JUNIPER NETWORKS | Engineering Simplicity

# IPsec VPN User Guide

Published
2023-12-14

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 © 2023 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 | xx

1 PKI Fundamentals

Public Key Infrastructure (PKI) | 2

2 IPsec Fundamentals

Internet Key Exchange | 10

- Introduction to IKE | 10
- IKE Versions | 11
- Interaction Between IKE and IPSec | 11
- IKEv1 Message Exchange | 11
- Phase 1 of IKE Tunnel Negotiation | 12
- Phase 2 of IKE Tunnel Negotiation | 14
- IKEv2 Message Exchange | 15
- Proxy ID | 17
- Traffic Selectors | 17
- IKE Authentication (Preshared Key and Certificate-Based Authentication) | 17
- Network Address Translation-Traversal (NAT-T) | 18
- Suite B and PRIME Cryptographic Suites | 19

IPsec Basics | 20

- IPsec Overview | 20
- IPsec Key Management | 22
- IPsec Security Protocols | 24
- IPsec Tunnel Negotiation | 26
- Supported IPsec and IKE Standards | 26

3

## PKI in Junos OS

**PKI in Junos OS | 31**

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

PKI Components In Junos OS | **33**

**Digital Certificates | 36**

Manually Generating Digital Certificates: Configuration Overview | **36**

**Self-Signed Digital Certificates | 37**

Understanding Self-Signed Certificates | **37**

Example: Generating a Public-Private Key Pair | **38**

Requirements | **39**

Overview | **39**

Configuration | **39**

Verification | **39**

Example: Manually Generating Self-Signed Certificates | **40**

Requirements | **40**

Overview | **40**

Configuration | **40**

Verification | **41**

Using Automatically Generated Self-Signed Certificates (CLI Procedure) | **41**

**Certificate Authority | 43**

Configuring a Trusted CA Group | **43**

Creating a Trusted CA Group for a List of CA Profiles | **43**

Deleting a CA Profile from a Trusted CA Group | **45**

Deleting a Trusted CA Group | **46**

Understanding Certificate Authority Profiles | **46**

Example: Configuring a CA Profile | **47**

Requirements | **48**

Overview | **48**

Configuration | **48**

Verification | **50**

Example: Configuring an IPv6 address as the Source Address for a CA Profile | 50

**Certificate Enrollment | 51**

Enrolling Digital Certificates Online: Configuration Overview | 52

Understanding Online CA Certificate Enrollment | 53

Understanding Local Certificate Requests | 53

Enrolling a CA Certificate Online Using SCEP | 53

Example: Enrolling a Local Certificate Online Using SCEP | 54

- Requirements | 54
- Overview | 55
- Configuration | 55
- Verification | 56

Example: Using SCEP to Automatically Renew a Local Certificate | 57

- Requirements | 57
- Overview | 57
- Configuration | 58
- Verification | 58

Understanding CMPv2 and SCEP Certificate Enrollment | 59

Understanding Certificate Enrollment with CMPv2 | 59

Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server | 62

- Requirements | 62
- Overview | 62
- Configuration | 63
- Verification | 63

Example: Loading CA and Local Certificates Manually | 64

- Requirements | 64
- Overview | 64
- Configuration | 65
- Verification | 65

Deleting Certificates (CLI Procedure) | 66

**Certificate Revocation | 67**

Understanding Online Certificate Status Protocol and Certificate Revocation Lists | 67

Example: Manually Loading a CRL onto the Device | 70

- Requirements | 70
- Overview | 71
- Configuration | 71
- Verification | 71

Understanding Dynamic CRL Download and Checking | 72

Example: Configuring a Certificate Authority Profile with CRL Locations | 75

- Requirements | 75
- Overview | 76
- Configuration | 76
- Verification | 77

Example: Verifying Certificate Validity | 77

- Requirements | 77
- Overview | 77
- Configuration | 78
- Verification | 78

Deleting a Loaded CRL (CLI Procedure) | 79

**Certificate Validation | 79**

Understanding Digital Certificate Validation | 80

Example: Validating Digital Certificate by Configuring Policy OIDs on an SRX Series Firewall | 86

- Requirements | 86
- Overview | 87
- Configuration | 87
- Verification | 88

**Dynamic Updated of Trusted CA Certificates | 91**

Understanding Dynamic Update of Trusted CA Certificates | 91

Configuring Dynamic Update of Trusted CA Certificates | 93

- Checking connectivity to the CDN server | 94
- Enabling automatic download of default trusted CA certificates | 94
- Providing custom configuration for automatic download of default trusted CA certificates | 95

Downloading default trusted CA certificates explicitly | **96**

Checking the download status of default trusted CA certificates | **97**

Deactivating automatic download of trusted CA certificates | **98**

**ACME Protocol | 99**

Understanding ACME Protocol | **99**

Enroll Local Certificate Using Let's Encrypt Server | **100**

Manual Re-Enroll Local Certificate | **102**

Delete ACME Account | **102**

**Configure Multiple Certificate Types to Establish IKE and IPsec SA | 103**

Requirements | **103**

Overview | **104**

Topology | **104**

Configuration | **105**

Verification | **115**

4 **IPsec VPN in Junos OS**

**Internet Key Exchange (IKE) for IPsec VPN | 128**

IKE and IPsec Packet Processing | **128**

Introduction to IKE in Junos OS | **136**

IKE Proposal | **141**

IKE Policy | **142**

Rekeying and Reauthentication | **142**

IKE Authentication (Certificate-Based Authentication) | **144**

IKE Protection from DDoS Attacks | **147**

Configure Protection Against IKE DDoS Attacks | **150**

Configure the IKE Session for Half Open IKE SAs | **151**

Configure the IKE Session for Full Open IKE SAs | **154**

Configure the IKE Session Blocklists | **155**

Example: Configuring a Device for Peer Certificate Chain Validation | **157**

Requirements | **158**

Overview | **158**

Configuration | **159**

Verification | **167**

IKE and IPsec SA Failure for a Revoked Certificate | **169**

IKEv2 Fragmentation | **171**

IKE Policy with a Trusted CA | **172**

Configuring Establish-Tunnel Responder-only in IKE | **174**

**IPsec VPN Overview | 176**

IPsec VPN Topologies on SRX Series Firewalls | **177**

Comparing Policy-Based and Route-Based VPNs | **177**

Comparison of Policy-Based VPNs and Route-Based VPNs | **180**

Understanding IKE and IPsec Packet Processing | **181**

Distribution of IKE and IPsec Sessions Across SPUs | **184**

VPN Support for Inserting Services Processing Cards | **186**

Cryptographic acceleration support on SRX5K-SPC3 Card, SRX mid-range platforms and vSRX Virtual Firewall | **188**

IPsec VPN Feature Support on SRX5000 Line with SRX5K-SPC3 and vSRX Virtual Firewall Instances with New Package | **190**

Routing Protocols Support on IPsec VPN Tunnels | **191**

Anti-Replay Window | **197**

Understanding Hub-and-Spoke VPNs | **198**

5 **VPN Configuration Overview**

**IPsec VPN Configuration Overview | 202**

IPsec VPN with Autokey IKE Configuration Overview | **202**

Recommended Configuration Options for Site-to-Site VPN with Static IP Addresses | **203**

Recommended Configuration Options for Site-to-Site or Dialup VPNs with Dynamic IP Addresses | **205**

Understanding IPsec VPNs with Dynamic Endpoints | **207**

Understanding IKE Identity Configuration | **209**

Configuring Remote IKE IDs for Site-to-Site VPNs | **211**

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | **211**

Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Firewall | **214**

- Requirements | **214**
- Overview | **214**
- Configuration | **215**
- Verification | **219**

Configuring IPsec VPN Using the VPN Wizard | **220**

Example: Configuring a Hub-and-Spoke VPN | **221**

- Requirements | **221**
- Overview | **222**
- Configuration | **233**
- Verification | **264**

**Comparing Policy-Based and Route-Based VPNs | 272**

**Chassis Cluster HA Control Link Encryption | 275**

6 **Policy Based VPN**

**Policy-Based IPsec VPNs | 279**

Understanding Policy-Based IPsec VPNs | **279**

Example: Configuring a Policy-Based VPN | **280**

- Requirements | **280**
- Overview | **281**
- Configuration | **284**
- Verification | **298**

**Configure Policy-Based IPsec VPN with Certificates | 305**

- Requirements | **305**
- Overview | **306**
- Configuration | **309**
- Verification | **321**

Troubleshooting IKE, PKI, and IPsec Issues | 328

Configure IPsec VPN with OCSP for Certificate Revocation Status | 340

Requirements | 340

Overview | 340

Configuration | 344

Verification | 355

IPv6 IPsec VPNs | 361

VPN Feature Support for IPv6 Addresses | 362

Understanding IPv6 IKE and IPsec Packet Processing | 368

IPv6 IPsec Configuration Overview | 375

Example: Configuring an IPv6 IPsec Manual VPN | 375

Requirements | 376

Overview | 376

Configuration | 376

Verification | 379

Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN | 379

Requirements | 380

Overview | 380

Configuration | 385

Verification | 400

7 Route Based VPN

Route-Based IPsec VPNs | 406

Understanding Route-Based IPsec VPNs | 406

Example: Configuring a Route-Based VPN | 407

Requirements | 407

Overview | 408

Configuration | 411

Verification | 426

Route-Based VPN with IKEv2 | 433

Example: Configuring a Route-Based VPN for IKEv2 | 434

Requirements | **435**

Overview | **435**

Configuration | **439**

Verification | **454**

Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload | **460**

Requirements | **460**

Overview | **460**

Configuration | **466**

Verification | **487**

IKE Policy with a Trusted CA | **493**

**Secure Tunnel Interface in a Virtual Router | 495**

Understanding Virtual Router Support for Route-Based VPNs | **495**

Example: Configuring an st0 Interface in a Virtual Router | **497**

Requirements | **497**

Overview | **497**

Configuration | **500**

Verification | **506**

**Dual Stack Tunnels over an External Interface | 507**

Understanding VPN Tunnel Modes | **508**

Example: Configuring Dual-Stack Tunnels over an External Interface | **511**

Requirements | **511**

Overview | **512**

Configuration | **515**

Verification | **521**

**IPsec VPN Tunnels with Chassis Clusters | 525**

Understanding Dual Active-Backup IPsec VPN Chassis Clusters | **525**

Example: Configuring Redundancy Groups for Loopback Interfaces | **527**

Requirements | **528**

Overview | **528**

Configuration | **530**

Verification | **534**

**Traffic Selectors in Route-Based VPNs | 536**

Understanding Traffic Selectors in Route-Based VPNs | **536**

Example: Configuring Traffic Selectors in a Route-Based VPN | **543**

- Requirements | **543**
- Overview | **544**
- Configuration | **545**
- Verification | **559**

8 **Class-Of-Service Based VPN**

**CoS-Based IPsec VPNs | 565**

Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs | **565**

Understanding Traffic Selectors and CoS-Based IPsec VPNs | **568**

Example: Configuring CoS-Based IPsec VPNs | **571**

- Requirements | **571**
- Overview | **571**
- Configuration | **575**
- Verification | **596**

Understanding CoS Support on st0 Interfaces | **600**

9 **NAT-T**

**Route-Based and Policy-Based VPNs with NAT-T | 605**

Understanding NAT-T | **605**

Example: Configuring a Route-Based VPN with the Responder behind a NAT Device | **606**

- Requirements | **607**
- Overview | **607**
- Configuration | **613**
- Verification | **632**

Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device | **643**

- Requirements | **643**
- Overview | **643**
- Configuration | **650**
- Verification | **680**

Example: Configuring NAT-T with Dynamic Endpoint VPN | **690**

- Requirements | **690**
- Overview | **691**
- Configuration | **693**
- Verification | **709**

10 **Group VPN**

**Group VPNv1 | 715**

Group VPNv1 Overview | **715**

Group VPNv1 Configuration Overview | **725**

Understanding IKE Phase 1 Configuration for Group VPNv1 | **726**

Understanding IPsec SA Configuration for Group VPNv1 | **727**

Understanding Dynamic Policies for Group VPNv1 | **727**

Understanding Antireplay for Group VPNv1 | **728**

Example: Configuring Group VPNv1 Server and Members | **729**

- Requirements | **729**
- Overview | **729**
- Configuration | **730**
- Verification | **747**

Example: Configuring Group VPNv1 Server-Member Communication for Unicast Rekey Messages | **750**

- Requirements | **751**
- Overview | **751**
- Configuration | **751**
- Verification | **752**

Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages | **752**

- Requirements | **753**
- Overview | **753**
- Configuration | **753**
- Verification | **755**

Example: Configuring Group VPNv1 with Server-Member Colocation | **756**

Requirements | **756**
Overview | **757**
Configuration | **757**
Verification | **767**

**Group VPNv2 | 769**

Group VPNv2 Overview | **770**

Group VPNv2 Configuration Overview | **775**

Understanding IKE Phase 1 Configuration for Group VPNv2 | **777**

Understanding IPsec SA Configuration for Group VPNv2 | **777**

Understanding Group VPNv2 Traffic Steering | **778**

Understanding the Group VPNv2 Recovery Probe Process | **780**

Understanding Group VPNv2 Antireplay | **781**

Example: Configuring a Group VPNv2 Server and Members | **781**

Requirements | **782**
Overview | **782**
Configuration | **783**
Verification | **819**

Example: Configuring Group VPNv2 Server-Member Communication for Unicast Rekey Messages | **828**

Requirements | **828**
Overview | **828**
Configuration | **829**
Verification | **829**

**Group VPNv2 Server Clusters | 830**

Understanding Group VPNv2 Server Clusters | **830**

Understanding Group VPNv2 Server Cluster Limitations | **835**

Understanding Group VPNv2 Server Cluster Messages | **836**

Understanding Configuration Changes with Group VPNv2 Server Clusters | **838**

Migrating a Standalone Group VPNv2 Server to a Group VPNv2 Server Cluster | **843**

Example: Configuring a Group VPNv2 Server Cluster and Members | **844**

Requirements | **844**
Overview | **845**
Configuration | **847**
Verification | **916**

11 **ADVPN**

**Auto Discovery VPNs | 931**

Understanding Auto Discovery VPN | **931**

Understanding Traffic Routing with Shortcut Tunnels | **937**

Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels | **940**

Requirements | **941**
Overview | **941**
Configuration | **945**
Verification | **970**

Example: Configuring ADVPN with OSPFv3 for IPv6 Traffic | **993**

Requirements | **993**
Overview | **994**
Configuration | **997**
Verification | **1026**

Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established | **1029**

12 **AutoVPN**

**AutoVPN on Hub-and-Spoke Devices | 1033**

Understanding AutoVPN | **1033**

Understanding Spoke Authentication in AutoVPN Deployments | **1039**

AutoVPN Configuration Overview | **1042**

Example: Configuring Basic AutoVPN with iBGP | **1043**

Requirements | **1043**
Overview | **1044**
Configuration | **1047**
Verification | **1076**

Example: Configuring Basic AutoVPN with iBGP for IPv6 Traffic | **1080**

Requirements | **1080**

Overview | **1080**

Configuration | **1084**

Verification | **1115**

Example: Configuring AutoVPN with iBGP and ECMP | **1118**

Requirements | **1118**

Overview | **1119**

Configuration | **1122**

Verification | **1147**

Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels | **1151**

Requirements | **1152**

Overview | **1152**

Configuration | **1156**

Verification | **1181**

Example: Configuring Basic AutoVPN with OSPF | **1188**

Requirements | **1188**

Overview | **1189**

Configuration | **1192**

Verification | **1218**

Example: Configuring AutoVPN with OSPFv3 for IPv6 Traffic | **1221**

Requirements | **1222**

Overview | **1222**

Configuration | **1226**

Verification | **1254**

Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors | **1258**

Requirements | **1258**

Overview | **1259**

Configuration | **1262**

Verification | **1276**

Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors | **1280**

Requirements | **1280**

Overview | **1281**

Configuration | 1284

Verification | 1304

Example: Configuring AutoVPN with Pre-Shared Key | 1308

13 Remote Access VPN

Juniper Secure Connect | 1315

14 Monitoring VPN

VPN Monitoring Overview | 1319

VPN Monitoring Methods | 1320

IPsec Datapath Verification | 1321

Dead Peer Detection | 1323

VPN Tunnel Monitoring | 1326

Configure Dead Peer Detection | 1327

Configure VPN Tunnel Monitoring | 1328

VPN Alarms, Audits, and Events | 1330

VPN Alarms and Audits | 1331

VPN Tunnel Events | 1333

Configure VPN Alarms | 1334

Example: Configure an Audible Alert Notification | 1334

Requirements | 1335

Overview | 1335

Configuration | 1335

Verification | 1336

Example: Configure Security Alarms Generation | 1336

Requirements | 1336

Overview | 1336

Configuration | 1337

Verification | 1340

15 Performance Tuning

VPN Session Affinity | 1342

Understanding VPN Session Affinity | **1342**

Enabling VPN Session Affinity | **1344**

Accelerating the IPsec VPN Traffic Performance | **1346**

IPsec Distribution Profile | **1348**

Understanding the Loopback Interface for a High Availability VPN | **1349**

**PowerMode IPsec | 1350**

Improving IPsec Performance with PowerMode IPsec | **1351**

Example: Configuring Behavior Aggregate Classifier in PMI | **1358**

Requirements | **1358**

Overview | **1359**

Configuration | **1359**

Verification | **1363**

Example: Configuring Behavior Aggregate Classifier in PMI for vSRX Virtual Firewall instances | **1364**

Requirements | **1364**

Overview | **1364**

Configuration | **1365**

Verification | **1369**

Example: Configuring and Applying a Firewall Filter for a Multifield Classifier in PMI | **1370**

Requirements | **1371**

Overview | **1371**

Configuration | **1372**

Verification | **1377**

Example: Configuring and Applying Rewrite Rules on a Security Device in PMI | **1378**

Requirements | **1378**

Overview | **1378**

Configuration | **1379**

Verification | **1382**

Configure IPsec ESP Authentication-only Mode in PMI | **1383**

16 **Troubleshooting**

**Troubleshoot a Flapping VPN Tunnel | 1386**

Troubleshoot a VPN That Is Up But Not Passing Traffic | 1389

Troubleshoot a VPN Tunnel That is Down | 1394

How to Analyze IKE Phase 2 VPN Status Messages | 1396

17 Configuration Statements and Operational Commands

Junos CLI Reference Overview | 1402

# 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.

## 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

# PKI Fundamentals

Public Key Infrastructure (PKI) | 2

# Public Key Infrastructure (PKI)

IN THIS SECTION


- Introduction to PKI | 2
- Digital Certificate | 3
- Certificate Authority | 3
- Private/Public Key Pair | 3
- Certificate Enrollment Options | 3
- Certificate Revocation Options | 4
- Certificate Request Types | 4
- Certificate Signatures and Verification | 4
- Certificate Validation | 6


This topic describes the overview of public key infrastructure and includes the following sections:

## Introduction to PKI

Public key infrastructure (PKI) provides a way of verifying the identity of a remote site by using a digital certificate. PKI uses a certificate authority (CA) to validate your information and to sign it with a digital signature such that neither your information nor the signature can be modified. Once signed, the information becomes a digital certificate. Devices that receive a digital certificate can verify the information in the certificate by validating the signature using public key cryptography.

The Public Key Infrastructure (PKI) provides an infrastructure for digital certificate management and consists of:

- Registration Authority (RA) that verifies the identities of entities, authorizes their certificate requests, and generates unique asymmetric key pairs (unless the users' certificate requests already contain public keys)
- Certificate Authority (CA) that issues corresponding digital certificates for the requesting entities.
- A certificate revocation list (CRL) identifying the certificates that are no longer valid. Each entity possessing the authentic public key of a CA can verify the certificates issued by that CA.

## Digital Certificate

A digital certificate is an electronic file that verifies the identity of the certificate's holder to protect data exchanged online. Digital certificates provide a way of authenticating users through a trusted third party called a certificate authority (CA). The CA validates the identity of a certificate holder and "signs" the certificate to attest that it has not been forged or altered. Alternatively, you can use a self-signed certificate to attest to your identity.

A key pair is a critical element of a digital certificate implementation. The public key is included in the local digital certificate and the private key is used to decrypt data received from peers.

Certificates have a finite lifetime and are defined by a start time and an end time. The certificate becomes invalid when the life time expires. When the certificate expires, a certificate renewal or a new certificate request is required.

## Certificate Authority

A CA is a trusted third-party organization that creates, enrolls, validates, and revokes digital certificates. The CA guarantees a user's identity and issues public and private keys for message encryption and decryption (coding and decoding). A CA also generates certificate revocation lists (CRLs) which are lists of revoked certificates.

## Private/Public Key Pair

When setting up a PKI, you must include Public and private keys that are generated in pairs and linked mathematically.

When request for the certificate, you must include the public key in the certificate enrollment request. The public key will be included in the granted certificate and the private key is kept on the requesting device. A message encrypted with the public key can be decrypted by using the corresponding private key. The private-public key pair is also used for creating digital signatures.

## Certificate Enrollment Options

You can request a CA digital certificate either online or manually:

- Manual certificate enrollment—This process includes generation of a PKCS10 request, submission to the certificate authority (CA), retrieval of the signed certificate, and manually loading of the certificate into the Junos OS device as the local certificate.
- Online certificate enrollment—You can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) for online certificate enrollment.

## Certificate Revocation Options

- Certificate revocation list (or CRL)—Certificate authority (CA) periodically publishes a list of revoked certificate using a certificate revocation list (CRL). The CRL contains the list of digital certificates with serial numbers that have been canceled before their expiration date.
- Online Certificate Status Protocol (OCSP)—OCSP is used to check the revocation status of X509 certificates. The OCSP provides revocation status on certificates in real time and is useful in time-sensitive situations such as bank transactions and stock trades

## Certificate Request Types

Public Key Infrastructure (PKI) allows users to authenticate each other using digital certificates issued by CA. PKI Uses X.509, Public Key Cryptography Standards (PKCS) to define the standard formats for certificates and their use. In PKI, an applicant uses a certificate signing request (CSR) to apply for a digital certificate to a certificate authority (CA). The request can be in one of the standard:

- Public-Key Cryptography Standard # (PKCS#) (PKCS7, PKCS10, PKCS11, PKCS12)
- x509-signaturere.

## Certificate Signatures and Verification

A digital certificate is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity.

The CA server you use can be owned and operated by an independent CA or by your own organization, in which case you become your own CA. If you use an independent CA, you must contact them for the addresses of their CA and certificate revocation list (CRL) servers (for obtaining certificates and CRLs)

and for the information they require when submitting personal certificate requests. When you are your own CA, you determine this information yourself.

The CA that issues a certificate uses a hash algorithm to generate a digest, and then “signs” the certificate by encrypting the digest with its private key. The result is a digital signature. The CA then makes the digitally signed certificate available for download to the person who requested it. Figure 1 on page 5 illustrates this process.

