, and the link (detected through the VPN monitor) is up.
- **D** (down): The VPN tunnel is active, and the link (detected through the VPN monitor) is down.
- **Yes**: The IPsec SA state is active or up. Proceed to Step "2" on page 1281.
- **No**: The IPsec SA state is down. See How to troubleshoot a VPN tunnel that is down or not active.

2. Check whether the VPN is using the loopback interface lo0 as the external interface: **show configuration security ike**

```
root> show configuration security ike
policy ike_pol {
   proposal-set compatible;
   pre-shared-key ascii-text "$9$tMwDuIESreWX7yr4aGDkqIEhcvWbs2";
}
gateway gate1 {
  ike-policy ike_pol;
  address 10.10.10.2;
  external-interface lo0.0;
}
```

- **Yes**: VPN is using the loopback interface **lo0** as the external interface. Proceed to Step "3" on page 1281.
- **No**: VPN is not using the loopback interface **lo0** as the external interface. Proceed to Step "4" on page 1281.

3. Check whether the egress interface (physical interface) and lo0 used as the VPN external interface are in the same security zone.
   - **Yes**: Proceed to Step "4" on page 1281.
   - **No**: Update the security zone assignments so that both the VPN external interface and the physical egress interface are in the same security zone. See Traffic Loss when IPSec VPN is terminated on loopback interface.
4. If your VPN is a route-based VPN, proceed to Step "5" on page 1282. See What is the difference between a policy-based VPN and a route-based VPN?

5. Check whether a route is assigned to the remote network through the st0 interface: **show route *remote network***

```
root@siteA > show route 192.168.20.10
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.2.0/24    *[ARI-TS/5] 00:00:53
                          > via st0.0  <----------
```

- **Yes**: Proceed to Step "6" on page 1282.
- **No**: Assign a route to the remote network through the st0 interface. See Route-based VPN is up, but not passing traffic. Is a route missing?.

> **NOTE**: If you are using a dynamic routing protocol, such as BGP or OSPF, then check the routing protocol.

6. Based on the route assigned to the remote network in Step "5" on page 1282, check whether the VPN is pointing to the correct st0 interface: **show security ike** and **show security ipsec**

   a. First, check the IKE gateway using the **show security ike** command.

```
root@siteA # show security ike
...
gateway gw-siteB {        <---------
     ike-policy ike-phase1-policy;
     address 2.2.2.2;
     external-interface ge-0/0/3.0;
}
```

   b. Check the IPsec VPN for that IKE gateway using the **show security ipsec** command and in the output verify if `bind-interface` is pointing to `st0` interface.

   In this example, the VPN `ike-vpn-siteB` is pointing to the `st0.0` interface.

```
root@siteA # show security ipsec
...
vpn ike-vpn-siteB {
    bind-interface st0.0;
      ike {
```

```
          gateway gw-siteB;      <---------
          proxy-identity {
              local 192.168.2.0/24;
              remote 192.168.1.0/24;
              service any;
            }
           ipsec-policy ipsec-phase2-policy;
         }
      establish-tunnels immediately;
     }
```

- **Yes**: Proceed to Step "7" on page 1283.
- **No**: VPN is not pointing to the correct st0 interface. Delete the current route, and add the route to the correct st0 interface. See Route-based VPN is up, but not passing traffic. Is a route missing?.

7. Check whether there is a security policy that allows traffic from the internal zone to the st0 security zone: **show security policies**
   - **Yes**: Check if the configured policy is correct.
   - **No**: Create the appropriate security policy and test the VPN again. See How to configure a policy for a route-based VPN.
8. Check whether the traffic is matching in the policies identified in step "7" on page 1283 : **show security flow session source prefix *source address* destination prefix *destination address***

```
root@siteA> show security flow session source-prefix 192.168.2.0/24 destination-prefix
192.168.1.0/24

Session ID: 5801, Policy name: AtoB/2, Timeout: 1790, Valid
In: 192.168.2.222/1 --> 192.168.1.13/23053;icmp, If: fe-0/0/2.0, Pkts: 59878, Bytes: 4602292
Out: 192.168.1.13/23053 --> 192.168.2.222/1;icmp, If: st0.0, Pkts: 52505, Bytes: 4189289
```

   - **Yes**: Proceed to Step "9" on page 1284.
   - **No**: Verify the order of the security policies: **show security match policies**. See Understanding Security Policy Ordering.

     If the order is correct, see How to troubleshoot a security policy that is not passing data.