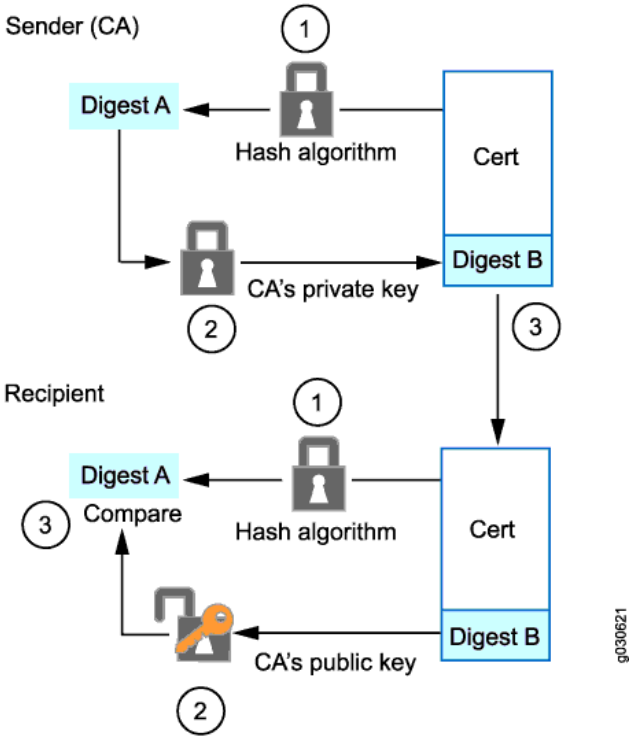
The recipient of the certificate generates another digest by applying the same hash algorithm to the certificate file, then uses the CA's public key to decrypt the digital signature. By comparing the decrypted digest with the digest just generated, the recipient can confirm the integrity of the CA's signature and, by extension, the integrity of the accompanying certificate. Figure 1 on page 5 illustrates this process.

A certificate is considered valid if the digital signature can be verified and the serial number of the certificate is not listed in a certificate revocation list.

**Figure 1: Digital Signature Verification**

1. Using a hash algorithm, the CA generates digest A from the certificate.
2. Using the private key, the CA encrypts digest A. The result is digest B, the digital signature.
3. The CA sends the digitally signed certificate to the person who requested it.

1. Using a hash algorithm, the recipient generates digest A from the certificate.
2. Using the CA's public key, the recipient decrypts digest B.
3. The recipient compares digest A with digest B. If they match, the recipient knows that the certificate has not been tampered with.




When Digital Signature Algorithm (DSA) signatures are used, the SHA-1 hash algorithm is used to generate the digest. When Rivest-Shamir-Adleman (RSA) signatures are used, SHA-1 is the default hash algorithm used to generate the digest; you can specify the SHA-256 hash algorithm with the `digest` option of the `request security pki generate-certificate-request` or `request security pki local-certificate generate-self-signed` commands. When Elliptic Curve Digital Signature Algorithm (ECDSA) signatures are

used, the SHA-256 hash algorithm is used for ECDSA-256 signatures and the SHA-384 hash algorithm is used for ECDSA-384 signatures.

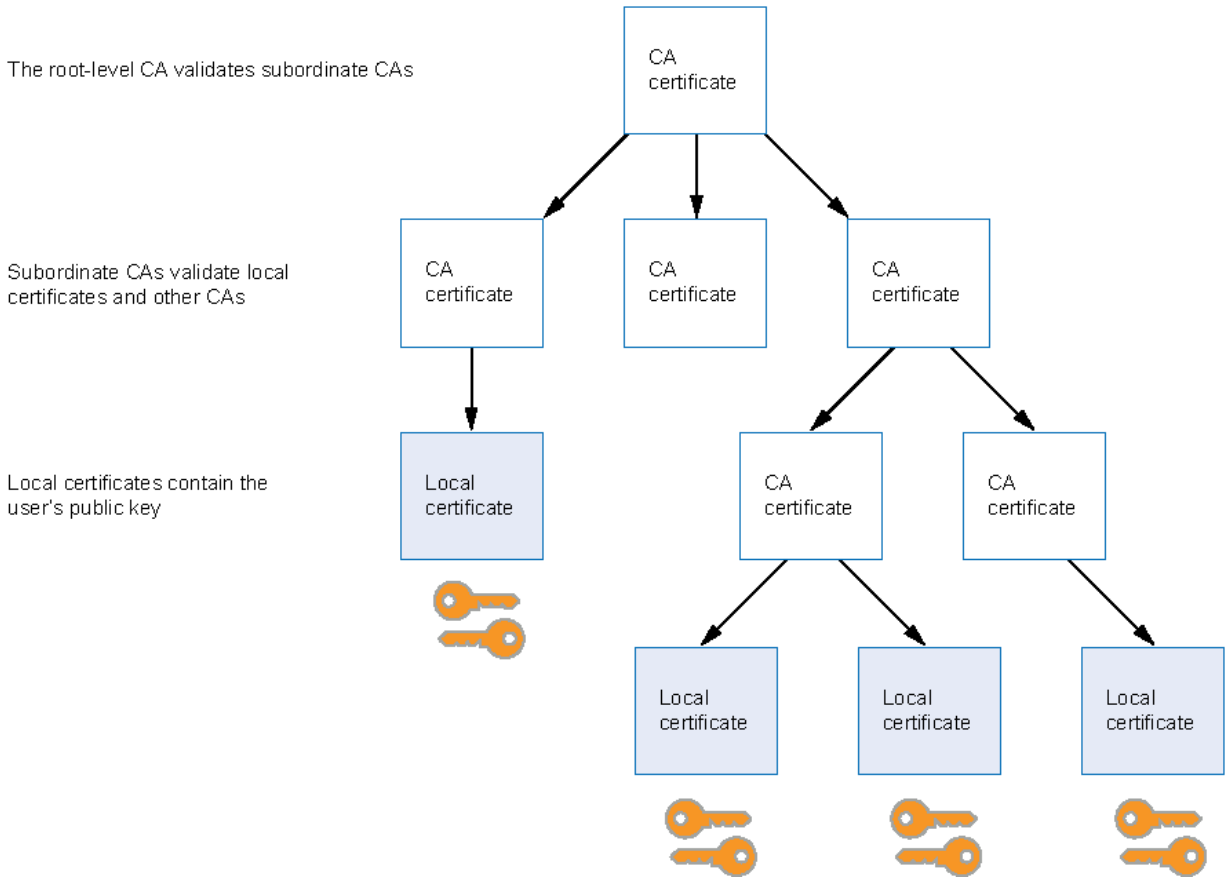
Starting in Junos OS Release 18.1R3, the default hash algorithm that is used for validating automatically and manually generated self-signed PKI certificates is Secure Hash Algorithm 256 (SHA-256). Prior to Junos OS Release 18.1R3, SHA-1 is used as default hash algorithm.

## Certificate Validation

To verify the trustworthiness of a certificate, you must be able to track a path of certified certificate authorities (CAs) from the one issuing your local certificate to the root authority of a CA domain. Public key infrastructure (PKI) refers to the hierarchical structure of trust required for the successful implementation of public key cryptography.

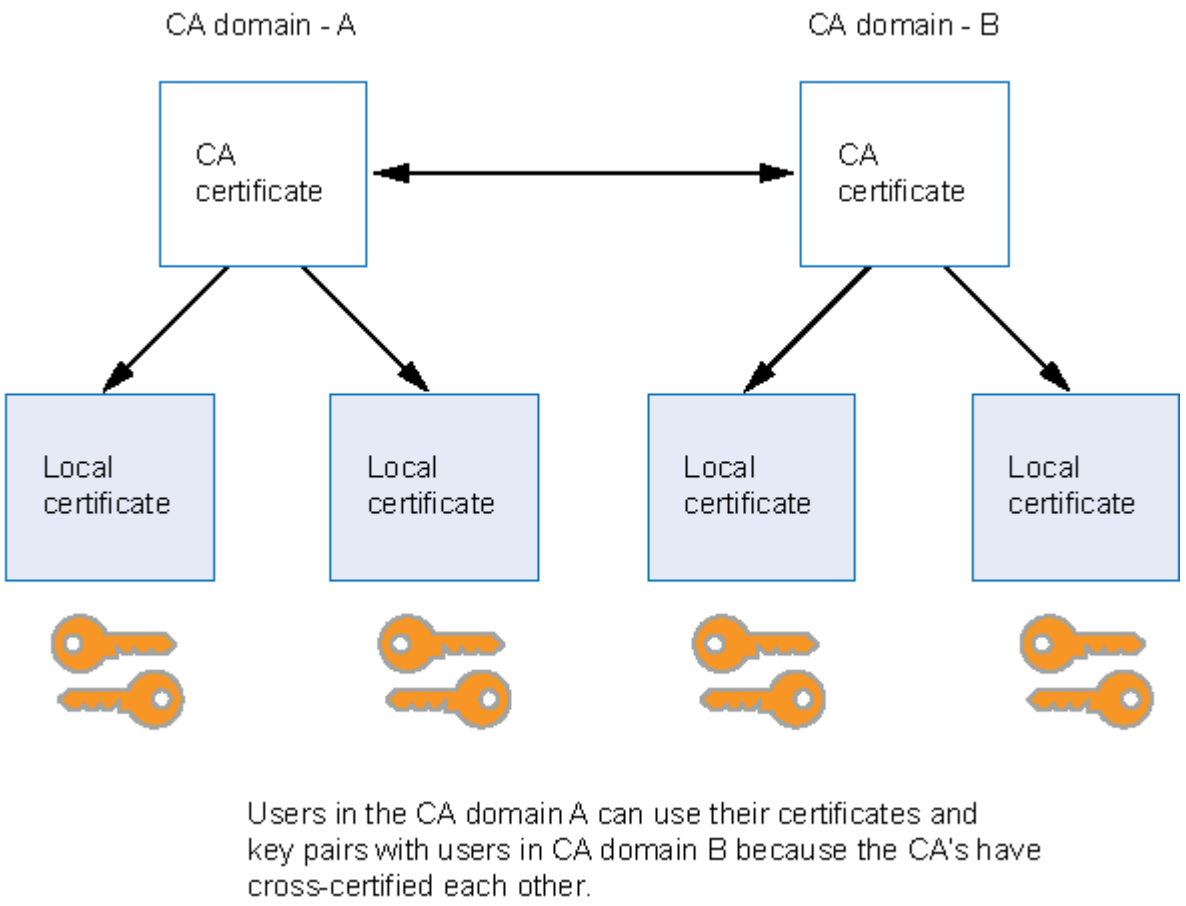
Figure 2 on page 7 shows the structure of a single-domain certificate authority with multiple hierarchy levels.

**Figure 2: PKI Hierarchy of Trust—CA Domain**




If certificates are used solely within an organization, that organization can have its own CA domain within which a company CA issues and validates certificates for its employees. If that organization later wants its employees to exchange their certificates with certificates from another CA domain (for example, with employees at another organization that has its own CA domain), the two CAs can develop cross-certification by agreeing to trust the authority of each other. In this case, the PKI structure does not extend vertically but does extend horizontally. See Figure 3 on page 8.

**Figure 3: Cross-Certification**




**Release History Table**

| Release | Description |
|---|---|
| 18.1R3 | Starting in Junos OS Release 18.1R3, the default hash algorithm that is used for validating automatically and manually generated self-signed PKI certificates is Secure Hash Algorithm 256 (SHA-256). Prior to Junos OS Release 18.1R3, SHA-1 is used as default hash algorithm. |

2

CHAPTER

# IPsec Fundamentals

Internet Key Exchange | 10

IPsec Basics | 20

# Internet Key Exchange

IN THIS SECTION


- Introduction to IKE | **10**
- IKE Versions | **11**
- Interaction Between IKE and IPSec | **11**
- IKEv1 Message Exchange | **11**
- Phase 1 of IKE Tunnel Negotiation | **12**
- Phase 2 of IKE Tunnel Negotiation | **14**
- IKEv2 Message Exchange | **15**
- Proxy ID | **17**
- Traffic Selectors | **17**
- IKE Authentication (Preshared Key and Certificate-Based Authentication) | **17**
- Network Address Translation-Traversal (NAT-T) | **18**
- Suite B and PRIME Cryptographic Suites | **19**


## 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 | 12
- Aggressive Mode | 13


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 20.)
- Authentication algorithms—Message Digest 5 (MD5 ) and Secure Hash Algorithm (SHA). (See "IPsec Overview" on page 20.)
- Diffie-Hellman (DH) group. (See "IPsec Overview" on page 20.)
- Preshared key or RSA/DSA certificates. (See "IPsec Overview" on page 20.)

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 | **726**

*proposal-set (Security IKE)*

# Phase 2 of IKE Tunnel Negotiation

IN THIS SECTION


- Proxy IDs | 14
- Perfect Forward Secrecy | 14
- Replay Protection | 15


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 | **777**

*policy (Security IPsec)*

## IKEv2 Message Exchange

**IN THIS SECTION**

- IKEv2 Configuration Payload | **16**
- IKEv2 Rekeying and Reauthentication | **16**
- IKEv2 Fragmentation | **16**
- Traffic Selectors for IKEv2 | **17**

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 both route-based and policy-based VPNs. However, the multi-proxy ID is supported for only 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.<br>• 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.<br>• 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

**IN THIS SECTION**


- IPsec Overview | **20**
- IPsec Key Management | **22**
- IPsec Security Protocols | **24**
- IPsec Tunnel Negotiation | **26**
- Supported IPsec and IKE Standards | **26**


## IPsec Overview

**IN THIS SECTION**


- Security Associations | **21**


IPsec is a suite of related protocols for cryptographically securing communications at the IP Packet Layer. IPsec also provides methods for the manual and automatic negotiation of security associations (SAs) and key distribution, all the attributes for which are gathered in a domain of interpretation (DOI). The IPsec DOI is a document containing definitions for all the security parameters required for the successful negotiation of a VPN tunnel—essentially, all the attributes 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. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. There are 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—that specify the security parameter index (SPI), destination IP address, and security protocol (Authentication Header [AH] or Encapsulating Security Payload [ESP] employed. An SA groups together 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 181.)
- 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.
- Security parameter index (SPI) value.

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

# IPsec Key Management

IN THIS SECTION


- Manual Key | 22
- AutoKey IKE | 22
- Diffie-Hellman Exchange | 23


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 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 10.

This topic includes the following sections:

## Manual Key

With manual keys, administrators at both ends of a tunnel configure all the security parameters. This 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 sure that the keys have not been compromised while 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 security associations 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 preshared key 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 preshared key 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 key pair and acquires a certificate. As long as the issuing certificate authority (CA) is trusted by both sides, the participants can retrieve the peer's public key and verify the peer's signature. There is no need to 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. Diffie Hellman (DH) exchange operations can be performed either in software or in hardware. The following Table 1 on page 23 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 |
| DH Group 5 | 1536-bit |
| DH Group 14 | 2048-bit |
| DH Group 15 | 3072-bit |

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

| Diffie-Hellman (DH) Group | Prime Module Size |
|---|---|
| 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 |

Starting in Junos OS Release 19.1R1, SRX Series Firewalls support DH groups 15, 16, and 21.

Starting in Junos OS Release 20.3R1, vSRX Virtual Firewall instances with junos-ike package installed support DH groups 15, 16, and 21.

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) | **25**
- IPsec Encryption Algorithms (ESP Protocol) | **25**


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 10.

For each VPN tunnel, both AH and ESP tunnel sessions are installed on Services Processing Units (SPUs) and the control plane. Tunnel sessions are updated with the negotiated protocol after negotiation is completed. For SRX5400, SRX5600, and SRX5800 devices, tunnel sessions on anchor SPUs are updated with the negotiated protocol while non-anchor SPUs retain ESP and AH tunnel sessions. ESP and AH tunnel sessions are displayed in the outputs for the `show security flow session` and `show security flow cp-session` operational mode commands.

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. It is generally regarded as more secure than MD5 because of the larger hashes it produces. Because the computational processing is done in the ASIC, the performance cost is negligible.

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

## IPsec Encryption Algorithms (ESP Protocol)

The Encapsulating Security Payload (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 181.)

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.

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

Even though it is possible to select NULL for encryption, it has been demonstrated 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 that determine how the traffic is exchanged in the VPN.

- 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. This is most commonly used when hosts within separate private networks want to communicate over a public network. This mode can be used by both VPN clients and VPN gateways, and protects 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. The data portion of the IP packet is encrypted, but the IP header is not. 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 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

**Release History Table**

| Release | Description |
| --- | --- |
| 19.1R1 | Starting in Junos OS Release 19.1R1, SRX Series Firewalls support DH groups 15, 16, and 21. |

3

CHAPTER

# PKI in Junos OS

PKI in Junos OS | 31

Digital Certificates | 36

Self-Signed Digital Certificates | 37

Certificate Authority | 43

Certificate Enrollment | 51

Certificate Revocation | 67

Certificate Validation | 79

Dynamic Updated of Trusted CA Certificates | 91

ACME Protocol | 99

Configure Multiple Certificate Types to Establish IKE and IPsec SA | 103

# PKI in Junos OS

**SUMMARY**

This topic describes the basic elements of public key infrastructure (PKI) in Junos OS.

**IN THIS SECTION**

- Introduction to PKI in Junos OS | 31
- PKI Components In Junos OS | 33

A public key infrastructure (PKI) supports the distribution and identification of public encryption keys, enabling users to both securely exchange data over networks such as the Internet and verify the identity of the other party.

## Introduction to PKI in Junos OS

**IN THIS SECTION**

- PKI Applications Overview | 31
- Components for Administering PKI in Junos OS | 32
- Basic Elements of PKI in Junos OS | 32

### PKI Applications Overview

The Junos OS uses public/private keys in the following areas:

- SSH/SCP (for secure command-line interface [CLI]-based administration)
- Secure Sockets Layer (SSL) (for secure Web-based administration and for https-based webauth for user authentication)
- Internet Key Exchange (IKE) (for IPsec VPN tunnels)

**NOTE**: Note the following points:

- Currently Junos OS supports only IKE (using public key infrastructure (PKI) certificates for public key validation).
- The SSH and SCP are used exclusively for system administration and depends on the use of out-of-band fingerprints for public key identity binding and validation. Details on SSH are not covered in this topic.

## Components for Administering PKI in Junos OS

The following components are required for administrating PKI in Junos OS:

- CA certificates and authority configuration
- Local certificates including the devices identity (example: IKE ID type and value) and private and public keys
- Certificate validation through a certificate revocation list (CRL)

## Basic Elements of PKI in Junos OS

Junos OS supports three specific types of PKI objects:

- Private/public key pair
- Certificates
  - Local certificate—The local certificate contains the public key and identity information for the Juniper Networks device. The Juniper Networks device owns the associated private key. This certificate is generated based on a certificate request from the Juniper Networks device.
  - Pending certificate — A pending certificate contains a key pair and identity information that is generated into a PKCS10 certificate request and manually sent to a certificate authority (CA). While the Juniper Networks device waits for the certificate from the CA, the existing object (key pair and the certificate request) is tagged as a certificate request or pending certificate.

    **NOTE**: Junos OS Release 9.0 and later supports automatic sending of certificate requests through SCEP.

  - CA certificate — When the certificate is issued by the CA and loaded into the Junos OS device, the pending certificate is replaced by the newly generated local certificate. All other certificates loaded into the device are considered CA certificates.

- Certificate revocation lists (CRLs)

Note the following points about certificates:

- Local certificates are generally used when a Junos OS device has VPNs in more than one administrative domain.
- All PKI objects are stored in a separate partition of persistent memory, apart from the Junos OS image and the system's general configuration.
- Each PKI object has a unique name or certificate-ID given to it when it is created and maintains that ID until its deletion. You can view the certificate-ID by using the `show security pki local-certificate` command.
- A certificate cannot be copied from a device under most circumstances. The private key on a device must be generated on that device only, and it should never be viewed or saved from that device. So PKCS12 files (which contain a certificate with the public key and the associated private key) are not supported on Junos OS devices.
- CA certificates validate the certificates received by the IKE peer. If the certificate is valid, then it is verified in the CRL to see whether the certificate has been revoked.

  Each CA certificate includes a CA profile configuration that stores the following information:

  - CA identity, which is typically the domain name of the CA
  - E-mail address for sending the certificate requests directly to the CA
  - Revocation settings:
    - Revocation check enable/disable option
    - Disabling of revocation check in case of CRL download failure.
    - Location of CRL Distribution Point (CDP) (for manual URL setting)
    - CRL refresh interval

## PKI Components In Junos OS

**IN THIS SECTION**

PKI Management and Implementation | 34

Internet Key Exchange | 34
Trusted CA Group | 34
Cryptographic Key Handling Overview | 35

This topic includes the following sections:

## PKI Management and Implementation

The minimum PKI elements required for certificate-based authentication in Junos OS are:

- CA certificates and authority configuration.
- Local certificates including the device's identity (example: IKE ID type and value) and private and public keys
- Certificate validation through a CRL.

Junos OS supports three different types of PKI objects:

## Internet Key Exchange

The procedure for digitally signing messages sent between two participants in an Internet Key Exchange (IKE) session is similar to digital certificate verification, with the following differences:

- Instead of making a digest from the CA certificate, the sender makes it from the data in the IP packet payload.
- Instead of using the CA's public-private key pair, the participants use the sender's public-private key pair.

## Trusted CA Group

A Certificate Authority (CA) is a trusted third party responsible for issuing and revoking certificates. You can group multiple CAs (CA profiles) in one trusted CA group for a given topology. These certificates are used to establish connection between two endpoints. To establish IKE or IPsec, both the endpoints must trust the same CA. If either of the endpoints are unable to validate the certificate using their respective trusted CA (ca-profile) or trusted CA group, the connection is not established.

For example, there are two endpoints, endpoint A and endpoint B are trying to establish a secure connection. When endpoint B presents it's certificate to endpoint A, the endpoint A will check if the certificate is valid. The CA of the endpoint A verifies the signed certificate that the endpoint B is using

to get authorized. When `trusted-ca` or `trusted-ca-group` is configured, the device will only use the CA profiles added in this `trusted-ca-group` or the CA profile configured under `trusted-ca` to validate the certificate coming from endpoint B. If the certificate is verified as valid, the connection is allowed, else the connection is rejected.

Benefits:

- For any incoming connection request, only the certificate issued by that particular trusted CA of that endpoint gets validated. If not, the authorization will reject establishing the connection.

## Cryptographic Key Handling Overview

With cryptographic key handling, persistent keys are stored in the memory of the device without any attempt to alter them. While the internal memory device is not directly accessible to a potential adversary, those who require a second layer of defense can enable special handling for cryptographic keys. When enabled, the cryptographic key handling encrypts keys when not immediately in use, performs error detection when copying a key from one memory location to another, and overwrites the memory location of a key with a random bit pattern when the key is no longer in use. Keys are also protected when they are stored in the flash memory of the device. Enabling cryptographic key handling feature does not cause any externally observable change in the behavior of the device, and the device continues to interoperate with the other devices.

A cryptographic administrator can enable and disable the cryptographic self-test functions; however, the security administrator can modify the behavior of the cryptographic self-test functions such configuring periodic self-tests or selecting a subset of cryptographic self-tests.

The following persistent keys are currently under the management of IKE and PKI:

- IKE preshared keys (IKE PSKs)
- PKI private keys
- Manual VPN keys

### SEE ALSO

IKE Authentication (Certificate-Based Authentication) | **144**

IPsec Overview | **20**

*request security pki generate-key-pair (Security)*

### RELATED DOCUMENTATION

Certificate Enrollment | **51**

# Digital Certificates

IN THIS SECTION


- Manually Generating Digital Certificates: Configuration Overview | 36


A digital certificate is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity.

Manual certificate processing includes generation of a PKCS10 request, submission to the CA, retrieval of the signed certificate, and manually loading the certificate into the Juniper Networks device. Based on your deployment environment, you can use either SCEP or CMPv2 for online certificate enrollment.

To use a digital certificate to authenticate your identity when establishing a secure VPN connection, you must first do the following:

- Obtain a CA certificate from which you intend to obtain a local certificate, and then load the CA certificate onto the device. The CA certificate can contain a CRL to identify invalid certificates.
- Obtain a local certificate from the CA whose CA certificate you have previously loaded, and then load the local certificate in the device. The local certificate establishes the identity of the Juniper Networks device with each tunnel connection.

## Manually Generating Digital Certificates: Configuration Overview

To obtain digital certificates manually:

1. Generate a key pair on the device. See "Self-Signed Digital Certificates" on page 37.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 47.
3. Generate the CSR for the local certificate and send it to the CA server. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.
4. Load the certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 64.

5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 57.

6. If necessary, load the certificate's CRL on the device. See "Example: Manually Loading a CRL onto the Device" on page 70.

7. If necessary, configure the CA profile with CRL locations. See "Example: Configuring a Certificate Authority Profile with CRL Locations" on page 75

# Self-Signed Digital Certificates

**IN THIS SECTION**


- Understanding Self-Signed Certificates | **37**
- Example: Generating a Public-Private Key Pair | **38**
- Example: Manually Generating Self-Signed Certificates | **40**
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) | **41**


A self-signed certificate is a certificate that is signed by the same entity who created it rather than by a Certificate Authority (CA). Junos OS provides two methods for generating a self-signed certificate - automatic generation and manual generation.

## Understanding Self-Signed Certificates

A self-signed certificate is a certificate that is signed by its creator rather than by a Certificate Authority (CA).

Self-signed certificates allow for use of SSL-based (Secure Sockets Layer) services without requiring that the user or administrator to undertake the considerable task of obtaining an identity certificate signed by a CA.

Self-signed certificates do not provide additional security as do those generated by CAs. This is because a client cannot verify that the server connected to is the one advertised in the certificate.

Junos OS provides two methods for generating a self-signed certificate:

- Automatic generation

  In this case, the creator of the certificate is the Juniper Networks device. An automatically generated self-signed certificate is configured on the device by default.

  After the device is initialized, it checks for the presence of an automatically generated self-signed certificate. If it does not find one, the device generates one and saves it in the file system.

- Manual generation

  In this case, you create the self-signed certificate for the device.

  At any time, you can use the CLI to generate a self-signed certificate. These certificates are also used to gain access to SSL services.

Self-signed certificates are valid for five years from the time they were generated.

An automatically generated self-signed certificate allows for use of SSL-based services without requiring that the administrator obtain an identity certificate signed by a CA.

A self-signed certificate that is automatically generated by the device is similar to a Secure Shell (SSH) host key. It is stored in the file system, not as part of the configuration. It persists when the device is rebooted, and it is preserved when a `request system snapshot` command is issued.

A self-signed certificate that you manually generate allows for use of SSL-based services without requiring that you obtain an identity certificate signed by a CA. A manually generated self-signed certificate is one example of a public key infrastructure (PKI) local certificate. As is true of all PKI local certificates, manually generated self-signed certificates are stored in the file system.

**SEE ALSO**

PKI in Junos OS | 31

## Example: Generating a Public-Private Key Pair

**IN THIS SECTION**


- Requirements | 39
- Overview | 39
- Configuration | 39

- Verification | 39


This example shows how to generate a public-private key pair.

## Requirements

No special configuration beyond device initialization is required before configuring this feature.

## Overview

In this example, you generate a public-private key pair named self-cert.

## Configuration

**IN THIS SECTION**


- Procedure | 39


### Procedure

### Step-by-Step Procedure

To generate a public-private key pair:

- Create a certificate key pair.

```
user@host> request security pki generate-key-pair certificate-id self-cert
```

## Verification

After the public-private key pair is generated, the Juniper Networks device displays the following:

```
generated key pair ca-ipsec, key size 1024 bits
```

SEE ALSO

Example: Verifying Certificate Validity | 77

# Example: Manually Generating Self-Signed Certificates

IN THIS SECTION


- Requirements | 40
- Overview | 40
- Configuration | 40
- Verification | 41


This example shows how to generate self-signed certificates manually.

## Requirements

Before you begin, generate a public private key pair. See "Digital Certificates" on page 36.

## Overview

For a manually generated self-signed certificate, you specify the distinguished name (DN) when you create it. For an automatically generated self-signed certificate, the system supplies the DN, identifying itself as the creator.

In this example, you generate a self-signed certificate with the e-mail address as `mholmes@example.net`. You specify a certificate-id of *self-cert* to be referenced by web management.

## Configuration

IN THIS SECTION


- Procedure | 41

**Procedure**

### Step-by-Step Procedure

To generate the self-signed certificate manually, enter the following command in operational mode:

```
user@host> request security pki local-certificate generate-self-signed certificate-id self-cert
subject CN=abc  domain-name example.net ip-address 1.2.3.4 email mholmes@example.net
```

To specify the manually generated self-signed certificate for Web management HTTPS services, enter the following command in configuration mode:

```
[edit]
user@host# set system services web-management https local-certificate self-cert
```

### Verification

To verify the certificate is properly generated and loaded, enter the following command in operational mode:

```
user@host> show security pki local-certificate
```

Notice the `Certificate identifier` information for `Issued to`, `validity`, `algorithm`, and `keypair location` details in the displayed output.

To verify the certificate that is associated with the web management, enter the following command in configuration mode:

```
user@host# show system services web-management https local-certificate
```

## Using Automatically Generated Self-Signed Certificates (CLI Procedure)

After the device is initialized, it checks for the presence of a self-signed certificate. If a self-signed certificate is not present, the device automatically generates one. If the device is rebooted, a self-signed certificate is automatically generated at boot time.

To check the system-generated certificate, run the following command in operational mode:

```
user@host> show security pki local-certificate system-generated
```

Notice the `Certificate identifier` details in the output. It displays the following details distinguished name (DN) for the automatically generated certificate:

- `CN = device serial number`
- `CN = system generated`
- `CN = self-signed`

Use the following command in configuration mode to specify the automatically generated self-signed certificate to be used for Web management HTTPS services:

```
[edit]
user@host# set system services web-management https system-generated-certificate
```

Use the following operational command to delete the automatically generated self-signed certificate:

```
user@host# exit
user@host> clear security pki local-certificate system-generated
```

After you delete the system-generated self-signed certificate, the device automatically generates a new one and saves it in the file system.

## RELATED DOCUMENTATION

PKI in Junos OS | 31

# Certificate Authority

IN THIS SECTION


- Configuring a Trusted CA Group | 43
- Understanding Certificate Authority Profiles | 46
- Example: Configuring a CA Profile | 47
- Example: Configuring an IPv6 address as the Source Address for a CA Profile | 50


A certificate authority (CA) profile define every parameter associated with a specific certificate to establish secure connection between two endpoints. The profiles specify which certificates to use, how to verify certificate revocation status, and how that status constrains access.

## Configuring a Trusted CA Group

IN THIS SECTION


- Creating a Trusted CA Group for a List of CA Profiles | 43
- Deleting a CA Profile from a Trusted CA Group | 45
- Deleting a Trusted CA Group | 46


This section describes the procedure to create a trusted CA group for a list of CA profiles and delete a trusted CA group.

### Creating a Trusted CA Group for a List of CA Profiles

You can configure and assign a trusted CA group to authorize an entity. When a peer tries to establish a connection with a client, only the certificate issued by that particular trusted CA of that entity gets validated. The device validates if the issuer of the certificate and the one presenting the certificate belongs to the same client network. If the issuer and the presenter belong to the same client network then the connection is established. If not, the connection will not be established.

Before you begin, you must have a list of all the CA profiles you want to add to the trusted group.

In this example, we are creating three CA profiles named `orgA-ca-profile`, `orgB-ca-profile`, and `orgC-ca-profile` and associating the following CA identifiers `ca-profile1`, `ca-profile2`, and `ca-profile3` for the respective profiles. You can group all the three CA profiles to belong to a trusted CA group `orgABC-trusted-ca-group`.

You can configure a maximum of 20 CA profiles for a trusted CA group.

1. Create CA profiles and associate CA identifiers to the profile.

```
[edit]
user@host# set security pki ca-profile orgA-ca-profile ca-identity ca-profile1
user@host# set security pki ca-profile orgB-ca-profile ca-identity ca-profile2
user@host# set security pki ca-profile orgC-ca-profile ca-identity ca-profile3
```

2. Group the CA profiles under a trusted CA group.

```
[edit]
set security pki trusted-ca-group orgABC-trusted-ca-group ca-profiles [orgA-ca-profile orgB-
ca-profile orgC-ca-profile]]
```

3. Commit the configuration when you are done configuring the CA profiles and the trusted CA groups.

```
[edit]
user@host# commit
```

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

```
user@host# show security pki
ca-profile orgA-ca-profile {
    ca-identity ca-profile1;
}
ca-profile orgB-ca-profile {
    ca-identity ca-profile2;
}
ca-profile orgC-ca-profile {
    ca-identity ca-profile3;
}
trusted-ca-group orgABC-trusted-ca-group {
```

```
    ca-profiles [ orgA-ca-profile orgB-ca-profile orgC-ca-profile ];
}
```

The `show security pki` command displays all the CA profiles that are grouped under the `orgABC_trusted-ca-group`.

## Deleting a CA Profile from a Trusted CA Group

You can delete a specific CA profile in a trusted CA group or you can delete the trusted CA group itself.

For example, if you want to delete a CA profile named `orgC-ca-profile` from a trusted CA group `orgABC-trusted-ca-group`, configured on your device as shown in "Configuring a Trusted CA Group" on page 43 topic perform the following steps:

1. Delete a CA profile from the trusted CA group.

```
[edit]
user@host# delete security pki trusted-ca-group orgABC-trusted-ca-group ca-profiles orgC-ca-profile
```

2. If you are done deleting the CA profile from the trusted CA group, commit the configuration.

```
[edit]
user@host# commit
```

To view the `orgC-ca-profile` being deleted from the `orgABC-trusted-ca-group` , run the `show security pki` command.

```
user@host# show security pki
ca-profile orgA-ca-profile {
    ca-identity ca-profile1;
}
ca-profile orgB-ca-profile {
    ca-identity ca-profile2;
}
trusted-ca-group orgABC-trusted-ca-group {
    ca-profiles [ orgA-ca-profile orgB-ca-profile ];
}
```

The output does not display the `orgC-ca-profile` profile as it is deleted from the trusted CA group.

### Deleting a Trusted CA Group

An entity can support many trusted CA groups and you can delete any trusted CA group for an entity.

For example, if you want to delete a trusted CA group named `orgABC-trusted-ca-group`, configured on your device as shown in "Configuring a Trusted CA Group" on page 43 topic perform the following steps:

1. Delete a trusted CA group.

```
[edit]
user@host# delete security pki trusted-ca-group orgABC-trusted-ca-group
```

2. If you are done deleting the CA profile from the trusted CA group, commit the configuration.

```
[edit]
user@host# commit
```

To view the `orgABC-trusted-ca-group` being deleted from the entity , run the `show security pki` command.

```
user@host# show security pki
ca-profile orgA-ca-profile {
    ca-identity ca-profile1;
}
ca-profile orgB-ca-profile {
    ca-identity ca-profile2;
}
```

The output does not display the `orgABC-trusted-ca-group` as it is deleted from the entity.

RELATED DOCUMENTATION

Understanding Certificate Authority Profiles | 46

## Understanding Certificate Authority Profiles

A certificate authority (CA) profile configuration contains information specific to a CA. You can have multiple CA profiles on an SRX Series Firewall. For example, you might have one profile for orgA and one

for orgB. Each profile is associated with a CA certificate. If you want to load a new CA certificate without removing the older one then create a new CA profile (for example, Microsoft-2008).

Starting with Junos OS Release 18.1R1, the CA server can be an IPv6 CA server.

The PKI module supports IPv6 address format to enable the use of SRX Series Firewalls in networks where IPv6 is the only protocol used.

A CA issues digital certificates, which helps to establish secure connection between two endpoints through certificate validation. You can group multiple CA profiles in one trusted CA group for a given topology. These certificates are used to establish a connection between two endpoints. To establish IKE or IPsec, both the endpoints must trust the same CA. If either of the endpoints are unable to validate the certificate using their respective trusted CA (ca-profile) or trusted CA group, the connection is not established. A minimum of one CA profile is mandatory to create a trusted CA group and maximum of 20 CAs are allowed in one trusted CA group. Any CA from a particular group can validate the certificate for that particular endpoint.

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. A group of trusted CA servers can be created with the `trusted-ca-group` configuration statement at the [`edit security pki`] hierarchy level; one or multiple CA profiles can be specified. The trusted CA server is bound to the IKE policy configuration for the peer at [`edit security ike policy` *`policy`* `certificate`] hierarchy level.

If proxy profile is configured in CA profile, the device connects to the proxy host instead of the CA server while certificate enrollment, verification or revocation. The proxy host communicates with the CA server with the requests from the device, and then relay the response to the device.

CA proxy profile supports SCEP, CMPv2, and OCSP protocols.

CA proxy profile is supported only on HTTP and is not supported on HTTPS protocol.

**SEE ALSO**

PKI Components In Junos OS | 33

## Example: Configuring a CA Profile

**IN THIS SECTION**

Requirements | 48

- Overview | 48
- Configuration | 48
- Verification | 50


This example shows how to configure a CA profile.

## Requirements

No special configuration beyond device initialization is required before configuring this feature.

## Overview

In this example, you create a CA profile called `ca-profile-ipsec` with CA identity microsoft-2008. You then create proxy profile to the CA profile. The configuration specifies that the CRL be refreshed every 48 hours, and the location to retrieve the CRL is `http://www.my-ca.com`. Within the example, you set the enrollment retry value to 20. (The default retry value is 10.)

Automatic certificate polling is set to every 30 minutes. If you configure retry only without configuring a retry interval, then the default retry interval is 900 seconds (or 15 minutes). If you do not configure retry or a retry interval, then there is no polling.

## Configuration

**IN THIS SECTION**


- Procedure | 48


### Procedure

### Step-by-Step Procedure

To configure a CA profile:

1. Create a CA profile.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008
user@host#
```

2. Optionally, configure the proxy profile to the CA profile.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec proxy-profile px-profile
```

   Public key infrastructure (PKI) uses proxy profile configured at the system-level. The proxy profile being used in the CA profile must be configured at the `[edit services proxy]` hierarchy. There can be more than one proxy profile configured under `[edit services proxy]` hierarchy. Each CA profile is referred to the most one such proxy profile. You can configure host and port of the proxy profile at the `[edit system services proxy]` hierarchy.

3. Create a revocation check to specify a method for checking certificate revocation.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008 revocation-
check crl
```

4. Set the refresh interval, in hours, to specify the frequency in which to update the CRL. The default values are next-update time in CRL, or 1 week, if no next-update time is specified.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008 revocation-
check crl refresh-interval 48 url http://www.my-ca.com/my-crl.crl
```

5. Specify the enrollment retry value.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry 20
```

6. Specify the time interval in seconds between attempts to automatically enroll the CA certificate online.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry-interval 1800
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the `show security pki` command.

## Example: Configuring an IPv6 address as the Source Address for a CA Profile

This example shows how to configure an IPv6 address as the source address for a CA profile.

No special configuration beyond device initialization is required before configuring this feature.

In this example, create a CA profile called `orgA-ca-profile` with CA identity `v6-ca` and set the source address of the CA profile to be an IPv6 address, such as `2001:db8:0:f101::1`. You can configure the enrollment URL to accept an IPv6 address `http://[2002:db8:0:f101::1]:/.../`.

1. Create a CA profile.

```
[edit]
user@host# set security pki ca-profile orgA-ca-profile ca-identity v6_ca
```

2. Configure the source address of the CA profile to be an IPv6 address.

```
[edit]
user@host# set security pki ca-profile v6_ca source-address 2001:db8:0:f101::1
```

3. Specify the enrollment parameters for the CA.

```
[edit]
user@host# set security pki ca-profile v6_ca enrollment url http://[2002:db8:0:f101::1]:/.../
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

**SEE ALSO**

Example: Configuring a Certificate Authority Profile with CRL Locations | 75

**Release History Table**

| Release | Description |
|---|---|
| 18.1R1 | Starting with Junos OS Release 18.1R1, the CA server can be an IPv6 CA server. |
| 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**

Self-Signed Digital Certificates | 37

# Certificate Enrollment

**IN THIS SECTION**


- Enrolling Digital Certificates Online: Configuration Overview | 52
- Understanding Online CA Certificate Enrollment | 53
- Understanding Local Certificate Requests | 53

- Enrolling a CA Certificate Online Using SCEP | 53
- Example: Enrolling a Local Certificate Online Using SCEP | 54
- Example: Using SCEP to Automatically Renew a Local Certificate | 57
- Understanding CMPv2 and SCEP Certificate Enrollment | 59
- Understanding Certificate Enrollment with CMPv2 | 59
- Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server | 62
- Example: Loading CA and Local Certificates Manually | 64
- Deleting Certificates (CLI Procedure) | 66


A certificate authority (CA) issues digital certificates, which helps to establish a secure connection between two endpoints through certificate validation. The following topics describe how to configure CA certificates online or local using Simple Certificate Enrollment Protocol (SCEP):

## Enrolling Digital Certificates Online: Configuration Overview

You can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) to enroll digital certificates. To enroll a certificate online:

1. Generate a key pair on the device. See "Self-Signed Digital Certificates" on page 37.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 47.
3. For SCEP only, enroll the CA certificate. See "Enrolling a CA Certificate Online Using SCEP" on page 53.
4. Enroll the local certificate from the CA whose CA certificate you have previously loaded. See "Example: Enrolling a Local Certificate Online Using SCEP" on page 54.
5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 57.

## Understanding Online CA Certificate Enrollment

With Simple Certificate Enrollment Protocol (SCEP), you can configure your Juniper Networks device to obtain a certificate authority (CA) certificate online and start the online enrollment for the specified certificate ID. The CA public key verifies certificates from remote peers.

## Understanding Local Certificate Requests

When you create a local certificate request, the device generates an end entity certificate in PKCS #10 format from a key pair you previously generated using the same certificate ID.

A subject name is associated with the local certificate request in the form of a common name (CN), organizational unit (OU), organization (O), locality (L), state (ST), country (C), and domain component (DC). Additionally, a subject alternative name is associated in the following form:

- IP address
- E-mail address
- Fully qualified domain name (FQDN)

  Specify the subject name in the distinguished name format in quotation marks, including the domain component (DC), common name (CN), serial number (SN), organizational unit name (OU), organization name (O), locality (L), state (ST), and country (C).

  Some CAs do not support an e-mail address as the domain name in a certificate. If you do not include an e-mail address in the local certificate request, you cannot use an e-mail address as the local IKE ID when configuring the device as a dynamic peer. Instead, you can use a fully qualified domain name (if it is in the local certificate), or you can leave the local ID field empty. If you do not specify a local ID for a dynamic peer, enter the *hostname.domain-name* of that peer on the device at the other end of the IPsec tunnel in the peer ID field.

## Enrolling a CA Certificate Online Using SCEP

Before you begin:

1. Generate a public and private key pair. See "Self-Signed Digital Certificates" on page 37.
2. Create a CA profile. See "Example: Configuring a CA Profile" on page 47.

To enroll a CA certificate online:

1. Retrieve the CA certificate online using SCEP. (The attributes required to reach the CA server are obtained from the defined CA profile.)

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

The command is processed synchronously to provide the fingerprint of the received CA certificate.

```
Fingerprint:
e6:fa:d6:da:e8:8d:d3:00:e8:59:12:e1:2c:b9:3c:c0:9d:6c:8f:8d (sha1)
82:e2:dc:ea:48:4c:08:9a:fd:b5:24:b0:db:c3:ba:59 (md5)
Do you want to load the above CA certificate ? [yes,no]
```

2. Confirm that the correct certificate is loaded. The CA certificate is loaded only when you type **yes** at the CLI prompt.

For more information on the certificate, such as the bit length of the key pair, use the command `show security pki ca-certificate`.

## Example: Enrolling a Local Certificate Online Using SCEP

**IN THIS SECTION**


- Requirements | 54
- Overview | 55
- Configuration | 55
- Verification | 56


This example shows how to enroll a local certificate online using Simple Certificate Enrollment Protocol (SCEP).

### Requirements

Before you begin:

- Generate a public and private key pair. See "Self-Signed Digital Certificates" on page 37.

- Configure a certificate authority profile. See "Example: Configuring a CA Profile" on page 47.
- For SCEP, enroll the CA certificate. See "Enrolling a CA Certificate Online Using SCEP" on page 53.

## Overview

In this example, you configure your Juniper Networks device to obtain a local certificate online and start the online enrollment for the specified certificate ID with SCEP. You specify the URL path to the CA server in the CA profile name `ca-profile-ipsec`.

You use the `request security pki local-certificate enroll scep` command to start the online enrollment for the specified certificate ID. (Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the `scep` keyword is supported and required.) You must specify the CA profile name (for example, `ca-profile-ipsec`), the certificate ID corresponding to a previously generated key-pair (for example, `qqq`), and the following information:

- The challenge password provided by the CA administrator for certificate enrollment and reenrollment.
- At least one of the following values:
  - The domain name to identify the certificate owner in IKE negotiations—for example, `qqq.example.net`.
  - The identity of the certificate owner for IKE negotiation with the e-mail statement—for example, `qqq@example.net`.
  - The IP address if the device is configured for a static IP address—for example, `10.10.10.10`.

Specify the subject name in the distinguished name format in quotation marks, including the domain component (DC), common name (CN), serial number (SN), organizational unit name (OU), organization name (O), locality (L), state (ST), and country (C).

Once the device certificate is obtained and the online enrollment begins for the certificate ID. The command is processed asynchronously.