> **NOTE**: If only the `pkts` counter in the out direction of the session is incrementing, then validate with the VPN peer that the traffic is being received.
>
> This is to check the packet counters on the VPN peer with which this tunnel is formed to see whether the other end is receiving the packets.

9. Collect logs and flow trace options and open a case with the Juniper Networks support team:

   - See the IPsec VPN route-based VPN sections in Data Collection Checklist - Logs/data to collect for troubleshooting.
   - For information regarding flow trace options, see How to use 'flow traceoptions' and the 'security datapath-debug'.
   - To open a JTAC case with the Juniper Networks support team, see Data Collection for Customer Support for the data you should collect to assist in troubleshooting before opening a JTAC case.

Troubleshoot a VPN Tunnel That is Down

Problem: IPsec VPN is not active and does not pass data.

1. What type of VPN tunnel are you having trouble with?

   - Site-to-site (LAN-to-LAN) VPN:

     Proceed to Step 2.

   - Remote Access IPsec VPN:

     Proceed to Step 2.

     See KB17220.

2. Is the SA (security association) for the VPN tunnel active?

   Run the `show security ipsec security-associations` command and locate the gateway address of the VPN. If the remote gateway is not displayed, then the VPN SA is not active. For more information about SA, see KB10090.

```
user@host> show security ipsec security-associations
   total configured sa: 2
   ID     Gateway         Port  Algorithm       SPI      Life:sec/kb  Mon vsys
   <32785 2.2.2.2         1398  ESP:3des/sha1   29e26eba 28735/unlim   -   0
```

```
>32785 2.2.2.2          1398  ESP:3des/sha1   6d4e790b 28735/unlim   -   0
total configured sa: 2
ID     Gateway          Port  Algorithm       SPI      Life:sec/kb  Mon vsys
<32786 3.3.3.3          500   ESP:3des/sha1   5c13215d 28782/unlim   U   0
>32786 3.3.3.3          500   ESP:3des/sha1   18f67b48 28782/unlim   U   0
```

- If SA is not listed in the output, proceed to Step 3.
- If SA is listed (Phase 2 is up) and if traffic is not passing, see "Troubleshoot a VPN That Is Up But Not Passing Traffic" on page 1280.
- If SA oscillates between active and inactive states, see "Troubleshoot a Flapping VPN Tunnel" on page 1277.

3. Is the IKE Phase 1 up?

   Run the `show security ike security-associations` command. Verify that the remote address of the VPN is listed and that the value of the `State` field is UP.

```
user@host> show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
1       2.2.2.2         UP     744a594d957dd513  1e1307db82f58387  Main
2       3.3.3.3         UP     744a594d957dd513  1e1307db82f58387  Main
```

   - If the remote address is not listed or if the value of the `State` field is `DOWN`, analyze the IKE Phase 1 messages on the responder for a solution. See KB10101.
   - If the state is `UP`, analyze the IKE Phase 2 messages on the responder for a solution. See KB10101.

   If the issue is still not resolved, analyze Phase 1 or Phase 2 logs for the VPN tunnel on the initiating VPN device. If you can't find your solution in the logs on the initiating side, proceed to Step 4.

4. Collect logs, flow trace options, and IKE trace options, and then open a case with your technical support representative. For information about:

   - Collecting logs, see Data Collection for Customer Support.
   - Flow trace options, see KB16233.
   - IKE trace options, see KB19943.

Troubleshoot Issues in Quantum Safe IPsec VPN

SUMMARY

Read this topic to troubleshoot issues in CA certificates, retrieving keys, or establishing tunnels, along with relevant solutions for quantum safe IPsec VPN.

IN THIS SECTION

- Troubleshooting Summary | **1287**
- Troubleshoot CA Certificate Load Failure Due to Multiple Certificates in a PEM File | **1289**
- Troubleshoot QKD Key Retrieval Failure Due to Issues in Certificate Chain Validation | **1291**
- Troubleshoot Key Retrieval Failure Due to SAE Mismatch | **1294**
- Troubleshoot QKD Key Retrieval Failure Due to KME Server Error | **1296**
- Other Common Issues in Quantum Safe IPsec VPN | **1298**

Use this topic to diagnose and resolve certificate, Transport Layer Security (TLS), quantum key retrieval, and tunnel establishment issues in Quantum Safe IPsec VPN deployments.