## Configuration

**IN THIS SECTION**

Procedure | 56

**Procedure**

### Step-by-Step Procedure

To enroll a local certificate online:

1. Specify the CA profile.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec  enrollment url path-to-ca-server
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

3. Initiate the enrollment process by running the operational mode command.

```
user@host> request security pki local-certificate enroll scep ca-profile ca-profile-ipsec
certificate-id qqq challenge-password ca-provided-password domain-name qqq.example.net email
qqq@example.net ip-address 10.10.10.10 subject DC=example, CN=router3, SN, OU=marketing,
O=example, L=sunnyvale, ST=california, C=us
```

   If you define SN in the subject field without the serial number, then the serial number is read directly from the device and added to the certificate signing request (CSR).

Starting in Junos OS Release 19.4R2, a warning message `ECDSA Keypair not supported with SCEP for cert_id <certificate id>` is displayed when you try to enroll local certificate using an Elliptic Curve Digital Signature Algorithm (ECDSA) key with Simple Certificate Enrollment Protocol (SCEP) as ECDSA key is not supported with SCEP.

## Verification

To verify the configuration is working properly, enter the `show security pki` command.

# Example: Using SCEP to Automatically Renew a Local Certificate

IN THIS SECTION


- Requirements | 57
- Overview | 57
- Configuration | 58
- Verification | 58


You can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) to enroll digital certificates. This example shows how to renew the local certificates automatically using SCEP.

## Requirements

Before you begin:

- Obtain a certificate either on line or manually. See "Digital Certificates" on page 36.
- Obtain a local certificate. See "Example: Enrolling a Local Certificate Online Using SCEP" on page 54.

## Overview

You can enable the device to automatically renew certificates that were acquired by online enrollment or loaded manually. Automatic certificate renewal saves you from having to remember to renew certificates on the device before they expire, and helps to maintain valid certificates at all times.

Automatic certificate renewal is disabled by default. You can enable automatic certificate renewal and configure the device to automatically send out a request to reenroll a certificate before it expires. You can specify when the certificate reenrollment request is to be sent; the trigger for reenrollment is the percentage of the certificate's lifetime that remains before expiration. For example, if the renewal request is to be sent when the certificate's remaining lifetime is 10 percent, then configure 10 for the reenrollment trigger.

For this feature to work, the device must be able to reach the CA server, and the certificate must be present on the device during the renewal process. Furthermore, you must also ensure that the CA issuing the certificate can return the same DN. The CA must not modify the subject name or alternate subject name extension in the new certificate.

You can enable and disable automatic SCEP certificate renewal either for all SCEP certificates or on a per-certificate basis. You use the `set security pki auto-re-enrollment scep` command to enable and configure

certificate reenrollment. In this example, you specify the certificate ID of the CA certificate as `ca-ipsec` and set the CA profile name associated with the certificate to `ca-profile-ipsec`. You set the challenge password for the CA certificate to the challenge password provided by the CA administrator; this password must be the same one configured previously for the CA. You also set the percentage for the reenrollment trigger to `10`. During automatic reenrollment, the Juniper Networks device by default uses the existing key pair. A good security practice is to regenerate a new key pair for reenrollment. To generate a new key pair, use the `re-generate-keypair` command.

## Configuration

**IN THIS SECTION**


- Procedure | 58


### Procedure

### Step-by-Step Procedure

To enable and configure local certificate reenrollment:

1. To enable and configure certificate reenrollment.

```
[edit]
user@host# set security pki auto-re-enrollment scep certificate-id ca-ipsec  ca-profile-name
ca-profile-ipsec  challenge-password ca-provided-password re-enroll-trigger-time-percentage
10 re-generate-keypair
```

   Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the `scep` keyword is supported and required.

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the `show security pki local-certificate detail` operational mode command.

## Understanding CMPv2 and SCEP Certificate Enrollment

Based on your deployment environment, you can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) for online certificate enrollment. This topic describes some of the basic differences between the two protocols.

Table 2 on page 59 describes the differences between the CMPv2 and SCEP certificate enrollment protocols.

**Table 2: Comparison of CMPv2 and SCEP Certificate Enrollment**

| Attribute | CMPv2 | SCEP |
|---|---|---|
| Supported certificate types: | DSA, ECDSA, and RSA | RSA only |
| Supported standards | RFCs 4210 and 4211 | Internet Engineering Task Force draft |

Certificate enrollment and reenrollment requests and responses differ between CMPv2 and SCEP. With CMPv2, there is no separate command to enroll CA certificates. With SCEP, you enroll CA certificates with the `request security pki ca-certificate enroll` command and specify the CA profile. A CA profile must be configured with either CMPv2 or SCEP.

## Understanding Certificate Enrollment with CMPv2

**IN THIS SECTION**


- Certificate Enrollment and Reenrollment Messages | 60
- End-Entity Certificate with Issuer CA Certificate | 60
- End-Entity Certificate with CA Certificate Chain | 60


The `request security pki local-certificate enroll cmpv2` command uses CMPv2 to enroll a local digital certificate online. This command loads both end-entity and CA certificates based on the CA server configuration. The CA profile must be created prior to CA certificate enrollment because the enrollment URL is extracted from the CA profile.

This topic describes certificate enrollment with the CMPv2 protocol.

## Certificate Enrollment and Reenrollment Messages

The CMPv2 protocol mainly involves certificate enrollment and reenrollment operations. The certificate enrollment process includes Initialization Request and Initialization Response messages, while certificate reenrollment includes Key Update Request and Key Update Response messages.

CMPv2 server responds back with Initialization Response (IP). The response contains end-entity certificate along with optional CA certificates. The message integrity and message authenticity of Initialization Response can be verified using shared-secret-information according to RFC 4210. The Initialization Response can also be verified using issuer CA public-key. Before you re-enroll an end-entity certificate, you must have a valid CA certificate enrolled on the device.

The Initialization Response or Key Update Response message can contain an issuer CA certificate or a chain of CA certificates. The CA certificates received in the responses are treated as trusted CA certificates and stored in the receiving device if they are not already present in the trusted CA store. These CA certificates are later used for end-entity certificate validation.

We do not support CA certificate re-enrollment. If a CA certificate expires, you must unenroll the current CA certificate and enroll it back again.

## End-Entity Certificate with Issuer CA Certificate

In a simple scenario, the Initialization Response message might contain only an end-entity certificate, in which case the CA information is provided separately. The certificate is stored in the end-entity certificate store.

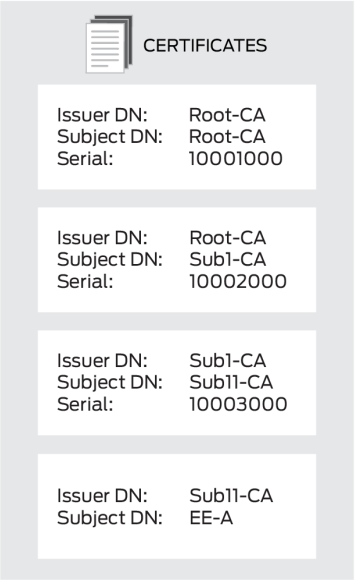
The Initialization Response message can contain an end-entity certificate as well as a self-signed issuer CA certificate. The end-entity certificate is first stored in the certificate store, and then the CA certificate is checked. If the CA certificate is found and the subject distinguished name (DN) of the CA certificate in the Initialization Response message matches the issuer DN of the end-entity certificate, the CA certificate is stored in the CA certificate store for the CA profile name specified in the CMPv2 certificate enrollment command. If the CA certificate already exists in the CA certificate store, no action is taken.

## End-Entity Certificate with CA Certificate Chain

In many deployments, the end-entity certificate is issued by an intermediate CA in a certificate chain. In this case, the Initialization Response message can contain the end-entity certificate along with a list of CA certificates in the chain. The intermediate CA certificates and the self-signed root CA certificates are all required to validate the end-entity certificate. The CA chain might also be needed to validate certificates received from peer devices with similar hierarchies. The following section describes how certificates in the CA chain are stored.

In Figure 4 on page 61, the Initialization Response message includes the end-entity certificate and three CA certificates in a certificate chain.

**Figure 4: End-Entity Certificate with CA Certificate Chain**




The end-entity certificate is stored in the end-entity certificate store. Each CA certificate needs a CA profile. The CA certificate with the subject DN Sub11-CA is the first CA in the chain and is the issuer of the end-entity certificate. It is stored in the CA profile that is specified with the CMPv2 certificate enrollment command.

Each of the remaining CA certificates in the chain is checked for its presence in the CA store. If a CA certificate is not present in the CA store, it is saved and a CA profile is created for it. The new CA profile name is created using the least significant 16 digits of the CA certificate serial number. If the serial number is longer than 16 digits, the most significant digits beyond 16 digits are truncated. If the serial number is shorter than 16 digits, the remaining most significant digits are filled with `0`s. For example, if the serial number is 11111000100010001000, then the CA profile name is `1000100010001000`. If the serial number is 10001000, then the CA profile name is `0000000010001000`.

It is possible that multiple certificate serial numbers can have the same least significant 16 digits. In that case, `-00` is appended to the profile name to create a unique CA profile name; additional CA profile names are created by incrementing the appended number, from `-01` up to `-99`. For example, CA profile names can be `1000100010001000`, `1000100010001000-00`, and `1000100010001000-01`.

SEE ALSO

Understanding Certificate Authority Profiles | 46

IKE Authentication (Certificate-Based Authentication)

# Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server

IN THIS SECTION


- Requirements | 62
- Overview | 62
- Configuration | 63
- Verification | 63


This example shows how to generate a certificate signing request manually.

## Requirements

Generate a public and private key. See "Self-Signed Digital Certificates" on page 37.

## Overview

In this example, you generate a certificate request using the certificate ID of a public-private key pair you previously generated (ca-ipsec). Then you specify the domain name (example.net) and the associated common name (abc). The certificate request is displayed in PEM format.

You copy the generated certificate request and paste it into the appropriate field at the CA website to obtain a local certificate. (Refer to the CA server documentation to determine where to paste the certificate request.) When the PKCS #10 content is displayed, the MD5 hash and SHA-1 hash of the PKCS #10 file is also displayed.

## Configuration

IN THIS SECTION

- Procedure | 63

### Procedure

### Step-by-Step Procedure

To generate a local certificate manually:

- Specify certificate ID, domain name, and common name.

```
user@host> request security pki generate-certificate-request certificate-id ca-ipsec domain-
name example.net subject CN=abc
```

## Verification

To view the certificate signing request, enter the `show security pki certificate-request detail` command.

```
Certificate identifier: ca-ipsec
Certificate version: 1
Issued to: CN = abc
Public key algorithm: rsaEncryption(1024 bits)
30:81:89:02:81:81:00:da:ea:cd:3a:49:1f:b7:33:3c:c5:50:fb:57
de:17:34:1c:51:9b:7b:1c:e9:1c:74:86:69:a4:36:77:13:a7:10:0e
52:f4:2b:52:39:07:15:3f:39:f5:49:d6:86:70:4b:a6:2d:73:b6:68
39:d3:6b:f3:11:67:ee:b4:40:5b:f4:de:a9:a4:0e:11:14:3f:96:84
03:3c:73:c7:75:f5:c4:c2:3f:5b:94:e6:24:aa:e8:2c:54:e6:b5:42
c7:72:1b:25:ca:f3:b9:fa:7f:41:82:6e:76:8b:e6:d7:d2:93:9b:38
fe:fd:71:01:2c:9b:5e:98:3f:0c:ed:a9:2b:a7:fb:02:03:01:00:01
Fingerprint:
0f:e6:2e:fc:6d:52:5d:47:6e:10:1c:ad:a0:8a:4c:b7:cc:97:c6:01 (sha1)
f8:e6:88:53:52:c2:09:43:b7:43:9c:7a:a2:70:98:56 (md5)
```

# Example: Loading CA and Local Certificates Manually

IN THIS SECTION


- Requirements | 64
- Overview | 64
- Configuration | 65
- Verification | 65


This example shows how to load CA and local certificates manually.

## Requirements

Before you begin:

- Generate a public-private key pair. See "Self-Signed Digital Certificates" on page 37.
- Create a CA profile. See "Understanding Certificate Authority Profiles" on page 46.

  CA Profile is only required for the CA certificate and not for the local certificate
- Generate a certificate request. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.

## Overview

In this example, you download the local.cert and ca.cert certificates and save them to the /var/tmp/ directory on the device.

After you download certificates from a CA, you transfer them to the device (for example, using FTP), and then load them.

You can load the following certificate files onto a device running Junos OS:

- A local or end-entity (EE) certificate that identifies your local device. This certificate is your public key.
- A CA certificate that contains the CA's public key.
- A CRL that lists any certificates revoked by the CA.

  You can load multiple EE certificates onto the device.

## Configuration

IN THIS SECTION

- Procedure | 65

### Procedure

### Step-by-Step Procedure

To load the certificate files onto a device:

1. Load the local certificate.

```
[edit]
user@host> request security pki local-certificate load certificate-id local.cert
filename /var/tmp/local.cert
```

2. Load the CA certificate.

```
[edit]
user@host> request security pki ca-certificate load ca-profile ca-profile-ipsec
filename /var/tmp/ca.cert
```

3. Examine the fingerprint of the CA certificate, if it is correct for this CA certificate select yes to accept.

## Verification

To verify the certificates loaded properly, enter the `show security pki local-certificate` and `show security pki ca-certificate` commands in operational mode.

```
Fingerprint:
e8:bf:81:6a:cd:26:ad:41:b3:84:55:d9:10:c4:a3:cc:c5:70:f0:7f (sha1)
19:b0:f8:36:e1:80:2c:30:a7:31:79:69:99:b7:56:9c (md5)
Do you want to load this CA certificate ? [yes,no] (no) yes
```

SEE ALSO

Example: Using SCEP to Automatically Renew a Local Certificate | 57

Example: Verifying Certificate Validity | 77

Example: Configuring a Certificate Authority Profile with CRL Locations | 75

## Deleting Certificates (CLI Procedure)

You can delete a local or trusted CA certificate that is automatically or manually generated.

Use the following command to delete a local certificate:

```
user@host> clear security pki local certificate certificate-id (certificate-id| all | system-generated )
```

Specify a certificate ID to delete a local certificate with a specific ID, use `all` to delete all local certificates, or specify `system-generated` to delete the automatically generated self-signed certificate.

When you delete an automatically generated self-signed certificate, the device generates a new one.

To delete a CA certificate:

```
user@host> clear security pki ca-certificate ca-profile (ca-profile-name | all)
```

Specify a CA profile to delete a specific CA certificate, or use `all` to delete all CA certificates present in the persistent store.

You are asked for confirmation before a CA certificate can be deleted.

**Release History Table**

| Release | Description |
|---|---|
| 19.4R2 | Starting in Junos OS Release 19.4R2, a warning message `ECDSA Keypair not supported with SCEP for cert_id <certificate id>` is displayed when you try to enroll local certificate using an Elliptic Curve Digital Signature Algorithm (ECDSA) key with Simple Certificate Enrollment Protocol (SCEP) as ECDSA key is not supported with SCEP. |
| 15.1X49-D40 | Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the `scep` keyword is supported and required. |

| | |
|---|---|
| 15.1X49-D40 | Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the scep keyword is supported and required. |

# Certificate Revocation

**IN THIS SECTION**


- Understanding Online Certificate Status Protocol and Certificate Revocation Lists | **67**
- Example: Manually Loading a CRL onto the Device | **70**
- Understanding Dynamic CRL Download and Checking | **72**
- Example: Configuring a Certificate Authority Profile with CRL Locations | **75**
- Example: Verifying Certificate Validity | **77**
- Deleting a Loaded CRL (CLI Procedure) | **79**


Digital certificates have an expiration date, however, prior to expiration, a certificate may no longer be valid due to many reasons. You can manage certificate revocations and validations locally and by referencing a Certificate Authority (CA) certificate revocation list (CRL).

## Understanding Online Certificate Status Protocol and Certificate Revocation Lists

**IN THIS SECTION**


- Comparison of Online Certificate Status Protocol and Certificate Revocation List | **69**


OCSP is used to check the revocation status of X509 certificates. OCSP provides revocation status on certificates in real time and is useful in time-sensitive situations such as bank transactions and stock trades.

The revocation status of a certificate is checked by sending a request to an OCSP server that resides outside of an SRX Series Firewall. Based on the response from the server, the VPN connection is allowed or denied. OCSP responses are not cached on SRX Series Firewalls.

The OCSP server can be the *certificate authority (CA)* that issues a certificate or a designated authorized responder. The location of the OCSP server can be configured manually or extracted from the certificate that is being verified. Requests are sent first to OCSP server locations that are manually configured in CA profiles with the `ocsp url` statement at the [`edit security pki ca-profile` *`profile-name`* `revocation-check`] hierarchy level; up to two locations can be configured for each CA profile. If the first configured OCSP server is not reachable, the request is sent to the second OCSP server. If the second OCSP server is not reachable, the request is then sent to the location in the certificate's AuthorityInfoAccess extension field. The `use-ocsp` option must also be configured, as *certificate revocation list (CRL)* is the default checking method.

SRX Series Firewalls accept only signed OCSP responses from the CA or authorized responder. The response received is validated using trusted certificates. The response is validated as follows:

1. The CA certificate enrolled for the configured CA profile is used to validate the response.
2. The OCSP response might contain a certificate to validate the OCSP response. The received certificate must be signed by a CA certificate enrolled in the SRX Series Firewall. After the received certificate is validated by the CA certificate, it is used to validate the OCSP response.

The response from the OCSP server can be signed by different CAs. The following scenarios are supported:

- The CA server that issues the end entity certificate for a device also signs the OCSP revocation status response. The SRX Series Firewall verifies the OCSP response signature using the CA certificate enrolled in the SRX Series Firewall. After the OCSP response is validated, the certificate revocation status is checked.
- An authorized responder signs the OCSP revocation status response. The certificate for the authorized responder and the end entity certificate being verified must be issued by the same CA. The authorized responder is first verified using the CA certificate enrolled in the SRX Series Firewall. The OCSP response is validated using the responder's CA certificate. The SRX Series Firewall then uses the OCSP response to check the revocation status of the end entity certificate.
- There are different CA signers for the end entity certificate being verified and the OCSP response. The OCSP response is signed by a CA in the certificate chain for the end entity certificate being verified. (All peers participating in an IKE negotiation need to have at least one common trusted CA in their respective certificate chains.) The OCSP responder's CA is verified using a CA in the certificate chain. After validating the responder CA certificate, the OCSP response is validated using the responder's CA certificate.

To prevent replay attacks, a *nonce* payload can be sent in an OCSP request. Nonce payloads are sent by default unless it is explicitly disabled. If enabled, the SRX Series Firewall expects the OCSP response to

contain a nonce payload, otherwise the revocation check fails. If OCSP responders are not capable of responding with a nonce payload, then the nonce payload must be disabled on the SRX Series Firewall.

In the normal course of business, certificates are revoked for various reasons. You might wish to revoke a certificate if you suspect that it has been compromised, for example, or when a certificate holder leaves the company.

You can manage certificate revocations and validations in two ways:

- Locally— This is a limited solution.
- By referencing a Certificate Authority (CA) certificate revocation list (CRL)— You can automatically access the CRL online at intervals you specify or at the default interval set by the CA.

In Phase 1 negotiations, SRX Series Firewall verifies the EE certificate received from the peer during an IKE exchange and uses the CRL to make sure the EE certificate is not revoked by its CA.

If a CRL is not loaded on the device and the peer certificate issuer is a trusted CA:

1. Junos OS retrieves the CRL through the configured LDAP or HTTP CRL locations (that is, the CRL Distribution Points (CDP)), if they are defined in the CA profile.
2. If the CRL Distribution Points is not configured in the CA profile, the device uses the CDP extension in a certificate issued by the CA (if present). The certificate issued by the CA can be a certificate enrolled by the administrator or received during the Phase 1 negotiation.

If the EE certificate is not issued by a root CA, the certificates of each intermediate CAs goes through the same verification and revocation check. The CRL of the root CA is used to check if the certificate issued by the root CA is revoked. If the CDP is not configured in the root CA profile, the device uses the CDP extension in the certificate issued by the CA (if present).

The CRL distribution point extension (.cdp) in an X509 certificate can be added to either an HTTP URL or an LDAP URL.

If the certificate does not contain a certificate distribution point extension, and you cannot automatically retrieve the CRL through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP), you can retrieve a CRL manually and load that in the device.

Local certificates are being validated against certificate revocation list (CRL) even when CRL check is disabled. This can be stopped by disabling the CRL check through the Public Key Infrastructure (PKI) configuration. When CRL check is disabled, PKI will not validate local certificate against CRL.

## Comparison of Online Certificate Status Protocol and Certificate Revocation List

Online Certificate Status Protocol (OCSP) and certificate revocation list (CRL) can both be used to check the revocation status of a certificate. There are advantages and disadvantages to each method.

- OCSP provides certificate status in real time, while CRL uses cached data. For time-sensitive applications, OCSP is the preferred approach.
- CRL checking is faster because lookup for certificate status is done on information cached on the VPN device. OCSP requires time to obtain the revocation status from an external server.
- CRL requires additional memory to store the revocation list received from a CRL server. OCSP does not require additional memory to save the revocation status of certificates.
- OCSP requires that the OCSP server be available at all times. CRL can use cached data to check the revocation status of certificates when the server is unreachable.

On MX Series and SRX Series Firewalls, CRL is the default method used to check the revocation status of a certificate.

**SEE ALSO**

Certificate Validation

## Example: Manually Loading a CRL onto the Device

**IN THIS SECTION**


- Requirements | **70**
- Overview | **71**
- Configuration | **71**
- Verification | **71**


This example shows how to load a CRL manually onto the device.

### Requirements

Before you begin:

1. Generate a public and private key pair. See "Self-Signed Digital Certificates" on page 37.
2. Generate a certificate request. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.

3. Configure a certificate authority (CA) profile. See "Example: Configuring a CA Profile" on page 47.

4. Load your certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 64.

## Overview

You can load a CRL manually, or you can have the device load it automatically, when you verify certificate validity. To load a CRL manually, you obtain the CRL from a CA and transfer it to the device (for example, using FTP).

In this example, you load a CRL certificate called `revoke.crl` from the /var/tmp directory on the device. The CA profile is called `ca-profile-ipsec`. (Maximum file size is 5 MB.)

If a CRL is already loaded into the ca-profile the command `clear security pki crl ca-profile ca-profile-ipsec` must be run first to clear the old CRL.

## Configuration

**IN THIS SECTION**

- Procedure | 71

### Procedure

### Step-by-Step Procedure

To load a CRL certificate manually:

1. Load a CRL certificate.

```
[edit]
user@host> request security pki crl load ca-profile ca-profile-ipsec filename /var/tmp/
revoke.crl
```

Junos OS supports loading of CA certificates in X509, PKCS #7, DER, or PEM formats.

## Verification

To verify the configuration is working properly, enter the `show security pki crl` operational mode command.

## Understanding Dynamic CRL Download and Checking

Digital certificates are issued for a set period of time and are invalid after the specified expiration date. A CA can revoke an issued certificate by listing it in a certificate revocation list (CRL). During peer certificate validation, the revocation status of a peer certificate is checked by downloading the CRL from a CA server to the local device.

To facilitate the CRL check for the certificates when a CA profile is not configured, dynamic CA profile is created. A dynamic CA profile is automatically created on the local device with the format `dynamic-nnn`.

A dynamic CA profile:

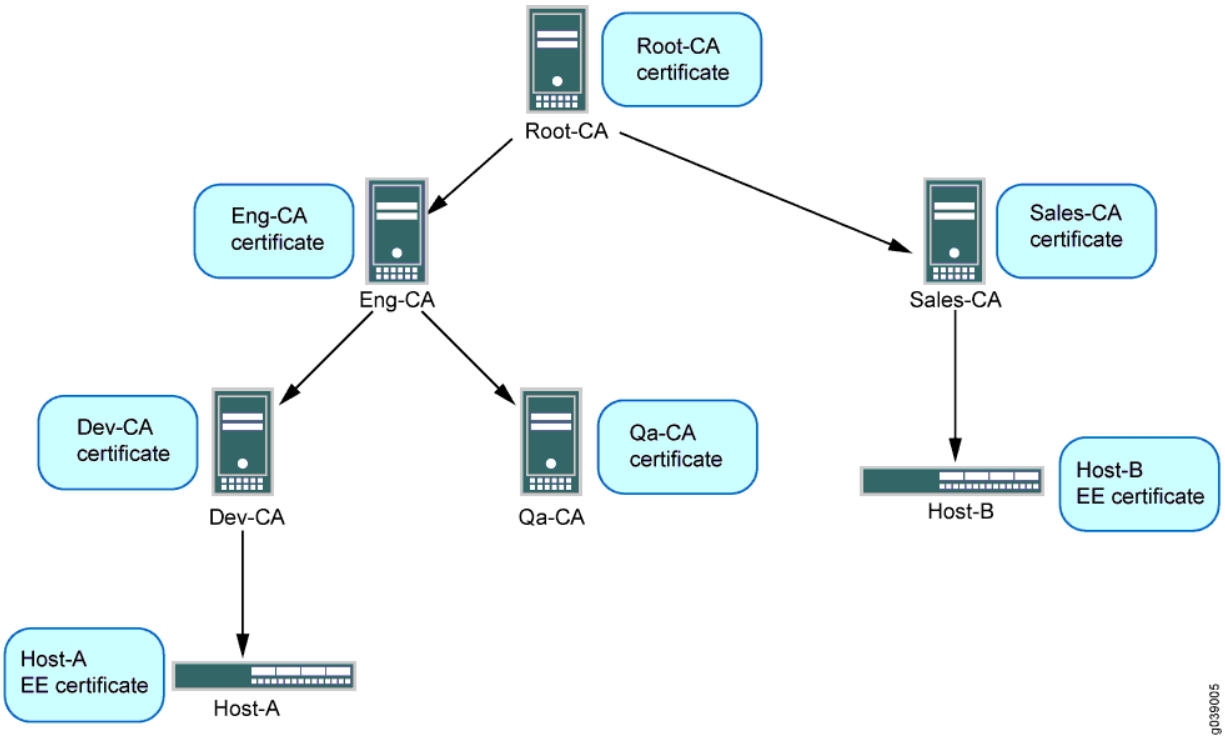
- Allows the local device to download the Dynamic CA and Dynamic CRL (for corresponding CA) as per peer's localcert issuer
- Checks the revocation status of the peer's certificate

A VPN device checks a peer's EE certificate for its revocation status. A VPN device uses the certificate received from its peer to do the following:

- Extract the URL to dynamically download the CA's CRL
- Check the revocation status of the peer's EE certificate

In Figure 5 on page 73, Host-A can use the Sales-CA and EE certificates received from Host-B to dynamically download the CRL for Sales-CA and check the revocation status of Host-B's certificate.

**Figure 5: Multilevel Hierarchy for Certificate-Based Authentication**




In case of single hierarchy CA servers or CA certificate chain, the local EE certificate and the received peer EE certificate are issued from the same CA server.

Following are some of the SRX Series Firewall behavior based on different configurations:

- If you have configured a SRX Series Firewall with a trusted-ca or trusted-ca-group, then the device does not validate or trust any other CAs.
- If you have defined a CA profile that has a chain of CAs where the SRX Series Firewall only trusts the root CA and peer has a certificate signed by a sub-CA to this root, then Dynamic CA and CRL will be added to the device.

Table 3 on page 74 provides few sample scenarios where Dynamic CA or CRL is not created:

**Table 3: Sample Scenarios**

| Scenario | Condition |
|---|---|
| Sample scenario 1 | In the CA profile, you have defined a trusted CA for ca-profile-name, and you receive a connection from a device that has a certificate signed by a different CA that was not defined as a trusted CA in your CA profile. |
| Sample scenario 2 | You have defined a CA profile that has a chain of CAs where the SRX Series Firewall only trust a sub-CA, and peer has a certificate signed by a level above this sub-CA. |

To enable dynamic CA profiles, you must configure the `revocation-check crl` option on a Root-CA profile at the [`edit security pki ca-profile` *`profile-name`*] hierarchy level.

The revocation check properties of a Root-CA profile are inherited for dynamic CA profiles. In Figure 5 on page 73, the CA profile configuration on Host-A for Root-CA enables dynamic CA profiles as shown in the following output:

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

A dynamic CA profile is created on Host-A for Sales-CA. Revocation checking is inherited for the Sales-CA dynamic CA profile from Root-CA.

If the `revocation-check disable` statement is configured in a Root-CA profile, dynamic CA profiles are not created and dynamic CRL download and checking is not performed.

The data for CRLs downloaded from dynamic CA profiles are displayed with the `show security pki crl` command in the same way as CRLs downloaded by configured CA profiles. The CRL from a dynamic CA profile is updated periodically as are those for CA profiles that are configured in the device. The peer CA certificate is also required for signature validation of CRL downloaded from CA server.

The CA certificate is required to validate the CRL received from a CA server; therefore, the CA certificate received from a peer is stored on the local device. The received CA certificate from peer is used to validate the CRL and the certificate it issued. Because the received CA certificate is not enrolled by an administrator, the result of a successful certificate verification is not conclusive until the whole certificate chain up to the root CA is verified. The certificate of the root CA must be enrolled by an administrator.

**SEE ALSO**

PKI in Junos OS | 31

Configuring a Trusted CA Group

Example: Configuring a Device for Peer Certificate Chain Validation | 157

Understanding Certificate Authority Profiles | 46

## Example: Configuring a Certificate Authority Profile with CRL Locations

**IN THIS SECTION**

- Requirements | 75
- Overview | 76
- Configuration | 76
- Verification | 77

This example shows how to configure a certificate authority profile with CRL locations.

### Requirements

Before you begin:

1. Generate a key pair in the device. See "Digital Certificates" on page 36.

2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 47.
3. Obtain a personal certificate from the CA. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.
4. Load the certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 64.
5. Configure automatic reenrollment. See Example: Configuring SecurID User Authentication.
6. If necessary, load the certificate's CRL on the device. See "Example: Manually Loading a CRL onto the Device" on page 70.

## Overview

In this example, you direct the device to check the validity of the CA profile called `my_profile` and, if a CRL did not accompany a CA certificate and is not loaded on the device, to retrieve the CRL from the URL `http://abc/abc-crl.crl`.

## Configuration

**IN THIS SECTION**

- Procedure | 76

### Procedure

### Step-by-Step Procedure

To configure certificate using CRL:

1. Specify the CA profile and URL.

```
[edit]
user@host# set security pki ca-profile my_profile revocation-check crl url http://abc/abc-
crl.crl
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the `show security pki` operational mode command.

#### SEE ALSO

Deleting Certificates (CLI Procedure) | 66

## Example: Verifying Certificate Validity

**IN THIS SECTION**


- Requirements | 77
- Overview | 77
- Configuration | 78
- Verification | 78


This example shows how to verify the validity of a certificate.

### Requirements

No special configuration beyond device initialization is required before configuring this feature.

### Overview

In this example, you verify certificates manually to find out whether a certificate has been revoked or whether the CA certificate used to create a local certificate is no longer present on the device.

When you verify certificates manually, the device uses the CA certificate (`ca-cert`) to verify the local certificate ( `local.cert`). If the local certificate is valid, and if `revocation-check` is enabled in the CA profile,

the device verifies that the CRL is loaded and valid. If the CRL is not loaded and valid, the device downloads the new CRL.

For CA-issued certificates or CA certificates, a DNS must be configured in the device's configuration. The DNS must be able to resolve the host in the distribution CRL and in the CA cert/revocation list url in the ca-profile configuration. Additionally, you must have network reachability to the same host in order for the checks to receive.

## Configuration

**IN THIS SECTION**


- Procedure | 78


### Procedure

### Step-by-Step Procedure

To manually verify the validity of a certificate:

1. Verify the validity of a local certificate.

```
[edit]
user@host> request security pki local-certificate verify certificate-id local.cert
```

2. Verify the validity of a CA certificate.

```
[edit]
user@host> request security pki ca-certificate verify ca-profile ca-profile-ipsec
```

The associated private key and the signature are also verified.

## Verification

To verify the configuration is working properly, enter the `show security pki ca-profile` command.

If an error is returned instead of a positive verification the failure is logged in pkid.

SEE ALSO

Deleting Certificates (CLI Procedure) | 66

## Deleting a Loaded CRL (CLI Procedure)

You can choose to delete a loaded CRL if you no longer need to use it to manage certificate revocations and validation.

Use the following command to delete a loaded certificate revocation list:

```
user@host> clear security pki crl ca-profile (ca-profile all)
```

Specify a CA profile to delete a CRL associated with the CA identified by the profile, or use `all` to delete all CRLs.

SEE ALSO

Deleting Certificates (CLI Procedure) | 66

RELATED DOCUMENTATION

Certificate Authority | 43

# Certificate Validation

IN THIS SECTION


- Understanding Digital Certificate Validation | 80
- Example: Validating Digital Certificate by Configuring Policy OIDs on an SRX Series Firewall | 86

# Understanding Digital Certificate Validation

IN THIS SECTION


- Policy Validation | 80
- Path Length Validation | 83
- Key Usage | 84
- Issuer and Subject Distinguished Name Validation | 85


During IKE negotiation, the PKI daemon on an SRX Series Firewall validates X509 certificates received from VPN peers. The certificate validation performed is specified in RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Basic certificate and certificate chain validations include signature and date validation as well as revocation checks. This topic describes additional digital certificate validations performed by the PKI daemon.

## Policy Validation

X509 certificates can include optional policy validation fields. If a policy validation field is present, policy validation is performed for the entire certificate chain including the end entity (EE) certificate and intermediate certificate authority (CA) certificates. Policy validation is not applicable to the root certificate. Policy validation ensures that the EE and intermediate CA certificates have a common policy. If no common policy exists for the certificate chain being validated, certificate validation fails.

Prior to policy validation, a certificate chain containing the self-signed root certificate, intermediate CA certificates, and EE certificate must be built. The policy validation starts with the intermediate CA certificate issued by the self-signed root certificate and continues through the EE certificate.

The following optional certificate fields are used for policy validation:

- **policy-oids**
- **requireExplicitPolicy**
- **skipCerts**

These fields are described in the following sections.

### Policy OIDs Configured on SRX Series Firewalls

In some situations, it might be desirable to only accept certificates with known policy object identifiers (OIDs) from peers. This optional configuration allows certificate validation to succeed only if the certificate chain received from the peer contains at least one policy OID that is configured on the SRX Series Firewall.

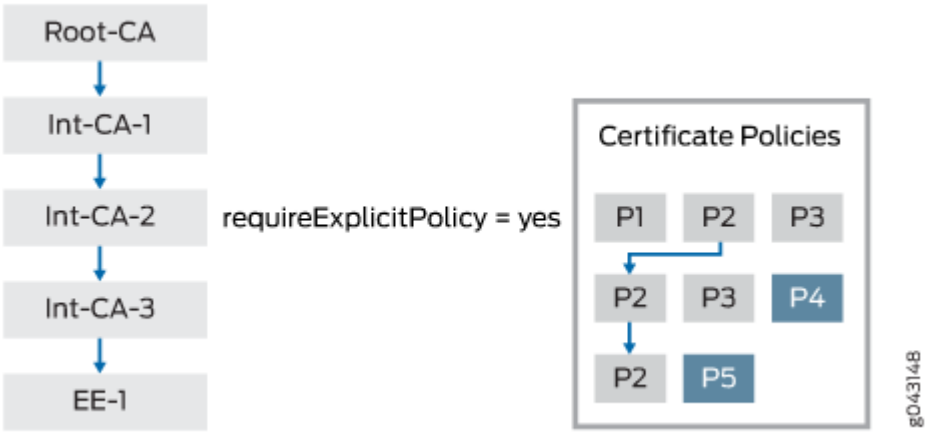
On the SRX Series Firewall, policy OIDs are configured in an IKE policy with the `policy-oids` configuration statement at the [`edit security ike policy` *`policy-name`* `certificate`] hierarchy level. You can configure up to five policy OIDs. For a peer's certificate to be validated successfully, the peer's certificate chain must contain at least one of the policy OIDs configured on the SRX Series Firewall. Note that the **policy-oids** field in a certificate is optional. If you configure policy OIDs on the SRX Series Firewall but the peer's certificate chain does not contain any policy OIDs, certificate validation fails.

### No Policy OIDs Configured on SRX Series Firewalls

If no policy OID is configured on the SRX Series Firewall, policy validation starts whenever the **requireExplicitPolicy** field is encountered in the certificate chain. A certificate can contain one or more certificate policy OIDs. For policy validation to succeed, there must be a common policy OID in the certificate chain.

Figure 6 on page 82 shows a certificate chain that consists of certificates for a root CA, three intermediate CAs, and an EE. The CA certificate for Int-CA-2 contains the **requireExplicitPolicy** field; therefore, policy validation starts with Int-CA-2 and continues through EE-1. The certificate for Int-CA-2 contains policy OIDs P1, P2, and P3. The certificate for Int-CA-3 contains policy OIDs P2, P3, and P4. The certificate for EE-1 contains policy OIDs P2 and P5. Because the policy OID P2 is common to the certificates being validated, policy validation succeeds.

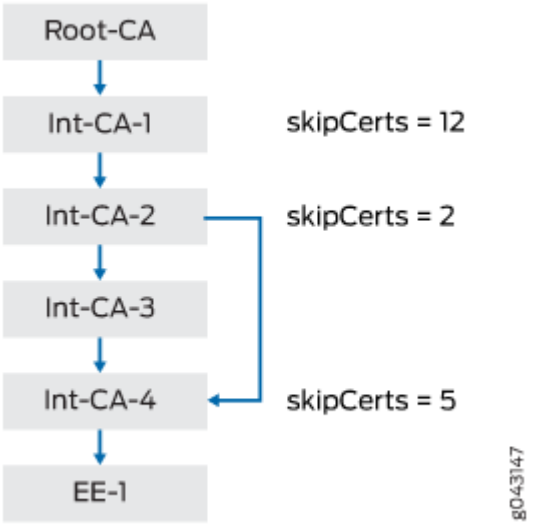
**Figure 6: Policy Validation with requireExplicitPolicy Field**




The optional **skipCerts** field in an intermediate CA certificate indicates the number of certificates, including the current CA certificate, that are to be excluded from policy validation. If **skipCerts** is 0, policy validation starts from the current certificate. If **skipCerts** is 1, the current certificate is excluded from policy validation. The value of the **skipCerts** field is checked in every intermediate CA certificate. If a **skipCerts** value is encountered that is lower than the current number of certificates being excluded, the lower **skipCerts** value is used.

Figure 7 on page 83 shows a certificate chain consisting of a root CA, four intermediate CAs, and an EE. The **skipCerts** value in Int-CA-1 is 12, which skips 12 certificates including the certificate for Int-CA-1. However, the **skipCerts** value is checked in every intermediate CA certificate in the chain. The **skipCerts** value in Int-CA-2 is 2, which is lower than 12, so now 2 certificates are skipped. The **skipCerts** value in Int-CA-4 is 5, which is greater than 2, so the Int-CA-4 **skipCerts** value is ignored.

**Figure 7: Policy Validation with skipCerts Field**




When policy OIDs are configured on the SRX Series Firewall, the certificate fields **requireExplicitPolicy** and **skipCerts** are ignored.

## Path Length Validation

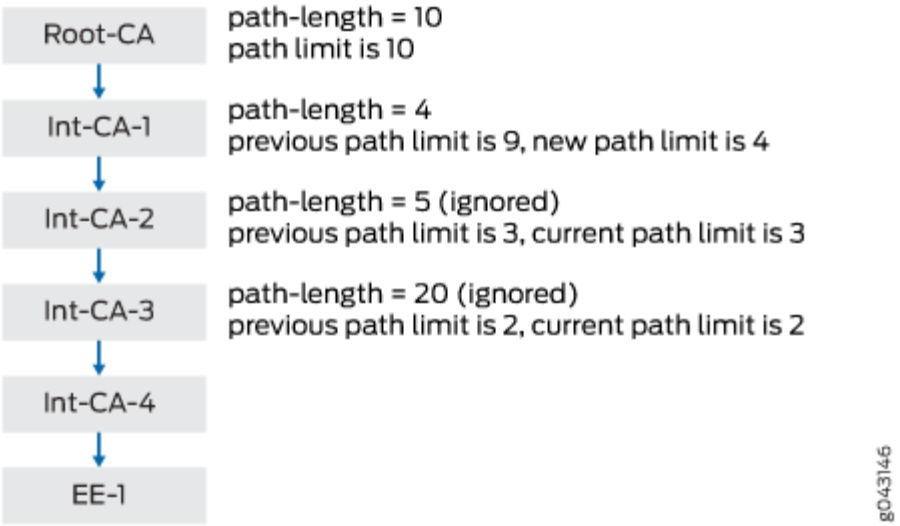
Certificate validation can involve a certificate chain that includes a root CA, one or more optional intermediate CAs, and an EE certificate. The number of intermediate CAs can grow depending upon the deployment scenario. Path length validation provides a mechanism to limit the number of intermediate certificates involved in certificate validation. **path-length** is an optional field in an X509 certificate. The value of **path-length** indicates the number of non-self-signed intermediate CA certificates allowed for certificate validation. The last certificate, which is generally the EE certificate, is not included in the path limit. If the root certificate contains a **path-length** value of 0, no intermediate CA certificates are allowed. If the **path-length** value is 1, there can be 0 or 1 intermediate CA certificates.

**path-length** can be present in multiple CA certificates in the certificate chain. The path length validation always begins with the self-signed root certificate. The path limit is decremented by 1 at each intermediate certificate in the chain. If an intermediate certificate contains a **path-length** value less than the current path limit, the new limit is enforced. On the other hand, if the **path-length** value is larger than the current path limit, it is ignored.

Figure 8 on page 84 shows a certificate chain that consists of a root CA, four intermediate CAs, and an EE. The **path-length** value in Root-CA is 10, therefore the initial path limit of non-self-signed intermediate CA certificates allowed for certificate validation is 10. At Int-CA-1, the path limit is 10-1 or 9. The **path-length** value in Int-CA-1 is 4, which is less than the path limit of 9, so the new path limit becomes 4. At Int-CA-2, the path limit is 4-1 or 3. The **path-length** value in Int-CA-2 is 5, which is larger

than the path limit of 3, so it is ignored. At Int-CA-3, the path limit is 3-1 or 2. The **path-length** value in Int-CA-3 is 20, which is larger than the path limit of 2, so it is also ignored.

**Figure 8: Path Length Validation**




## Key Usage

The key usage field in an EE or CA certificate defines the purpose of the key contained in the certificate.

- For EE certificates, if the key usage field is present but the certificate does not contain **digitalSignature** or **nonrepudiation** flags, the certificate is rejected. If the key usage field is not present, then key usage is not checked.
- For CA certificates, the key can be used for certificate or CRL signature validation. Because the PKI daemon is responsible for both X509 certificate validation and CRL downloads, key usage must be checked before validating the certificate or CRL.

  In certificate signature validation, the **keyCertSign** flag indicates that a CA certificate can be used for certificate signature validation. If this flag is not set, certificate validation is terminated.

  In Phase 1 negotiations of CRL signature validation, participants check the certificate revocation list (CRL) to see if certificates received during an IKE exchange are still valid. The CRL is periodically downloaded for CA profiles configured with CRL as the certificate revocation check. Downloaded CRL files must be verified before they are downloaded into the device. One of the verification steps is to validate the CRL signature using a CA certificate. The downloaded CRL is signed with the CA certificate's private key and it must be verified with the CA certificate's public key stored in the

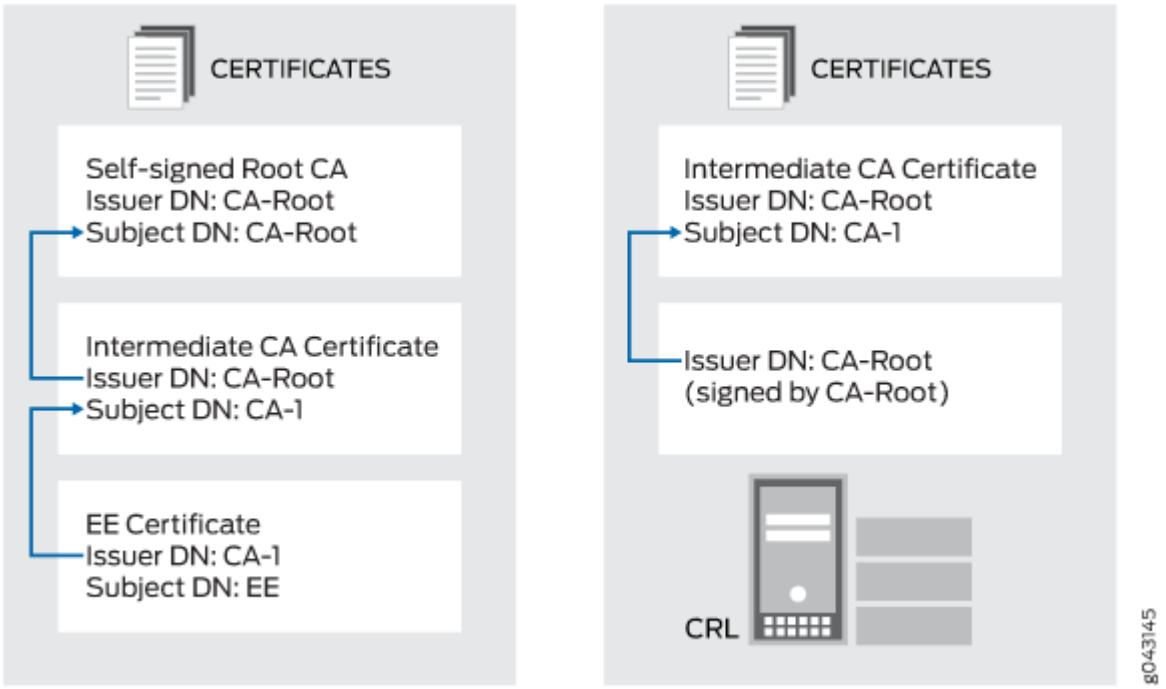
device. The key usage field in the CA certificate must contain the **CRLSign** flag to verify the downloaded CRL. If this flag is not present, the CRL is discarded.

## Issuer and Subject Distinguished Name Validation

Signature validation is performed for certificates received from a peer as well as for the CRL file downloaded from a CA server. Signature validation involves looking up the CA certificate in a CA database based on the issuer's distinguished name (DN) in the certificate or the CRL being verified.

Figure 9 on page 85 shows the lookup for CA certificates based on the issuer DN. In the EE certificate, the issuer DN is CA-1, which is the subject DN of the intermediate CA certificate in the chain. In the intermediate CA certificate, the issuer DN is CA-Root, which is the subject DN of the self-signed Root-CA certificate in the chain. In the CRL, the issuer DN is CA-Root, which is the subject DN of the self-signed Root-CA certificate.

**Figure 9: Issuer and Subject DN Validation**




The lookup for the issuer or subject DN must follow these rules for attribute values:

- Attribute values encoded in different ASN.1 types (for example, PrintableString and BMPString) are assumed to represent different strings.

- Attribute values encoded in PrintableString types are not case-sensitive. These attribute values are compared after removing leading and trailing white spaces and converting internal substrings of one or more consecutive white spaces to a single space.
- Attribute values encoded in types other than PrintableString are case-sensitive.

**SEE ALSO**

Example: Configuring a Certificate Authority Profile with CRL Locations | 75

Deleting Certificates (CLI Procedure) | 66

## Example: Validating Digital Certificate by Configuring Policy OIDs on an SRX Series Firewall

**IN THIS SECTION**

- Requirements | 86
- Overview | 87
- Configuration | 87
- Verification | 88

In some situations, it might be desirable to only accept certificates with known policy object identifiers (OIDs) from peers. This optional configuration allows certificate validation to succeed only if the certificate chain received from the peer contains at least one policy OID that is configured on the SRX Series Firewall. This example shows how to configure policy OIDs in the IKE policy on an SRX Series Firewall.

You must ensure that at least one of the policy OIDs configured on the SRX Series Firewall is included in a peer's certificate or certificate chain. Note that the **policy-oids** field in a peer's certificate is optional. If you configure policy OIDs in an IKE policy and the peer's certificate chain does not contain any policy OIDs, certificate validation for the peer fails.

### Requirements

Before you begin:

- Ensure that you are using Junos OS Release 12.3X48-D10 or later for SRX Series Firewalls.
- Configure an IPsec VPN tunnel. See "IPsec VPN with Autokey IKE Configuration Overview" on page 202. The complete IKE phase 1 and phase 2 VPN tunnel configuration is not shown in this example.

## Overview

This example shows an IKE policy configuration where policy OIDs 2.16.840.1.101.3.1.48.2 and 5.16.40.1.101.3.1.55.2 are specified. The IKE policy ike_cert_pol references the IKE proposal ike_cert_prop, which is not shown. The local certificate on the SRX Series Firewall is lc-igloo-root.

## Configuration

**IN THIS SECTION**


- Procedure | 87


### 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 ike policy ike_cert_pol mode main
set security ike policy ike_cert_pol proposals ike_cert_prop
set security ike policy ike_cert_pol certificate local-certificate lc-igloo-root
set security ike policy ike_cert_pol certificate policy-oids 2.16.840.1.101.3.1.48.2
set security ike policy ike_cert_pol certificate policy-oids 5.16.40.1.101.3.1.55.2
```