This topic applies to deployments that use:

- A firewall providing Quantum Safe IPsec VPN service
- Quantum Key Distribution (QKD)
- ETSI QKD APIs
- X.509 certificates with certificate authority (CA) validation
- Public key infrastructure (PKI) certificate chains with root and intermediate CAs
- RSA (2048-bit, 4096-bit) and ECDSA cryptographic keys

In QKD deployments, the firewall acts as a Key Management Entity (KME) client and establishes HTTPS connections to the KME server. Before retrieving quantum keys, the firewall validates the KME server certificate chain during the TLS handshake.

To collect diagnostic information, enable PKI and Junos Key Manager (JKM) logging on the firewall:

- Use PKI tracing for certificate import, certificate-chain validation, and certificate-related TLS failures.

1. Configure PKI traceoptions to enable logging.

```
set security pki traceoptions
set file pkid-debug size 5m files 5
set flag all
commit
```

2. Review the log for errors.

```
show log pkid-debug | match "error|fail"
```

- Use JKM tracing for QKD or Key Management Entity (KME) connectivity and key-retrieval failures. JKM tracing writes application logs to `/var/log/jkm`.

1. Configure JKM traceoptions to enable logging.

```
set security key-manager traceoptions
set file jkm-debug size 5m files 5
set flag all
commit
```

2. Review the log for errors.

```
show log jkm-debug | match "HTTP|UI|OPS|RES|TIME|error|fail"
```

Troubleshooting Summary

Review the summary table for an overview of the specific failures discussed in this topic:

Table 131: Specific Issues in Quantum Safe IPsec VPN

Issue Type	Root Cause	Solution
Certificate and PKI failures	• Certificate files that use carriage return line feed (CRLF) formatting. • Incorrect PEM file structure having multiple certificates in a PEM file • Missing intermediate CA certificates. • Duplicate certificate identities.	• Convert certificate files to UNIX line feed (LF) format. • Store each certificate in a separate PEM file. • Configure the complete CA chain. See "Troubleshoot CA Certificate Load Failure Due to Multiple Certificates in a PEM File" on page 1289.
TLS connection failures	Certificate chain validation fails with HTTP status code 0 or curl response code 60 or the KME server doesn't respond.	Verify the TLS certificate-verification failure. Check hostname mismatches, certificate validation dates, certificate chain details, CA certificate configuration. See "Troubleshoot QKD Key Retrieval Failure Due to Issues in Certificate Chain Validation " on page 1291.
Secure application entity (SAE) identifier mismatch	Firewall SAE identifiers do not match KME SAE identifiers. Although TLS authentication succeeds, ETSI QKD API requests fail to retrieve keys.	Configure matching `local-sae-id` and `peer-sae-ids` values on the firewall and the KME. Verify successful key retrieval using ETSI QKD API calls. See "Troubleshoot Key Retrieval Failure Due to SAE Mismatch" on page 1294.

Table 131: Specific Issues in Quantum Safe IPsec VPN *(Continued)*

Issue Type	Root Cause	Solution
QKD key retrieval failures	• Key expiration before retrieval. • Timing issues between peers (initiator and responder). • KME configuration errors. • Request rate limiting.	• Retrieve keys within the required validity period. • Validate KME configuration for key availability and key pairing. • Reduce request frequency or adjust KME rate limits. See "Troubleshoot QKD Key Retrieval Failure Due to KME Server Error" on page 1296.

Troubleshoot CA Certificate Load Failure Due to Multiple Certificates in a PEM File

IN THIS SECTION

- Problem | 1289
- Solution | 1290

Problem

Description

The firewall reads the certificate successfully but fails to load it into a CA profile. The issue occurs during PKI operations when loading a CA certificate into a CA profile.

Symptoms

Symptoms include:

- The following command displays certificate details but does not load the certificate:

```
request security pki ca-certificate load ca-profile CA_PROFILE filename /tmp/certs/ca.pem
```

  The command returns the following error:

```
error: Failed to write the CA certificate to local store
```

- A single PEM file contains multiple certificates. The following output appears when you run the command `cat /tmp/certs/ca.pem`.

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

  The PEM file contains multiple certificates. Junos OS requires one certificate per PEM file when loading a CA certificate into a CA profile.

- Multiple certificates in the chain use the same subject name. The firewall uses the certificate subject name as a unique identifier. If two certificates have the same subject name, the firewall treats the second certificate as a duplicate and does not load it. See [SRX] SRX cannot load two certificates with same Subject at once.