### 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 policy OIDs for certificate validation:

1. Configure the IKE policy:

```
[edit security ike policy ike_cert_pol]
user@host# set mode main
user@host# set proposals ike_cert_prop
user@host# set certificate local-certificate lc-igloo-root
user@host# set certificate policy-oids 2.16.840.1.101.3.1.48.2
user@host# set certificate policy-oids 5.16.40.1.101.3.1.55.2
```

### Results

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

```
user@host# show security ike policy ike_cert_pol
mode main;
proposals ike_cert_prop;
certificate {
    local-certificate lc-igloo-root;
    policy-oids [ 2.16.840.1.101.3.1.48.2 5.16.40.1.101.3.1.55.2 ];
}
```

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

## Verification

**IN THIS SECTION**


- Verifying the CA Certificate | 89
- Verifying Policy OID Validation | 90


Confirm that the configuration is working properly.

### Verifying the CA Certificate

#### Purpose

Display the CA certificate configured on the device.

#### Action

From operational mode, enter the `show security pki ca-certificate ca-profile ca-tmp` command.

```
user@host> show security pki ca-certificate ca-profile ca-tmp detail
                    Certificate identifier: ca-tmp
                           Certificate version: 3
                           Serial number: 00000047
                           Issuer:
                             Organization: U.S. Government,
                             Organizational unit: DoD, Organizational unit: Testing, Country: US,
                             Common name: Trust Anchor
                           Subject:
                             Organization: U.S. Government,
                             Organizational unit: Dod, Organizational unit: Testing, Country: US,
                             Common name: CA1-PP.01.03
                           Subject string:
                             C=US, O=U.S. Government, OU=Dod, OU=Testing, CN=CA1-PP.01.03
                           Validity:
                             Not before: 01- 1-1998 12:01 UTC
                             Not after: 01- 1-2048 12:01 UTC

?Public key algorithm: rsaEncryption(1024 bits)
  30:81:89:02:81:81:00:cb:fd:78:0c:be:87:ac:cd:c0:33:66:a3:18
  9e:fd:40:b7:9b:bc:dc:66:ff:08:45:f7:7e:fe:8e:d6:32:f8:5b:75
  db:76:f0:4d:21:9a:6e:4f:04:21:4c:7e:08:a1:f9:3d:ac:8b:90:76
  44:7b:c4:e9:9b:93:80:2a:64:83:6e:6a:cd:d8:d4:23:dd:ce:cb:3b
  b5:ea:da:2b:40:8d:ad:a9:4d:97:58:cf:60:af:82:94:30:47:b7:7d
  88:c3:76:c0:97:b4:6a:59:7e:f7:86:5d:d8:1f:af:fb:72:f1:b8:5c
  2a:35:1e:a7:9e:14:51:d4:19:ae:c7:5c:65:ea:f5:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Certificate Policy:
  Policy Identifier = 2.16.840.1.101.3.1.48.2
Use for key: CRL signing, Certificate signing
Fingerprint:
```

```
  e0:b3:2f:2e:a1:c5:ee:ad:af:dd:96:85:f6:78:24:c5:89:ed:39:40 (sha1)
  f3:47:6e:55:bc:9d:80:39:5a:40:70:8b:10:0e:93:c5 (md5)
```

### Verifying Policy OID Validation

#### Purpose

If the peer's certificate is successfully validated, IKE and IPsec security associations are established. If the validation of the peer's certificate fails, no IKE security association is established.

#### Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands.

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
821765168 UP   88875c981252c1d8  b744ac9c21bde57e  IKEv2          192.0.2.2
1106977837 UP  1a09e32d1e6f20f1  e008278091060acb  IKEv2          198.51.100.202
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <213909506 ESP:aes-cbc-192/sha256 8cb9e40a 1295/ unlim - root 500 192.0.2.2
  >213909506 ESP:aes-cbc-192/sha256 8271d2b2 1295/ unlim - root 500 192.0.2.2
  <218365954 ESP:aes-cbc-192/sha256 d0153bc0 1726/ unlim - root 1495 198.51.100.202
  >218365954 ESP:aes-cbc-192/sha256 97611813 1726/ unlim - root 1495 198.51.100.202
```

#### 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. In this case, check for the PKID_CERT_POLICY_CHECK_FAIL message in the system logs. This message indicates that the peer's certificate chain does not contain a policy OID that is configured on the SRX Series Firewall. Check the **policy-oids** values in the peer's certificate chain with the values configured on the SRX Series Firewall.

It might also be that the peer's certificate chain does not contain any **policy-oids** fields, which are optional fields. If this is the case, certificate validation fails if there are any policy OIDs configured on the SRX Series Firewall.

**SEE ALSO**

Example: Configuring a Certificate Authority Profile with CRL Locations | 75

Deleting Certificates (CLI Procedure) | 66

# Dynamic Updated of Trusted CA Certificates

**SUMMARY**

Read this topic to understand and configure dynamic update of default trusted CA certificates on your Junos OS devices.

**IN THIS SECTION**

- Understanding Dynamic Update of Trusted CA Certificates | 91
- Configuring Dynamic Update of Trusted CA Certificates | 93

## Understanding Dynamic Update of Trusted CA Certificates

**IN THIS SECTION**

- Tasks involved in dynamic update of trusted CA bundle | 92

The Junos OS device like an SRX Series Firewall provides a list of default trusted CA (Certificate Authority) certificates. These certificates are managed dynamically by the Junos OS device. You can also create a custom list of trusted CA certificates and load them into the device. But the custom trusted CA certificates needs to be managed manually. This section focuses on dynamic management of default trusted CA certificates.

With dynamic update of default trusted CA bundle -

- Removal of a CA in the event of compromise is taken care automatically.
- Addition of new CA to the default trusted CA bundle is immediate without having to wait for the new Junos OS release.

To load default trusted CA certificates, see request security pki ca-certificate ca-profile-group load with `filename default` option.

### Tasks involved in dynamic update of trusted CA bundle

Following tasks are performed as part of dynamic update of default trusted CA bundle -

- Juniper CDN server (http://signatures.juniper.net/cacert) hosts the default trusted CA certificates.
- The server hosts signed copy of target file and manifest file along with the EE certificate to verify the signed copy of these files. The target file contains a list of default trusted CA certificates (`default-trusted-ca-certs`). The manifest file includes the revision number and date of modification of the default trusted CA bundle.
- Automatic download of trusted CA bundle is enabled by default in Junos OS device. You can either use default or non-default routing instance to connect to the Internet in order to download and update the default trusted CA certificates.
- Public Key Management (PKI) process using PKID securely downloads the default trusted CA bundle (`default-trusted-ca-certs`) from the CDN server into the device.

  **NOTE**: Dynamic update of trusted CA certificates does not manage any changes to the previously loaded `ca-profile-group`, manually added CA certificates and certificates that are part of other trusted groups.

  See "Configuring Dynamic Updated of Trusted CA Certificates" on page 93.

- Once the `ca-profile-group load` command is issued, PKI process loads the default trusted CA certificates in the background, unblocking the CLI, allowing you to proceed with other tasks.
- If there is no `ca-profile-group` associated with `default-trusted-ca-certs`, with each periodic polling, PKI still downloads the latest copy of trusted CA bundle to the device.
- If a CA certificate is deleted from the default trusted CA list, the PKI process ensures all references to the CA certificate are removed. If any references are present in the `trusted-ca-group`, it only holds the references to `ca-profile` names with actual CA certificates already deleted. See "Configuring Dynamic Updated of Trusted CA Certificates" on page 93.

- PKI process periodically, by default every 24 hours, polls the CDN server for the latest default trusted CA bundle and updates the list for any changes to the trusted CAs in the bundle. If there are any changes, PKI process loads them in the background. You can optionally change the polling duration and also disable this auto-update process. See "Configuring Dynamic Updated of Trusted CA Certificates" on page 93.

## Configuring Dynamic Update of Trusted CA Certificates

**IN THIS SECTION**


- Checking connectivity to the CDN server | **94**
- Enabling automatic download of default trusted CA certificates | **94**
- Providing custom configuration for automatic download of default trusted CA certificates | **95**
- Downloading default trusted CA certificates explicitly | **96**
- Checking the download status of default trusted CA certificates | **97**
- Deactivating automatic download of trusted CA certificates | **98**


### Preprequisites

Before configuring dynamic update of default trusted CA certificates, ensure to meet the following prerequisites -

- Basic configuration of Junos OS device is completed.
- Your Junos OS device is reachable to Juniper CDN server. You can use non-default routing instance as well to connect to Internet to download the default trusted CA certificates. Ensure non-default routing instance is configured prior to configuring dynamic update of trusted CA certificates. Contact Juniper sales for Juniper CDN server details.
- For custom CDN server, ensure to have latest CA certificates and the URL. Configuration of custom CDN server is out of scope of this topic.

Based on your requirements, navigate to the following tasks to configure dynamic update of default trusted CA bundle.

## Checking connectivity to the CDN server

IN THIS SECTION


- Overview | 94
- Configuration | 94


### Overview

Use the following CLI to check connectivity to the CDN server for downloading default trusted CA certificates. This command downloads the manifest file and displays the trusted-ca-bundle version available in the CDN server.

See request security pki ca-certificate ca-profile-group default-trusted-ca-certs, for details about the command.

### Configuration

1. To check connectivity to the CDN server from operational mode of the Junos OS device -

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download check-server
```

## Enabling automatic download of default trusted CA certificates

IN THIS SECTION


- Overview | 94
- Configuration | 95


### Overview

Juniper Networks regularly updates the default trusted CA certificates on Juniper CDN server and makes it available for download on Junos OS device. Automatic download of default trusted CA certificates is enabled by default on Junos OS device. You can customize the configuration and load the

latest default trusted CA certificates at specified intervals. The default periodicity is 24 hours when you don't specify a value. When you use the default Juniper CDN Server (http://signatures.juniper.net/cacert), no separate configuration is needed.

This example shows how to enable automatic download of default trusted CA certificates on Junos OS device using default configuration settings. See default-trusted-ca-certs (Security) for details about the configuration statement. Loading of the downloaded default trusted CA certificates automatically happens in the background using the statement request security pki ca-certificate ca-profile-group load command. You don't have to explicitly run this command to load the certificates.

#### Configuration

As automatic download of default trusted CA certificates is enabled by default, no separate configuration is needed.

## Providing custom configuration for automatic download of default trusted CA certificates

**IN THIS SECTION**


- Overview | 95
- Configuration | 95


#### Overview

In this example, you provide following custom configuration while enabling the automatic download of custom CA certificates -

- Configure the Junos OS device to download and install the default trusted CA certificates every 48 hours.
- Specify the custom CDN server reachable via the URL signatures.example.net.
- Specify non-default routing instance to reach the CDN server.

See default-trusted-ca-certs (Security) for details about the configuration statement.

#### Configuration

Configuration

1. Set the periodicity of download and load operations to 48 hours. This CLI automatically loads the certificates into the Junos OS device.

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download interval hours 48
```

2. Specify the custom URL.

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download url
signatures.example.net
```

3. Specify the routing instance.

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download routing-instance RI1
```

4. Commit the configuration.

```
[edit]
user@host# commit
```

## Downloading default trusted CA certificates explicitly

**IN THIS SECTION**


- Overview | 96
- Configuration | 97


### Overview

Use the following CLI to manually download default trusted CA certificates to the Junos OS device from the CDN server. This command is in addition to automatic download of default trusted CA certs at regular intervals.

See request security pki ca-certificate ca-profile-group default-trusted-ca-certs for details about the command.

### Configuration

Configuration

1. To explicitly download default trusted CA certificates from operational mode of the Junos OS device -

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download
```

## Checking the download status of default trusted CA certificates

**IN THIS SECTION**


- Overview | 97
- Configuration | 97


### Overview

Use the following CLI to check the download status of default trusted CA certificates on the Junos OS device from the CDN server. This command displays the version number and version date. You can use this command to check the previous downloaded version and date.

See request security pki ca-certificate ca-profile-group default-trusted-ca-certs for details about the command.

### Configuration

Configuration

1. To check the version number and version date available on the Junos OS device -

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download status
```

## Deactivating automatic download of trusted CA certificates

IN THIS SECTION


- Overview | 98
- Configuration | 98


### Overview

Automatic download is enabled by default. This example shows how to deactivate the automatic download of default trusted CA certificates, though we don't recommend.

See default-trusted-ca-certs (Security) for details about the configuration statement.

### Configuration

Configuration

1. To deactivate automatic download of default trusted CA certificates -

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download deactivate
```

2. Commit the configuration.

```
[edit]
user@host# commit
```

### RELATED DOCUMENTATION

- request security pki ca-certificate ca-profile-group load
- default-trusted-ca-certs (Security)
- request security pki ca-certificate ca-profile-group default-trusted-ca-certs

# ACME Protocol

IN THIS SECTION


- Understanding ACME Protocol | 99
- Enroll Local Certificate Using Let's Encrypt Server | 100
- Manual Re-Enroll Local Certificate | 102
- Delete ACME Account | 102


## Understanding ACME Protocol

IN THIS SECTION

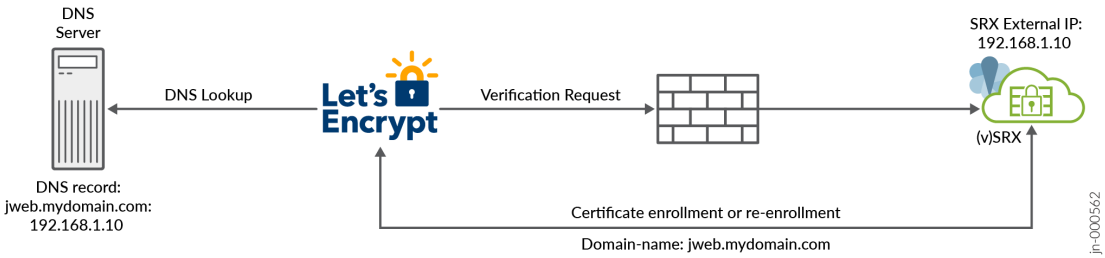

- Limitations | 100


Automated Certificate Management Environment (ACME) protocol is a new PKI enrollment standard used by several PKI servers such as Let's Encrypt. The Let's encrypt certificate allows for free usage of Web server certificates in SRX Series Firewalls, and this can be used in Juniper Secure Connect and J-Web. The Junos OS automatically re-enroll Let's Encrypt certificates on occurance of every 25 days.

The ACME protocol allows the enrollment of certificates from Let's Encrypt server or ACME enabled servers. The SRX Series Firewalls enrolls the certificates from Let's Encrypt server and Juniper Secure Connect validates the certificates without copying and downloading any CA certificates.

When using Let's Encrypt, ensure that the Let's Encrypt server is able to resolve the domain name to the IP address of the SRX Series Firewall interface as shown in Figure 10 on page 100. It must be able to reach the SRX Series Firewall interface on TCP port 80. During the certificate enrollment, the SRX Series Firewall will temporarily allow this incoming request automatically. If your SRX Series Firewall or an intermediate firewall or a router is blocking the TCP port 80, certificate enrollment will fail.

**Figure 10: Name Resolution for Let's Encrypt**




### Limitations

- ACME specification - The dns-01 and external account binding are not supported.
- ACME cannot be used when J-Web listen to port 80
- Wildcard certificate is not supported such as `*.mydomain.com`, instead you can enroll multiple dns names.

## Enroll Local Certificate Using Let's Encrypt Server

This example shows how to enroll the local certificate using Let's Encrypt.

1. Specify the CA profile.

```
[edit]

user@host#
set security pki ca-profile ISRG_Root_X1 ca-identity ISRG_Root_X1

user@host# set security pki ca-profile ISRG_Root_X1 revocation-check disable
user@host# set security pki ca-profile Lets_Encrypt ca-identity Lets_Encrypt

user@host#
set security pki ca-profile Lets_Encrypt enrollment url https://acme-v02.api.letsencrypt.org/
directory
```

2. Commit the configuration.

```
[edit]
user@host# commit
```

3. Load the CA certificate.

```
[edit]
user@host> request security pki ca-certificate load ca-profile ISRG_Root_X1 filename
ISRG_Root_X1.pem
```

4. Create ACME key ID.

```
[edit]
user@host> request security pki generate-key-pair size 2048 type rsa acme-key-id mydomain
```

5. Preparing enrollment of local certificate.

```
[edit]
user@host> request security pki generate-key-pair size 2048 type rsa certificate-id service-
mydomain
```

6. Enroll a certificate with one domain name.

```
[edit]
user@host> request security pki local-certificate enroll acme acme-key-id mydoamin
certificate-id service-mydomain ca-profile Lets_Encrypt domain-name jweb.mydomain.com email
jweb@acmejnpr.net letsencrypt-enrollment yes terms-of-service agree
```

Enroll a certificate with multiple domain names.

```
[edit]
user@host> request security pki local-certificate enroll acme acme-key-id mydomain
certificate-id service-mydomain ca-profile Lets_Encrypt domain-name jweb.mydomain.com,remote-
acess.mydomain.com  email jweb@acmejnpr.net letsencrypt-enrollment yes terms-of-service agree
```

7. Once the enrollment is finished the issued certificate will be loaded in certificate-id service-mydomian.

## Manual Re-Enroll Local Certificate

To re-enroll a local certificate online:

1. Initiate the re-enrollment request.

```
[edit]
user@host> request security pki local-certificate re-enroll acme acme-key-id mydomain
certificate-id serice-mydomain ca-profile Lets_Encrypt re-generate-keypair
```

2. Once the re-enrollment is finished the issued certificate will be loaded in certificate-id service-mydomian.

## Delete ACME Account

To delete the ACME account:

1. Delete the ACME account.

```
[edit]
user@host> clear security pki acme account acme-key-id mydomain ca-profile Lets_Encrypt
```

You can delete the ACME account key only if the ACME is activated or created by the enrollment.

# Configure Multiple Certificate Types to Establish IKE and IPsec SA

IN THIS SECTION


- Requirements | 103
- Overview | 104
- Topology | 104
- Configuration | 105
- Verification | 115


This example shows how to configure multiple certificate types to establish IKE and IPsec SA.

Starting in Junos OS Release 22.4R1, you can establish tunnels irrespective of the certificate type used on the initiator and responder if authentication-method is configured as `certificates` in IKE proposal using the `set security ike proposal` *`ike_proposal_name`* `authentication-method certificates` command.

You can view the certificate enrolled using `show security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

You can verify the enrolled certificate using the `request security pki local-certificate verify certificate-id` *`certificate-name`* command.

## Requirements

Before you begin:

- Ensure that you have certificates enrolled on your devices, see Certificate Enrollment.

  You can verify the certificates enrolled on your devices using the `request security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

- Ensure that you have IKE package installed, to verify the installed IKE package use the `show version | match ike` operational command.

If you don't have the IKE package installed on the device, you can install the IKE package using the operational command `request system software add optional://junos-ike.tgz`, for more information, see Enabling IPsec VPN Feature Set.

## Overview

This example configures multiple certificate types to establish IKE and IPsec SA between on SRX_A and on SRX_B.