- The certificate file contains Windows-style CRLF endings, such as \r\n, which can cause PEM parsing issues on the firewall.

Solution

Perform the following steps to resolve the issue:

1. Ensure that each PEM file contains only one certificate. Avoid combining root and intermediate CA certificates in the same file.

```
-----BEGIN CERTIFICATE-----
(single certificate only)
-----END CERTIFICATE-----
```

Use separate files, such as *root-ca.pem* and *int-ca.pem*.

2. Verify that all certificate files use UNIX line feed (LF) line endings before loading the files onto the firewall.

3. Configure a separate CA profile for each certificate in the trust chain.

   a. Load the root and intermediate CA certificates separately.

```
request security pki ca-certificate load ca-profile ROOT_CA_PROFILE filename /var/tmp/
certs/root-ca.pem
request security pki ca-certificate load ca-profile INT_CA_PROFILE filename /var/tmp/certs/
int-ca.pem
```

   b. Verify that the certificates are loaded.

```
show security pki ca-certificate detail
```

   c. Verify each certificate individually.

```
request security pki ca-certificate verify ca-profile ROOT_CA_PROFILE
request security pki ca-certificate verify ca-profile INT_CA_PROFILE
```

Troubleshoot QKD Key Retrieval Failure Due to Issues in Certificate Chain Validation

IN THIS SECTION

- Problem | **1292**
- Solution | **1293**

Problem

Description

The firewall cannot retrieve QKD keys because TLS certificate validation of the KME server fails.

If certificate validation fails, the TLS handshake does not complete and the key retrieval request fails.

Symptoms

The quantum key manager profile configuration on the firewall is successful. The quantum key manager on the firewall establishes an HTTPS connection to communicate with the KME server for encryption key retrieval. The firewall validates the KME server's TLS certificate chain using the trusted CAs configured in the quantum key manager profile. However, the TLS connection to the KME server fails, preventing the firewall from retrieving keys from the KME.

The following operational command fails to retrieve keys:

```
request security key-manager profiles get profile-keys name KM_PROFILE_1
```

The following logs indicate TLS failure with HTTP status code 0 and curl response code 60.

```
Feb 24 18:40:36.303721 [DET] [HTTP] ca-cert (/var/db/certs/common/certification-authority/root-
ca.cert) added in connection
Feb 24 18:40:36.303782 [DET] [HTTP] http-ssl-setup: CA certs added
Feb 24 18:40:36.304122 [DET] [HTTP] http-ssl-setup: client-cert-chain added
Feb 24 18:40:36.304451 [DET] [HTTP] http-socket-callback: curl socket callback conn-ctx (2.0)
action (1)
Feb 24 18:40:36.732843 [DET] [HTTP] http-socket-callback: curl socket callback conn-ctx (2.0)
action (4)
Feb 24 18:40:36.733046 [DET] [HTTP] http-client-request-check-status: req-id (4) URL: https://
kme.dmz.example.com:443/api/v1/keys/juniper2/enc_keys DONE http-rsp-code (0) status-code (60)
status-str (SSL peer certificate or SSH remote key was not OK)
Feb 24 18:40:36.733111 [DET] [TIME] jkm_timer_wheel_stop_timer, stopped timer 4 cb 0x42d150 cbp
0x15c9d40, module 7,..
Feb 24 18:40:36.733148 [DET] [OPS ] op-done: status: (-4097) op_type (3) op_output_data
(0xffffdc08) op_ctx (0x1448360)
Feb 24 18:40:36.733180 [DET] [TIME] jkm_timer_wheel_stop_timer, stopped timer 4 cb 0x424cf0 cbp
0x1448360, module 5,..
Feb 24 18:40:36.733234 [ERR] [ CM ] cm-client-req-stats-decr: client with id (0) not found
Feb 24 18:40:36.733298 [DET] [ UI ] get-data-done: status: (-4097), op_type (3), client_id (0),
request_id (4) op_output_data (0xffffdaf0)
```

```
Feb 24 18:40:36.733728 [DET] [HTTP] http-process-rsp: req-id (4) conn-ctx (2.0) [DELETED]
Feb 24 18:40:36.733797 [DET] [ UI ] msg-pre-close-cb: msp-socket (0x1439540) (17) closed
```

The quantum key manager profile includes only the root CA:

```
show configuration security key-manager profiles KM_PROFILE_1
```

The HTTPS key request fails during the TLS handshake because the firewall cannot validate the KME server certificate chain.

This issue can occur when the KME server does not provide the required intermediate CA certificate and the intermediate CA certificate is not configured locally. It can also occur when the firewall policy requires the intermediate CA certificate in the local trusted CA store.

Solution

Determine whether the KME server provides the intermediate CA certificate during the TLS handshake. To resolve this issue, configure the firewall to trust the complete certificate chain as shown below:

1. If the KME server provides the intermediate certificate chain, configure the root CA certificate in the `trusted-cas` list. Root CA trust is sufficient when the KME server provides the required intermediate certificates.

   ```
   [edit]#In configuration mode#
   set security key-manager profiles KM_PROFILE_1 quantum-key-manager trusted-cas [ ca-root ]
   ```

   If the KME server does not provide the required intermediate certificate, or if the firewall policy requires local intermediate CA certificates, load the intermediate CA certificate and add it to the `trusted-cas` list.

   ```
   [edit]#In configuration mode#
   set security key-manager profiles KM_PROFILE_1 quantum-key-manager trusted-cas [ ca-root ca-
   intermediate ]
   ```

2. Verify that the key retrieval is successful after updating the configuration.

   ```
   #In operational mode#
   request security key-manager profiles get profile-keys name KM_PROFILE_1
   ```

Confirm the logs show a successful HTTP response code and status code 0.

For successful TLS communication, configure the root CA certificate as a trust anchor. Configure intermediate CA certificates locally only when the KME server does not provide them or when the firewall requires them.

Troubleshoot Key Retrieval Failure Due to SAE Mismatch

IN THIS SECTION

- Problem | 1294
- Solution | 1295

Problem

Description

TLS communication succeeds, but the firewall cannot retrieve quantum keys from the KME.

Symptoms

You observe the following behavior:

- From operational mode, enter the following command to view peer device's key manager profile and keys.

```
request security key-manager profiles get profile-keys name KM_PROFILE_1 peer-sae-id SRX02
```

The command fails. JKM requests one key by default. If the KME response does not return the requested number of keys, JKM treats the request as a failure. The operational command displays only the failed status:

```
- Response:
    - Status: FAILED
```

Review the JKM log for the KME HTTP response.

- From operational mode, enter the following command to view key manager profile details.

```
show security key-manager profiles name KM_PROFILE_1 detail
```

The command returns the following details. The output shows one failed request.

```
Name: KM_PROFILE_1, Index: 2, Type: quantum-key-manager
  Configured-at: 11.09.23 (02:04:32)
  Time-elapsed: 0 hrs 20 mins 23 secs
  Url: https://kme.juniper.net
  Local-sae-id: SRX01
  Local-certificate-id: SAE_A_CERT
  Trusted-cas: [ Root-CA ]
  Peer-sae-ids: N/A
  Default-key-size: N/A
  Request stats:
    Received: 0
    In-progress: 0
    Success: 0
    Failed: 1
```

Solution

Align the SAE identifiers on the firewall with the SAE identifiers on the KME. The firewall was initially configured with the SAE IDs SRX01 and SRX02 as local and peer SAE IDs. Perform the following steps to resolve the issue:

1. Update the firewall key manager profile to use the same SAE IDs configured on the KME. Use SAE_A and SAE_B as the local and peer SAE IDs.

```
set security key-manager profiles KM_PROFILE_1 quantum-key-manager local-sae-id SAE_A
set security key-manager profiles KM_PROFILE_1 quantum-key-manager peer-sae-ids SAE_B
set security key-manager profiles KM_PROFILE_1 quantum-key-manager url https://192.168.101.1
set security key-manager profiles KM_PROFILE_1 quantum-key-manager trusted-cas ROOT_CA_CERT
set security key-manager profiles KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_A_CERT
```

Key retrieval is successful after aligning the `local-sae-id` and `peer-sae-ids` values with the corresponding SAE IDs configured on the KME.

2. Alternatively, send a Get Key API request to the KME by using the firewall certificate and private key.

For ETSI QKD key retrieval to succeed, the SAE identifiers configured on the firewall must match the SAE identifiers defined on the KME.

Troubleshoot QKD Key Retrieval Failure Due to KME Server Error

IN THIS SECTION

- Problem | 1296
- Solution | 1297

Problem

Description

The firewall fails to retrieve QKD keys from the KME due to a server-side error. The problem is not related to Junos OS configuration or TLS validation, but to KME constraints. An HTTP 503 response confirms that the KME did not successfully fulfill the key-retrieval request.