**NOTE**: In this example, we have enrolled the RSA certificate on SRX_A and the ECDSA certificate on SRX_B devices. For more information about how to install the certificates, see Certificate Enrollment.

**Table 4: Topology Setup for SRX_A and SRX_B Devices**

| Device Name | Interface Used | IKE Gateway Address | IKE Gateway Local IP Address |
|---|---|---|---|
| SRX_A | ge-0/0/0 | 192.168.1.2 | 192.168.1.1 |
| SRX_B | ge-0/0/0 | 192.168.1.1 | 192.168.1.2 |

## Topology

The Figure 11 on page 104 describes topology for multiple certificate types support configuration.

**Figure 11: Multiple Certificate Types Support Configuration Example**

# Configuration

IN THIS SECTION


- Configuring SRX_A | 105
- Configuring SRX_B | 110


## 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-0/0/0 unit 0 family inet address 192.168.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
set interfaces st0 unit 1 family inet
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-0/0/1
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0
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
set security ike proposal IKE_PROP authentication-method certificates
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha-256
set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
```

```
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate r0_rsa_crt
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 192.168.1.2
set security ike gateway IKE_GW external-interface ge-0/0/0
set security ike gateway IKE_GW local-address 192.168.1.1
set security ike gateway IKE_GW version v2-only
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-192-cbc
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 establish-tunnels on-traffic
```

### 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 Configuration Mode Overvie in the CLI User Guide.

To configure multiple certificate types to establish IKE and IPsec SA:

1. View the certificates enrolled on your devices using the `show security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

   Install the certificate on your device if your device does not have the certificates enrolled. For more information, see Certificate Enrollment.

2. Configure interfaces.

```
user@srxa# set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
user@srxa# set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
user@srxa# set interfaces st0 unit 1 family inet
```

3. Configure security zones and the security policy.

```
user@srxa# set security zones security-zone trust host-inbound-traffic  system-services all
user@srxa# set security zones security-zone trust host-inbound-traffic  protocols all
user@srxa# set security zones security-zone trust interfaces ge-0/0/1
user@srxa# set security zones security-zone untrust host-inbound-traffic  system-services ike
```

```
user@srxa# set security zones security-zone untrust interfaces ge-0/0/0
user@srxa# set security zones security-zone VPN interfaces st0.1
user@srxa# set security policies from-zone VPN to-zone trust policy 1 match source-address
any
user@srxa# set security policies from-zone VPN to-zone trust policy 1 match destination-
address any
user@srxa# set security policies from-zone VPN to-zone trust policy 1 match application any
user@srxa# set security policies from-zone VPN to-zone trust policy 1 then permit
user@srxa# set security policies from-zone trust to-zone VPN policy 1 match source-address
any
user@srxa# set security policies from-zone trust to-zone VPN policy 1 match destination-
address any
user@srxa# set security policies from-zone trust to-zone VPN policy 1 match application any
user@srxa# set security policies from-zone trust to-zone VPN policy 1 then permit
user@srxa# set security policies default-policy deny-all
```

4. Configure the IKE proposal.

```
[edit]
user@srxa# set security ike proposal IKE_PROP authentication-method certificates
user@srxa# set security ike proposal IKE_PROP dh-group group5
user@srxa# set security ike proposal IKE_PROP authentication-algorithm sha-256
user@srxa# set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
```

5. Configure the IKE policy.

```
[edit]
user@srxa# set security ike policy IKE_POL proposals IKE_PROP
user@srxa# set security ike policy IKE_POL certificate local-certificate r0_rsa_crt
```

6. Configure the IKE gateway.

```
[edit]
user@srxa# set security ike gateway IKE_GW ike-policy IKE_POL
user@srxa# set security ike gateway IKE_GW address 192.168.1.2
user@srxa# set security ike gateway IKE_GW external-interface ge-0/0/0
user@srxa# set security ike gateway IKE_GW local-address 192.168.1.1
user@srxa# set security ike gateway IKE_GW version v2-only
```

7. Configure the IPsec proposal.

```
[edit]
user@srxa# set security ipsec proposal IPSEC_PROP protocol esp
user@srxa# set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@srxa# set security ipsec proposal IPSEC_PROP encryption-algorithm aes-192-cbc
```

8. Configure the IPsec policy.

```
[edit]
user@srxa# set security ipsec policy IPSEC_POL proposals IPSEC_PROP
```

9. Configure the IPsec VPN.

```
[edit]
user@srxa# set security ipsec vpn IPSEC_VPN bind-interface st0.1
user@srxa# set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
user@srxa# set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
user@srxa# set security ipsec vpn IPSEC_VPN establish-tunnels on-traffic
```

### Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `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@srxa# show interfaces
ge-0/0/0 {
    description untrust;
    unit 0 {
        family inet {
            address 192.168.1.1/24;
            }
        }
    }
ge-0/0/1 {
    description trust;
```

```
    unit 0 {
        family inet {
            address 172.16.1.1/24;
            }
        }
    }
st0 {
    unit 1 {
        family inet;
    }
}

[edit]
user@srxa# show security ike
proposal IKE_PROP {
    authentication-method certificates;
    dh-group group5;
    authentication-algorithm sha-256;
    encryption-algorithm aes-128-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate r0_crt_rsa;
    }
}
gateway IKE_GW {
    ike-policy IKE_POL;
    address 192.168.1.2;
    external-interface ge-0/0/0;
    local-address 192.168.1.1;
    version v2-only;
}

[edit]
user@srxa# show security ipsec
    proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-192-cbc;
}
policy IPSEC_POL {
    proposals IPSEC_PROP;
```

```
}
vpn IPSEC_VPN {
    bind-interface st0.1;
    ike {
        gateway IKE_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels on-traffic;
}
```

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-0/0/0 unit 0 family inet address 192.168.1.2/24
set interfaces ge-0/0/1 unit 0 family inet address 172.18.1.2/24
set interfaces st0 unit 1 family inet
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-0/0/1
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0
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
set security ike proposal IKE_PROP authentication-method certificates
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha-256
```

```
set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate r1_crt_ecdsa384
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 192.168.1.1
set security ike gateway IKE_GW external-interface ge-0/0/0
set security ike gateway IKE_GW local-address 192.168.1.2
set security ike gateway IKE_GW version v2-only
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-192-cbc
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 establish-tunnels on-traffic
```

### 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 Configuration Mode Overview in the CLI User Guide.

To configure multiple certificate types to establish IKE and IPsec SA:

1. View the certificates enrolled on your devices using the `request security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

   Install the certificate on your device if your device does not have the certificates enrolled. For more information, see Certificate Enrollment.

2. Configure interfaces.

```
user@srxb# set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.2/24
user@srxb# set interfaces ge-0/0/1 unit 0 family inet address 172.18.1.2/24
user@srxb# set interfaces st0 unit 1 family inet
```

3. Configure security zones and the security policy.

```
user@srxb# set security zones security-zone trust host-inbound-traffic  system-services all
user@srxb# set security zones security-zone trust host-inbound-traffic  protocols all
user@srxb# set security zones security-zone trust interfaces ge-0/0/1
```

```
user@srxb# set security zones security-zone untrust host-inbound-traffic  system-services ike
user@srxb# set security zones security-zone untrust interfaces ge-0/0/0
user@srxb# set security zones security-zone VPN interfaces st0.1
user@srxb# set security policies from-zone VPN to-zone trust policy 1 match source-address
any
user@srxb# set security policies from-zone VPN to-zone trust policy 1 match destination-
address any
user@srxb# set security policies from-zone VPN to-zone trust policy 1 match application any
user@srxb# set security policies from-zone VPN to-zone trust policy 1 then permit
user@srxb# set security policies from-zone trust to-zone VPN policy 1 match source-address
any
user@srxb# set security policies from-zone trust to-zone VPN policy 1 match destination-
address any
user@srxb# set security policies from-zone trust to-zone VPN policy 1 match application any
user@srxb# set security policies from-zone trust to-zone VPN policy 1 then permit
user@srxb# set security policies default-policy deny-all
```

4. Configure the IKE proposal.

```
[edit]
user@srxb# set security ike proposal IKE_PROP authentication-method certificates
user@srxb# set security ike proposal IKE_PROP dh-group group5
user@srxb# set security ike proposal IKE_PROP authentication-algorithm sha-256
user@srxb# set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
```

5. Configure the IKE policy.

```
[edit]
user@srxb# set security ike policy IKE_POL proposals IKE_PROP
user@srxb# set security ike policy IKE_POL certificate local-certificate r1_crt_ecdsa384
```

6. Configure the IKE gateway.

```
[edit]
user@srxb# set security ike gateway IKE_GW ike-policy IKE_POL
user@srxb# set security ike gateway IKE_GW address 192.168.1.1
user@srxb# set security ike gateway IKE_GW external-interface ge-0/0/0
user@srxb# set security ike gateway IKE_GW local-address 192.168.1.2
user@srxb# set security ike gateway IKE_GW version v2-only
```

7. Configure the IPsec proposal.

```
[edit]
user@srxb# set security ipsec proposal IPSEC_PROP protocol esp
user@srxb# set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@srxb# set security ipsec proposal IPSEC_PROP encryption-algorithm aes-192-cbc
```

8. Configure the IPsec policy.

```
[edit]
user@srxb# set security ipsec policy IPSEC_POL proposals IPSEC_PROP
```

9. Configure the IPsec VPN.

```
[edit]
user@srxb# set security ipsec vpn IPSEC_VPN bind-interface st0.1
user@srxb# set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
user@srxb# set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
user@srxb# set security ipsec vpn IPSEC_VPN establish-tunnels immediately
```

### Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `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@srxb# show interfaces
ge-0/0/0 {
    description untrust;
    unit 0 {
        family inet {
            address 192.168.1.2/24;
            }
        }
    }
ge-0/0/1 {
    description trust;
```

```
    unit 0 {
        family inet {
            address 172.18.1.2/24;
            }
        }
    }
st0 {
    unit 1 {
        family inet;
    }
}

[edit]
user@srxb# show security ike
proposal IKE_PROP {
    authentication-method certificates;
    dh-group group5;
    authentication-algorithm sha-256;
    encryption-algorithm aes-128-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate r1_crt_ecdsa384;
    }
}
gateway IKE_GW {
    ike-policy IKE_POL;
    address 192.168.1.1;
    external-interface ge-0/0/0;
    local-address 192.168.1.2;
    version v2-only;
}

[edit]
user@srxb# show security ipsec
    proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-192-cbc;
}
policy IPSEC_POL {
    proposals IPSEC_PROP;
```

```
}
vpn IPSEC_VPN {
    bind-interface st0.1;
    ike {
        gateway IKE_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
```

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

## Verification

**IN THIS SECTION**


- Verify SRX_A | **115**
- Verify SRX_B | **121**


Confirm that the configuration is working properly.

### Verify SRX_A

The sample outputs shown are on SRX-A.

#### Purpose

Verify the IPsec Phase 2 status.

### Action

From operational mode, enter the `show security ike security-associations` command.

```
user@srxa> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
32      UP     6723643250f0f357  f6295f11b0d7c8ab  IKEv2          192.168.1.2
```

From operational mode, enter the `show security ipsec security-associations` command.

```
user@srxa> show security ipsec security-associations
  Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <500033 ESP:aes-cbc-192/sha256 0x5f156c1b 2750/ unlim - root 500 192.168.1.2
  >500033 ESP:aes-cbc-192/sha256 0x7ea065e7 2750/ unlim - root 500 192.168.1.2
```

From operational mode, enter the `show security ike security-associations detail` command.

```
user@srxa> show security ike security-associations detail
  IKE peer 192.168.1.2, Index 32, Gateway Name: IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 6723643250f0f357, Responder cookie: f6295f11b0d7c8ab
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/0.0
  Routing instance: default
  Local: 192.168.1.1:500, Remote: 192.168.1.2:500
  Lifetime: Expires in 28165 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: 192.168.1.2
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                1346
   Output bytes  :                1887
```

```
   Input  packets:                    3
   Output packets:                    4
   Input  fragmented packets:       2
   Output fragmented packets:       3
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500033

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.1.1:500, Remote: 192.168.1.2:500
    Local identity: 192.168.1.1
    Remote identity: 192.168.1.2
    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
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@srxa> show security ipsec security-associations detail
  ID: 500033 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(0.0.0.0-255.255.255.255)
  TS Type: proxy-id
  Version: IKEv2
  PFS group: N/A
```

```
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Tunnel MTU: 0, 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 09 2023 22:41:36: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x5f156c1b, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2895 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2286 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 32
  Direction: outbound, SPI: 0x7ea065e7, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2895 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2286 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 32
```

From operational mode, enter the `show security pki local-certificate certificate-id r0_rsa_cr detail` command.

```
user@srxa> show security pki local-certificate certificate-id r0_rsa_crt detail
  LSYS: root-logical-system
Certificate identifier: r0_rsa_crt
  Certificate version: 3

  Serial number:
    hexadecimal: 0x0186a62478ae8f0cdd766eb38dbd53
```

```
    decimal: 792330290775730184700710622630638 7
  Issuer:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject:
    Organization: juniper, Organizational unit: marketing, State: california, Locality:
sunnyvale, Common name: r0, Domain component: juniper
  Subject string:
    DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
  Alternate subject: "r0@juniper.net", r0.juniper.net, 192.168.1.1
  Cert-Chain: Root-CA
  Validity:
    Not before: 03- 3-2023 05:54 UTC
    Not after: 06- 6-2027 12:36 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:b0:e5:53:8d:7e:20:fa:6b:21:c2:d1
    2b:48:8f:af:c3:eb:8b:23:4a:f7:c5:1f:cf:2c:6a:b3:2e:8a:ef:1b
    f7:97:aa:fd:1d:ab:1c:76:9b:40:a3:ac:bb:49:f6:93:f9:e1:4e:62
    df:3d:ca:e5:d2:95:9c:a0:f4:2b:d7:7e:1d:20:94:69:a8:e4:cf:dc
    15:90:4c:be:1d:d8:1c:52:08:3a:d1:05:a3:bb:2f:8f:31:0c:6b:21
    ef:76:c3:c7:fb:be:4a:cb:da:cc:8d:04:3a:75:0c:eb:5d:e2:f6:13
    50:fe:39:67:c0:77:2f:32:b0:5e:38:6f:9c:79:b3:5d:f3:57:f4:f8
    42:f5:22:5b:6c:58:67:90:4e:1e:ec:6a:03:e2:c0:87:65:02:ca:da
    6f:95:0a:8c:2a:fd:45:4f:3a:b5:ef:18:05:1c:54:e6:fe:45:bb:73
    53:81:b2:c6:b7:36:36:57:6d:9c:d3:d9:80:e7:d6:85:92:74:32:88
    16:01:03:27:57:76:8e:5e:d6:73:ac:bf:68:fd:6d:a1:2a:8f:f5:3a
    29:b0:c9:44:9b:c8:46:c1:bf:c0:52:2a:f0:51:be:b5:f6:e1:f5:3e
    96:1d:3a:42:29:28:d3:cf:60:b9:eb:24:04:47:d3:f1:3f:5e:38:fc
    7f:33:f6:94:9d:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Fingerprint:
    4d:f6:89:c5:d6:3c:74:73:db:3e:f6:4b:1e:26:6c:c1:1c:1d:a7:4d (sha1)
    6b:1c:a8:1f:de:5a:9b:3e:d5:c4:85:29:af:3f:82:f2 (md5)

6b:7a:b5:d1:57:cf:75:9d:1f:63:b9:f6:49:e4:4e:b3:13:2c:83:f1:f7:25:44:6f:45:2f:0d:2f:ae:a8:80:85
(sha256)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

From operational mode, enter the `show security pki ca-certificate ca-profile Root-CA detail` command.

```
user@srxa> show security pki ca-certificate ca-profile Root-CA detail
  LSYS: root-logical-system
  CA profile: Root-CA
Certificate identifier: Root-CA
  Certificate version: 3

  Serial number:
    hexadecimal: 0x00000440
    decimal: 1088
  Issuer:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject string:
    C=India, O=juniper, CN=Root-CA
  Validity:
    Not before: 06- 7-2022 12:36 UTC
    Not after: 06- 6-2027 12:36 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:cd:9c:e6:9f:62:6c:49:15:c2:da:eb
    8e:e6:e5:a1:88:40:d8:b5:2e:5b:1a:0e:de:96:d7:0b:19:f9:03:44
    98:49:d5:cc:a8:90:2b:7f:1b:58:7b:1f:26:92:18:4c:2d:37:65:5c
    9f:0f:6e:10:b5:34:6f:2d:b5:9c:27:3b:a6:b1:b5:a0:e2:a6:92:3d
    e4:68:fe:5d:71:06:6f:ce:e6:0f:0f:e3:94:2a:23:57:98:a0:6a:9c
    e0:52:a2:47:ff:ce:b0:47:bd:36:95:80:a7:af:d2:49:b1:5d:2a:3d
    28:e4:95:06:b8:b3:d9:07:11:3c:13:af:c6:e2:51:08:22:82:2d:ec
    4f:26:40:b0:b0:55:2d:6e:c0:c8:19:34:a7:99:5a:bc:58:98:69:ae
    04:d6:6d:ec:4a:c9:55:a5:ff:00:cb:3b:02:85:fa:02:a1:5c:c1:9d
    6d:44:b8:95:8f:77:c0:53:fc:7f:a4:09:a3:25:1c:4a:e2:9d:0c:81
    08:b4:c8:b8:0d:bc:94:75:54:75:57:4f:d3:a4:17:0d:5d:1a:f3:c1
    1d:5d:73:2f:fe:8b:cb:fc:1f:93:87:72:d6:be:df:86:d7:e6:d1:c7
    0d:00:1a:6e:58:db:6a:1c:2f:1d:17:46:9a:f2:69:b4:21:db:08:5d
    8d:ab:30:7d:7f:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Distribution CRL:
     http://10.102.40.55:8080/crl-as-der/currentcrl-11.crl?id=11
  Use for key: CRL signing, Certificate signing, Key encipherment, Digital signature
  Fingerprint:
    8b:84:60:2a:58:5b:80:f0:b9:ae:25:9f:67:3d:d6:81:ee:43:6c:d4 (sha1)
    ab:ec:4d:fe:d4:04:9c:c9:79:1d:9a:33:4e:6d:78:f6 (md5)
```

```
9d:f0:c0:a0:93:74:11:53:d3:4d:2d:75:d3:60:37:5f:fb:b7:a9:67:42:cd:7c:3c:0e:0f:9b:58:36:3c:14:f5
(sha256)
```

## Verify SRX_B

The sample outputs shown are on SRX-B.

### Purpose

Verify the IPsec Phase 2 status.

### Action

From operational mode, enter the `show security ike security-associations` command.

```
user@srxb> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
56042   UP     6723643250f0f357  f6295f11b0d7c8ab  IKEv2          192.168.1.1
```

From operational mode, enter the `show security ipsec security-associations` command.

```
user@srxb> show security ipsec security-associations
Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <500230 ESP:aes-cbc-192/sha256 0x7ea065e7 2638/ unlim - root 500 192.168.1.1
  >500230 ESP:aes-cbc-192/sha256 0x5f156c1b 2638/ unlim - root 500 192.168.1.1
```

From operational mode, enter the `show security ike security-associations detail` command.

```
user@srxb> show security ike security-associations detail
  IKE peer 192.168.1.1, Index 56042, Gateway Name: IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 6723643250f0f357, Responder cookie: f6295f11b0d7c8ab
  Exchange type: IKEv2, Authentication method: ECDSA-384-signatures
  Local gateway interface: ge-0/0/0.0
  Routing instance: default
  Local: 192.168.1.2:500, Remote: 192.168.1.1:500
  Lifetime: Expires in 18995 seconds
```

```
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: 192.168.1.1
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                2934
   Output bytes  :                2379
   Input  packets:                  10
   Output packets:                   9
   Input  fragmented packets:       3
   Output fragmented packets:       2
  IPSec security associations: 8 created, 3 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500230

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.1.2:500, Remote: 192.168.1.1:500
    Local identity: 192.168.1.2
    Remote identity: 192.168.1.1
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                Responder stats:
    Request Out             : 1                     Request In             :
2
    Response In             : 1                     Response Out           :
2
    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
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@srxb> show security ipsec security-associations detail
  ID: 500230 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.1
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(0.0.0.0-255.255.255.255)
  TS Type: proxy-id
  Version: IKEv2
  PFS group: N/A
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Tunnel MTU: 0, 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 02 2023 22:26:16: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x7ea065e7, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2633 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2002 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 56042
  Direction: outbound, SPI: 0x5f156c1b, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2633 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2002 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
```

```
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 56042
```

From operational mode, enter the `show security pki local-certificate certificate-id r1_crt_ecdsa384 detail` command.

```
user@srxb> show security pki local-certificate certificate-id r1_crt_ecdsa384 detail
  LSYS: root-logical-system
Certificate identifier: r1_crt_ecdsa384
  Certificate version: 3

  Serial number:
    hexadecimal: 0x0186a6254347a38063946d08595a55
    decimal: 792330315268321674029666848151125
  Issuer:
    Organization: juniper, Country: India, Common name: root-ecdsa-384
  Subject:
    Organization: juniper, Organizational unit: marketing, State: california, Locality:
sunnyvale, Common name: r1_spk1, Domain component: juniper
  Subject string:
    DC=juniper, CN=r1_spk1, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
  Alternate subject: "r1_spk1@juniper.net", r1_spk1.juniper.net, 192.168.2
  Cert-Chain: root-ecdsa-384
  Validity:
    Not before: 03- 3-2023 05:55 UTC
    Not after: 06- 6-2027 13:21 UTC
  Public key algorithm: ecdsaEncryption(384 bits)
    04:c2:ba:19:dc:0d:62:a7:94:7b:9b:1d:4d:ff:a1:e1:44:b5:57:a7
    cb:7d:33:6b:35:87:b8:e4:ca:44:b1:6c:6d:63:ae:6f:3c:31:7c:7e
    65:99:b3:2d:a3:76:30:23:e5:0e:34:e1:28:54:d6:3e:d3:8b:de:b6
    b9:45:05:82:6f:1d:20:b7:6f:3c:ce:a2:13:a2:b4:37:0b:db:35:1e
    20:54:b5:06:9d:f8:7f:19:7b:c5:d7:7b:57:8b:28:31:d3
  Signature algorithm: ecdsa-with-SHA384
  Fingerprint:
    9b:cb:5a:57:a8:60:a0:ee:5c:be:59:4c:db:35:39:d3:b7:29:ef:b1 (sha1)
    ef:b5:e3:be:35:1b:6e:02:0b:61:11:a5:53:07:b4:89 (md5)

8f:86:d0:12:ea:bc:a8:81:a8:17:3a:f9:03:e4:91:57:20:9c:11:bc:a4:dd:d1:7f:d1:48:3f:5b:d9:fb:93:32
(sha256)
  Auto-re-enrollment:
```

```
    Status: Disabled
    Next trigger time: Timer not started
```

s

From operational mode, enter the `show security pki ca-certificate ca-profile Root-CA detail` command.