Symptom

You observe the following:

The following command fails:

```
request security key-manager profiles get profile-keys-with-id name KM_PROFILE_1 peer-sae-id id1
key-id 12345678-1234-5678-1234-567812345678
```

The firewall sends a request to the KME and receives an HTTP 503 response. Log indicates that the firewall establishes TLS connection and validates the certificate chain. However, the KME server returns an HTTP response code 503, indicating a server-side error.

```
```
Feb 24 18:48:48.078293 [CRT] [UI] request_profile cli_params name KM_PROFILE_1 peer-sae-id id1
key-id 12345678-1234-5678-1234-567812345678
```
```

```
...
Feb 24 18:48:48.078373 [DET] [HTTP] http-process-req: req-id (3) processing scheduled: url:
(https://kme2.dmz.example.com:443/api/v1/keys/id1/dec_keys) post-data ({"key_IDs":
[{"key_ID":"12345678-1234-5678-1234-567812345678"}]})
...
Feb 24 18:48:48.260787 [DET] [HTTP] ca-cert (.../root-ca.cert) added in connection
Feb 24 18:48:48.261106 [DET] [HTTP] ca-cert (.../int-ca.cert) added in connection
...
Feb 24 18:48:49.671675 [DET] [HTTP] http-client-request-check-status: req-id (3) URL: https://
kme2.dmz.example.com:443/api/v1/keys/id1/dec_keys DONE http-rsp-code (503) status-code (0)
status-str (No error)
Feb 24 18:48:49.671753 [ERR] [RES ] kme-response-print-error-message:response msg: (Internal
server error.)
```

The internal server error message from the KME indicates that the issue occurs when the KME cannot provide the requested key within the allowed retrieval window. Review the KME logs to determine why the KME rejected or could not process the request. Common causes include:

- The key is no longer available on the responder-side KME.
- The allowed key retrieval window—typically less than 10 seconds—has expired.
- The KME does not have a matching key for the requested SAE pair.
- Rate limiting due to repeated or delayed requests.

In QKD workflows, the responder must retrieve the key shortly after the initiator. Delayed requests can cause the KME to return HTTP 503.

Solution

To troubleshoot a key-retrieval failure for which the KME returns HTTP 503, review the KME response message in the firewall log.

- Run the initiator and responder commands back-to-back with minimal delay.

  1. On the initiator, run the following command:

     ```
     request security key-manager profiles get profile-keys name KM_PROFILE_1 key-count 1 peer-
     sae-id id2
     ```

2. On the responder, immediately run the following command:

```
request security key-manager profiles get profile-keys-with-id name KM_PROFILE_1 peer-sae-
id id1 key-id 12345678-1234-5678-1234-567812345678
```

- Verify KME configuration to ensure the following settings on the KME2:
  - The KME2 stores the key for the correct SAE pair. If it does not, the firewall cannot return the key.
  - Key stream distribution configuration is correct; otherwise, KME2 has no key to return.
  - SAE pairing between endpoints is accurate.
  - Limit repeated requests to avoid rate limiting by the KME.
- If the KME returns temporary backoff responses, retry the key retrieval. If key retrieval returns HTTP 401, correct the client-certificate, authentication, or KME authorization configuration before retrying.

If the TLS validation succeeds but the KME returns HTTP 503, internal server error, the failure indicates a KME issue and did not fulfill the request. Ensure that the responder retrieves the key within the allowed time window. Verify key distribution settings on the KME and avoid excessive requests.

Other Common Issues in Quantum Safe IPsec VPN

Review the following table for other common failures in Quantum Safe IPsec VPN deployments:

Table 132: Common Issues in Quantum Safe IPsec VPN

Issue Type	Root Cause	Solution
Connectivity failure to QKD device	Loss of IP connectivity between the firewall and QKD device due to VLAN mismatch, storm-control actions, port security violations, DHCP issues, Address Resolution Protocol (ARP) inconsistencies, or switch forwarding problems.	Verify L2 and L3 connectivity. Check VLAN assignments, switch port status, storm-control profile, port-security settings, ARP tables, and routing. Restore stable connectivity before configuring QKD.

Certificate import failure	Unsupported certificate format (DER instead of PEM), passphrase-protected private keys, unsupported key format, incomplete certificate chain, or validation failures when using a self-signed CA.	Before importing certificates into the firewall: • Convert certificates to PEM format. • Remove private key passphrases. • Convert keys to a supported format (for example, PKCS#8) • Validate the certificate chain.
Quantum key retrieval failure due to certificate trust validation issues	The KME presents an unknown self-signed root certificate, which leads to TLS validation failure and causes the firewall to reject the connection. Typical errors include, `self-signed certificate in certificate chain` and `unable to verify the first certificate`. This issue doesn't occur when you configure trusted CAs.	Import the KME root CA certificate into the firewall CA profile and associate the profile with the Quantum Key Manager configuration. Create a CA profile so that the firewall trusts the KME server certificate: `set security pki ca-profile ROOT_CA_PROFILE ca-identity ROOT_CA`. Load the root CA certificate in PEM format using the command, `request security pki ca-certificate load ca-profile ROOT_CA_PROFILE filename /var/tmp/root-ca.pem`. After you import the root CA certificate, the firewall can validate the KME certificate chain, establish HTTPS sessions, and retrieve QKD keys successfully. Verify the CA certificate using the command, `show security pki ca-certificate`

RELATED DOCUMENTATION

- Quantum Safe IPsec VPN

How to Analyze IKE Phase 2 VPN Status Messages

IN THIS SECTION

- Problem | 1300
- Solution | 1300

Problem

Description

Review and analyze VPN status messages related to issues caused by an inactive IKE Phase 2.

Symptoms

- IKE Phase 2 is not active.
- The **show security ipsec security-associations** command output does not list the remote address of the VPN.

Solution

The best way to troubleshoot the IKE Phase 2 issues is by reviewing the VPN status messages of the responder firewall.

The responder firewall is the *receiver* side of the VPN that receives the tunnel setup requests. The initiator firewall is the *initiator* side of the VPN that sends the initial tunnel setup requests.

1. Using the CLI, configure a syslog file, **kmd-logs**, for VPN status logs on the responder firewall.

   See KB10097-How to configure syslog to display VPN status messages. As you bring up the VPN tunnel, the messages are captured in **ldm-logs**.

2. Using the CLI, check for Phase 2 error messages: **show log kmd-logs**

   Sample output messages:

   - ```
     Message:
     Jul 10 16:14:30 210-2 kmd[52472]: IKE Phase-2: Failed to match the peer proxy IDs
     ```
     ```

```
[p2_remote_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.10.0/24),
p2_local_proxy_id=ipv4_subnet(any:0,[0..7]=10.10.10.0/24)] for local ip: 2.2.2.1, remote
peer ip:2.2.2.2
```

- Meaning—The proxy identity of the peer device does not match the local proxy identity.
- Action—The proxy ID must be an exact reverse of the peer's configured proxy ID. See KB10124 - How to fix the Phase 2 error: Failed to match the peer proxy IDs.

- ```
  Message:
  Jul 16 21:14:20 kmd[1456]: IKE Phase-2 Failure: Quick mode - no proposal chosen
  [spi=cf0f6152, src_ip=4.4.4.4, dst_ip=3.3.3.2]
  Jul 16 21:14:20 kmd[1456]: KMD_VPN_PV_PHASE2: IKE Phase-2 Failure: Quick mode - no
  proposal chosen [spi=cf0f6152, src_ip=4.4.4.4, dst_ip=3.3.3.2]
  Jul 16 21:14:20 kmd[1456]: IKE Phase-2: Negotiations failed. Local gateway: 4.4.4.4,
  Remote gateway: 3.3.3.2
  ```

  - Meaning—The device running Junos OS did not accept any of the IKE Phase 2 proposals that the specified IKE peer sent.
  - Action—Verify the local Phase 2 VPN configuration elements. The Phase 2 proposal elements include the following:
    - Authentication algorithm
    - Encryption algorithm
    - Lifetime kilobytes
    - Lifetime seconds
    - Protocol
    - Perfect forward secrecy

  You can change the local configuration to accept at least one of the remote peer's Phase 2 proposals, or contact the remote peer's administrator and arrange for the IKE configurations at both ends of the tunnel to use at least one mutually acceptable Phase 2 proposal.

Sample output messages:

- **IPsec proposal mismatch**

  - ```
    Message:
    Sep 7 09:26:57 kmd[1393]: IKE negotiation failed with error: No proposal chosen. IKE
    Version: 1, VPN: vpn1 Gateway: ike-gw, Local: 10.10.10.1/500, Remote: 10.10.10.2/500,
    ```

```
Local IKE-ID: 10.10.10.1,
Remote IKE-ID: 10.10.10.2, VR-ID: 0
```

NOTE: If `Local IKE-ID` and `Remote IKE-ID` are displayed as `Not-Available`, then it is a Phase 1 failure message. See KB30548 - IKE Phase 1 VPN status messages in 12.1X44 and later releases.