```
user@srxb> show security pki ca-certificate ca-profile Root-CA detail
  LSYS: root-logical-system
  CA profile: Root-CA
Certificate identifier: Root-CA
  Certificate version: 3

  Serial number:
    hexadecimal: 0x00000440
    decimal: 1088
  Issuer:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject string:
    C=India, O=juniper, CN=Root-CA
  Validity:
    Not before: 06- 7-2022 12:36 UTC
    Not after: 06- 6-2027 12:36 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:cd:9c:e6:9f:62:6c:49:15:c2:da:eb
    8e:e6:e5:a1:88:40:d8:b5:2e:5b:1a:0e:de:96:d7:0b:19:f9:03:44
    98:49:d5:cc:a8:90:2b:7f:1b:58:7b:1f:26:92:18:4c:2d:37:65:5c
    9f:0f:6e:10:b5:34:6f:2d:b5:9c:27:3b:a6:b1:b5:a0:e2:a6:92:3d
    e4:68:fe:5d:71:06:6f:ce:e6:0f:0f:e3:94:2a:23:57:98:a0:6a:9c
    e0:52:a2:47:ff:ce:b0:47:bd:36:95:80:a7:af:d2:49:b1:5d:2a:3d
    28:e4:95:06:b8:b3:d9:07:11:3c:13:af:c6:e2:51:08:22:82:2d:ec
    4f:26:40:b0:b0:55:2d:6e:c0:c8:19:34:a7:99:5a:bc:58:98:69:ae
    04:d6:6d:ec:4a:c9:55:a5:ff:00:cb:3b:02:85:fa:02:a1:5c:c1:9d
    6d:44:b8:95:8f:77:c0:53:fc:7f:a4:09:a3:25:1c:4a:e2:9d:0c:81
    08:b4:c8:b8:0d:bc:94:75:54:75:57:4f:d3:a4:17:0d:5d:1a:f3:c1
    1d:5d:73:2f:fe:8b:cb:fc:1f:93:87:72:d6:be:df:86:d7:e6:d1:c7
    0d:00:1a:6e:58:db:6a:1c:2f:1d:17:46:9a:f2:69:b4:21:db:08:5d
    8d:ab:30:7d:7f:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Distribution CRL:
```

```
     http://10.102.40.55:8080/crl-as-der/currentcrl-11.crl?id=11
  Use for key: CRL signing, Certificate signing, Key encipherment, Digital signature
  Fingerprint:
    8b:84:60:2a:58:5b:80:f0:b9:ae:25:9f:67:3d:d6:81:ee:43:6c:d4 (sha1)
    ab:ec:4d:fe:d4:04:9c:c9:79:1d:9a:33:4e:6d:78:f6 (md5)

9d:f0:c0:a0:93:74:11:53:d3:4d:2d:75:d3:60:37:5f:fb:b7:a9:67:42:cd:7c:3c:0e:0f:9b:58:36:3c:14:f5
(sha256)
```

4

CHAPTER

# IPsec VPN in Junos OS

Internet Key Exchange (IKE) for IPsec VPN | 128

IPsec VPN Overview | 176

# Internet Key Exchange (IKE) for IPsec VPN

IN THIS SECTION


- IKE and IPsec Packet Processing | **128**
- Introduction to IKE in Junos OS | **136**
- IKE Proposal | **141**
- IKE Policy | **142**
- Rekeying and Reauthentication | **142**
- IKE Authentication (Certificate-Based Authentication) | **144**
- IKE Protection from DDoS Attacks | **147**
- Configure Protection Against IKE DDoS Attacks | **150**
- Example: Configuring a Device for Peer Certificate Chain Validation | **157**
- IKEv2 Fragmentation | **171**
- IKE Policy with a Trusted CA | **172**
- Configuring Establish-Tunnel Responder-only in IKE | **174**


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.

## IKE and IPsec Packet Processing

IN THIS SECTION


- IKE Packet Processing | **129**
- IPsec Packet Processing | **132**

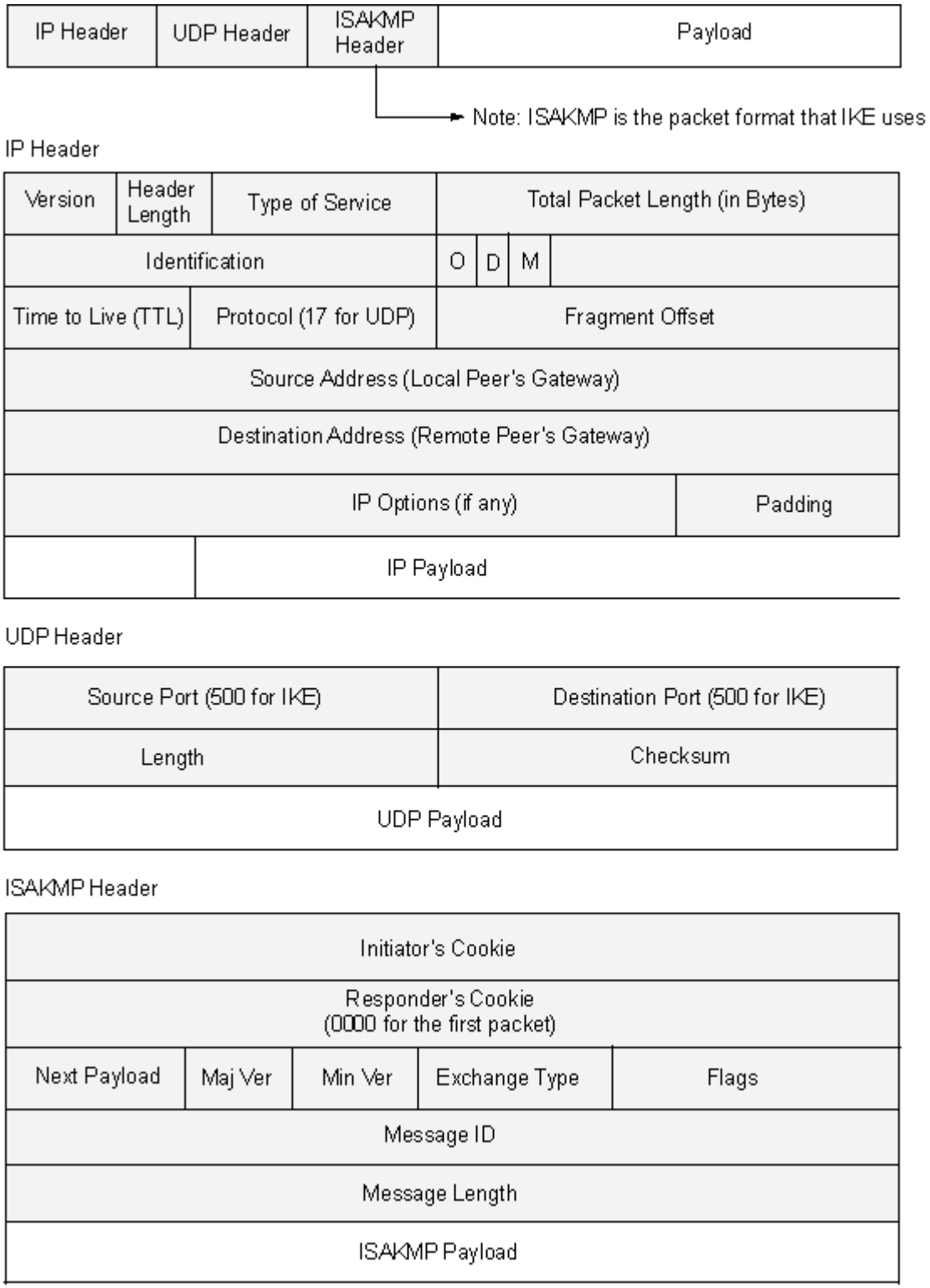
IKE provides tunnel management for IPsec and authenticates end entities. IKE performs a Diffie-Hellman (DH) key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the 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 12 on page 130.

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 12: IKE Packet for Phases 1 and 2**



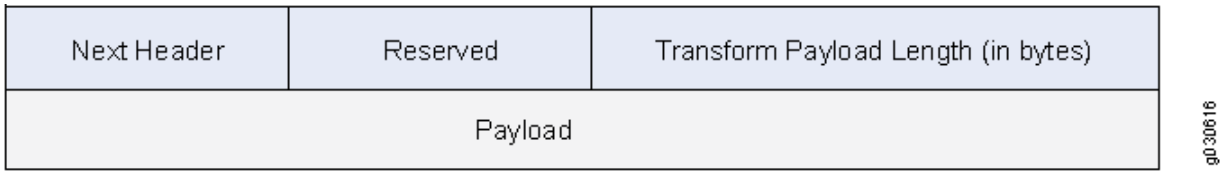
The Next Payload field contains a number indicating one of the following payload types:

- 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 an 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.

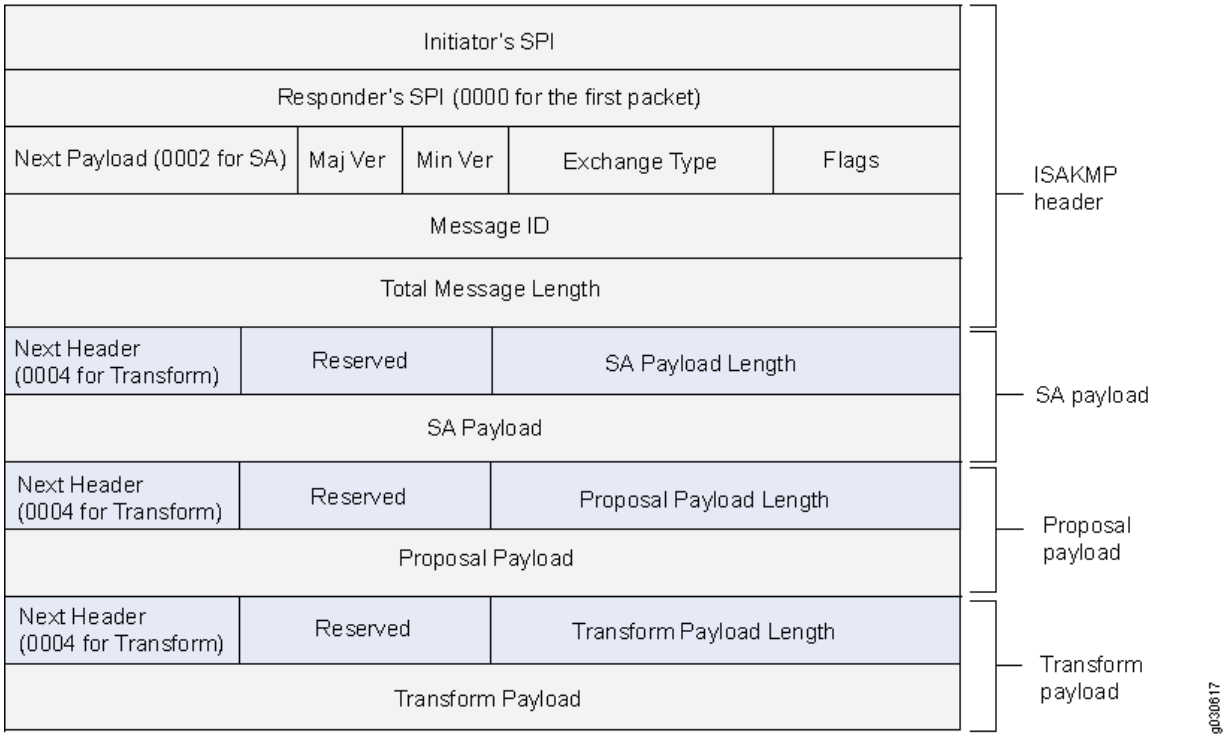
Each ISAKMP payload begins with the same generic header, as shown in Figure 13 on page 131.

**Figure 13: 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 14 on page 132 for an example.

**Figure 14: ISAKMP Header with Generic ISAKMP Payloads**

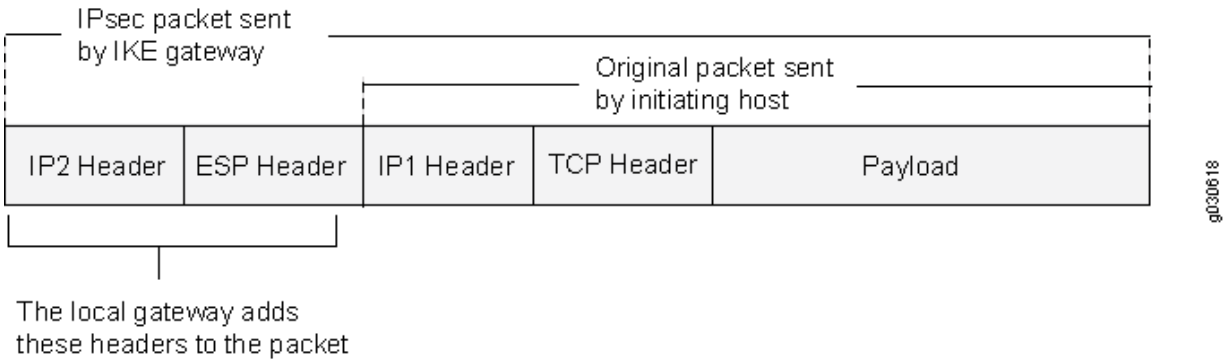



## IPsec Packet Processing

After IKE negotiations complete and the two IKE gateways have established Phase 1 and Phase 2 security associations (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 15 on page 133. The device adds two additional headers to the original packet that the initiating host sends.

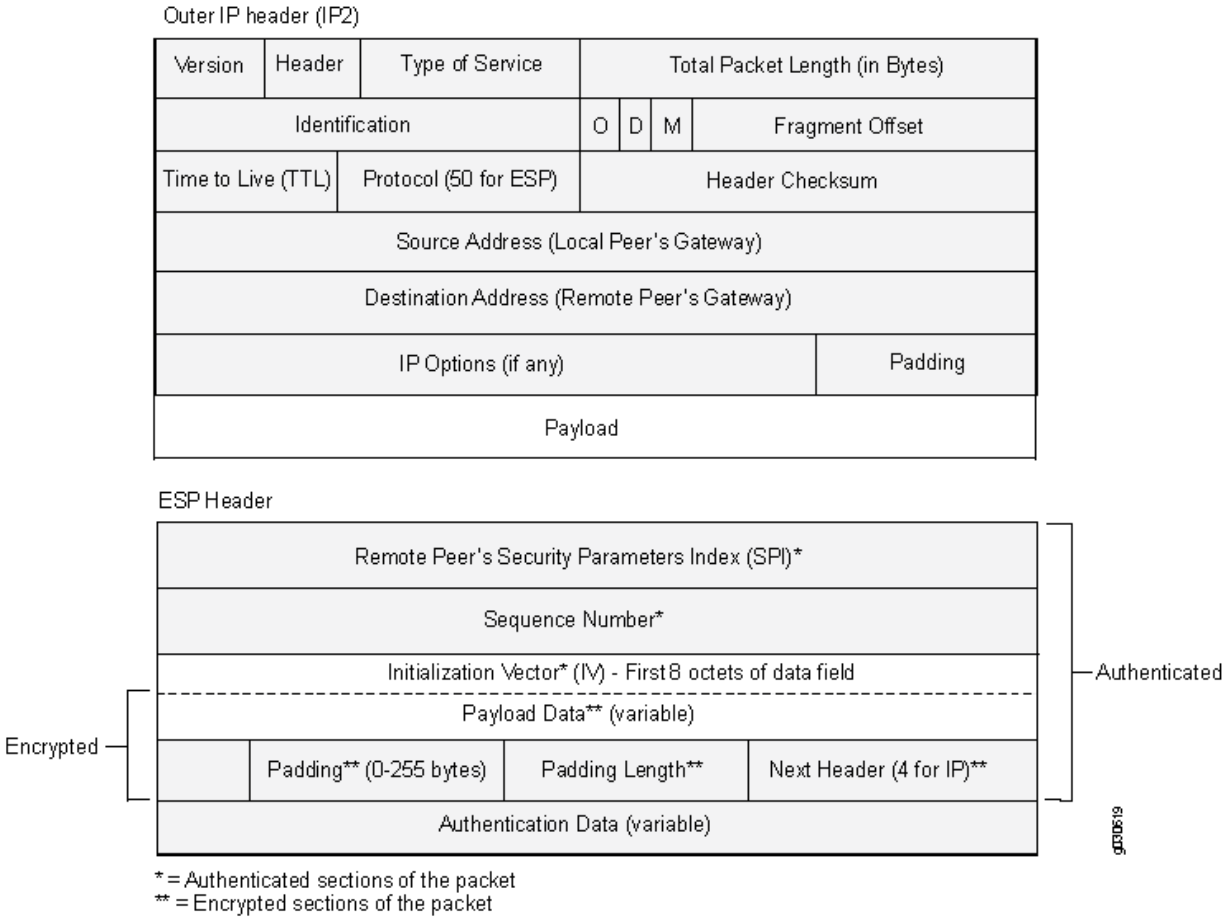
As shown in Figure 15 on page 133, the packet that the initiating host constructs includes the payload, the TCP header, and the inner IP header (IP1).

**Figure 15: 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. This is shown in Figure 16 on page 134.

**Figure 16: 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 17 on page 135.

**Figure 17: Inner IP Header (IP1) and TCP Header**

Inner IP Header (IP1)

| Version | Header | Type of Service | Total Packet Length (in Bytes) | | | |
|---|---|---|---|---|---|---|
| Identification | | | O | D | M | Fragment Offset |
| Time to Live (TTL) | Protocol (6 for TCP) | | Header Checksum | | | |
| Source Address (Installing Host) | | | | | | |
| Destination Address (Receiving Host) | | | | | | |
| IP Options (if any) | | | | | | Padding |
| Payload | | | | | | |

TCP Header

| Source Port | | | | | | | | Destination Port |
|---|---|---|---|---|---|---|---|---|
| Sequence Number | | | | | | | | |
| Acknowledgement Number | | | | | | | | |
| Header Length | Reserved | URG | ACK | PSH | RST | SYN | FIN | Window Size |
| Checksum | | | | | | | | Urgent Pointer |
| IP Options (if any) | | | | | | | | Padding |
| Data | | | | | | | | |

# Introduction to IKE in Junos OS

IN THIS SECTION


- Configuring IKEv2 in Junos OS | 137
- Understanding IKEv2 Configuration Payload | 137
- Understanding Pico Cell Provisioning | 140


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 5 on page 137 describes the IKEv2 configuration attributes supported on SRX Series Firewalls.

**Table 5: 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 5: 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.

Starting in Junos OS Release 20.3R1, on SRX5000 line with SPC3 and vSRX Virtual Firewall running iked process, we've improved IKEv2 configuration payload to:

- 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 SRX Series Firewalls 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.

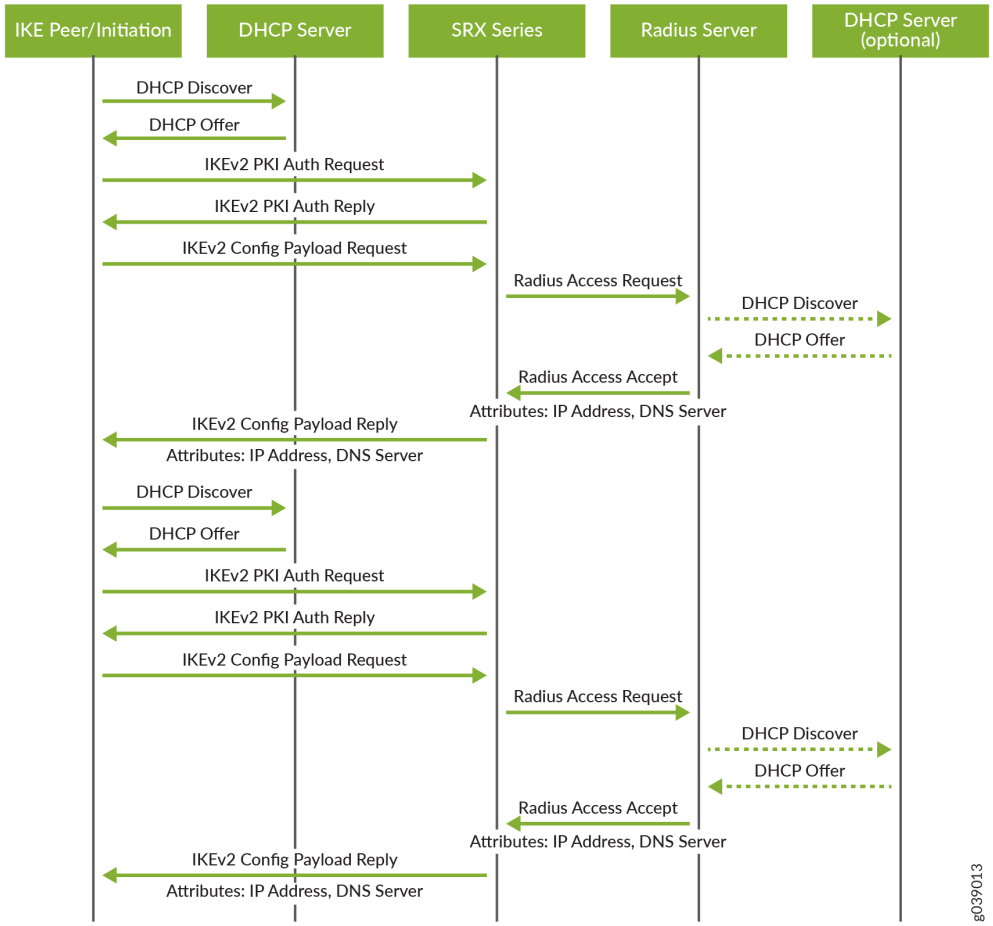
In a route-based VPN, secure tunnel (st0) interfaces operate in either point-to-multipoint or point-to-point mode. Address assignment through the IKEv2 configuration payload is now supported for point-to-multipoint or point-to-point mode. For point-to-multipoint interfaces, the interfaces must be numbered 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.

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 matches 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.

Both the SRX Series 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.

Figure 18 on page 140 shows a typical workflow for a IKEv2 Configuration Payload.

**Figure 18: 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.

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.

## Understanding Pico Cell Provisioning

IKEv2 configuration payload can be used to propagate provisioning information from an IKE responder, such as an SRX Series 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 SRX Series 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 SRX Series 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 SRX Series 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 SRX Series 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 | **690**

## 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 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 preshared key 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 | **142**
- Supported Features | **143**
- Limitations | **143**


### 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 for SRX5400, SRX5600, and SRX5800 devices
- In-service software upgrade (ISSU) on SRX5400, SRX5600, and SRX5800 devices
- 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 | **144**


### Multilevel Hierarchy for Certificate Authentication

Certificate-based authentication is an authentication method supported on SRX Series 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.