Action—Verify the local Phase 2 VPN configuration elements. The Phase 2 proposal elements include the following:

- Authentication algorithm
- Encryption algorithm
- Lifetime kilobytes
- Lifetime seconds
- Protocol
- Perfect forward secrecy

- **Proxy-ID mismatch**

  Sample output messages:

  - ```
    Sep 7 09:23:05 kmd[1334]: IKE Phase-2: Failed to match the peer proxy IDs
    [p2_remote_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.1.0/24),
    p2_local_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.3.0/24)] for local ip: 10.10.10.2,
    remote peer ip:10.10.10.1
    ```

  - ```
    Sep 7 09:23:05 kmd[1334]: IKE Phase-2: Failed to match the peer proxy IDs
    [p2_remote_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.1.0/24),
    p2_local_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.3.0/24)] for local ip: 10.10.10.2,
    remote peer ip:10.10.10.1
    ```

  Action—The proxy ID must be an exact reverse match of the peer's configured proxy ID. See KB10124 - How to fix the Phase 2 error: Failed to match the peer proxy IDs.

If the VPN connection is established successfully, you can see the following messages in the syslog:

- ```
  Sep 10 08:35:03 kmd[1334]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.2, Remote
  gateway: 10.10.10.1, Local ID: ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Remote ID:
  ```
  ```

```
ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Direction: inbound, SPI: 0x4b23e914, AUX-SPI: 0,
Mode: Tunnel, Type: dynamic
Sep 10 08:35:03 kmd[1334]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.2, Remote
gateway: 10.10.10.1, Local ID: ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Remote ID:
ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Direction: outbound, SPI: 0xa90982b3, AUX-SPI:
0, Mode: Tunnel, Type: dynamic
Sep 10 08:35:03 kmd[1334]: KMD_VPN_UP_ALARM_USER: VPN test_vpn from 10.10.10.1 is up.
Local-ip: 10.10.10.2, gateway name: ike-gw, vpn name: vpn1, tunnel-id: 131073, local
tunnel-if: st0.0, remote tunnel-ip: Not-Available, Local IKE-ID: 10.10.10.2, Remote IKE-
ID: 10.10.10.1, XAUTH username: Not-Applicable, VR id: 0
```

- ```
  Sep 9 06:57:34 kmd[1393]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.1, Remote
  gateway: 10.10.10.2, Local ID: ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Direction: inbound, SPI: 0xa90982b3, AUX-SPI: 0,
  Mode: Tunnel, Type: dynamic, Traffic-selector:
  Sep 9 06:57:34 kmd[1393]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.1, Remote
  gateway: 10.10.10.2, Local ID: ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Direction: outbound, SPI: 0x4b23e914, AUX-SPI:
  0, Mode: Tunnel, Type: dynamic, Traffic-selector:
  Sep 9 06:57:34 kmd[1393]: KMD_VPN_UP_ALARM_USER: VPN test_vpn from 10.10.10.2 is up. Local-
  ip: 10.10.10.1, gateway name: ike-gw, vpn name: vpn1, tunnel-id: 131073, local tunnel-if:
  st0.0, remote tunnel-ip: Not-Available, Local IKE-ID: 10.10.10.1, Remote IKE-ID:
  10.10.10.2, XAUTH username: Not-Applicable, VR id: 0, Traffic-selector: , Traffic-selector
  local ID: ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Traffic-selector remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.3.0/24)ze: 12px;">IPsec Proposal mismatch
  ```

3. If you could not locate any Phase 2 messages, proceed to Step "4" on page 1303.

4. Using the CLI, review the Phase 2 proposals and confirm that the configuration matches the Phase 2 proposals configured by the peer: **show security ipsec**

```
show security ipsec
proposal ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
```
```

```
    proposals ipsec-phase2-proposal;
}
vpn ike-vpn-srx1 {
    vpn-monitor;
    ike {
        gateway gw-srx1;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

5. If the issue persists, to open a JTAC case with the Juniper Networks support team, see Data Collection for Customer Support for the data you should collect to assist in troubleshooting before opening a JTAC case.

14

CHAPTER

Configuration Statements and Operational Commands

IN THIS CHAPTER

- Junos CLI Reference Overview | **1306**

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands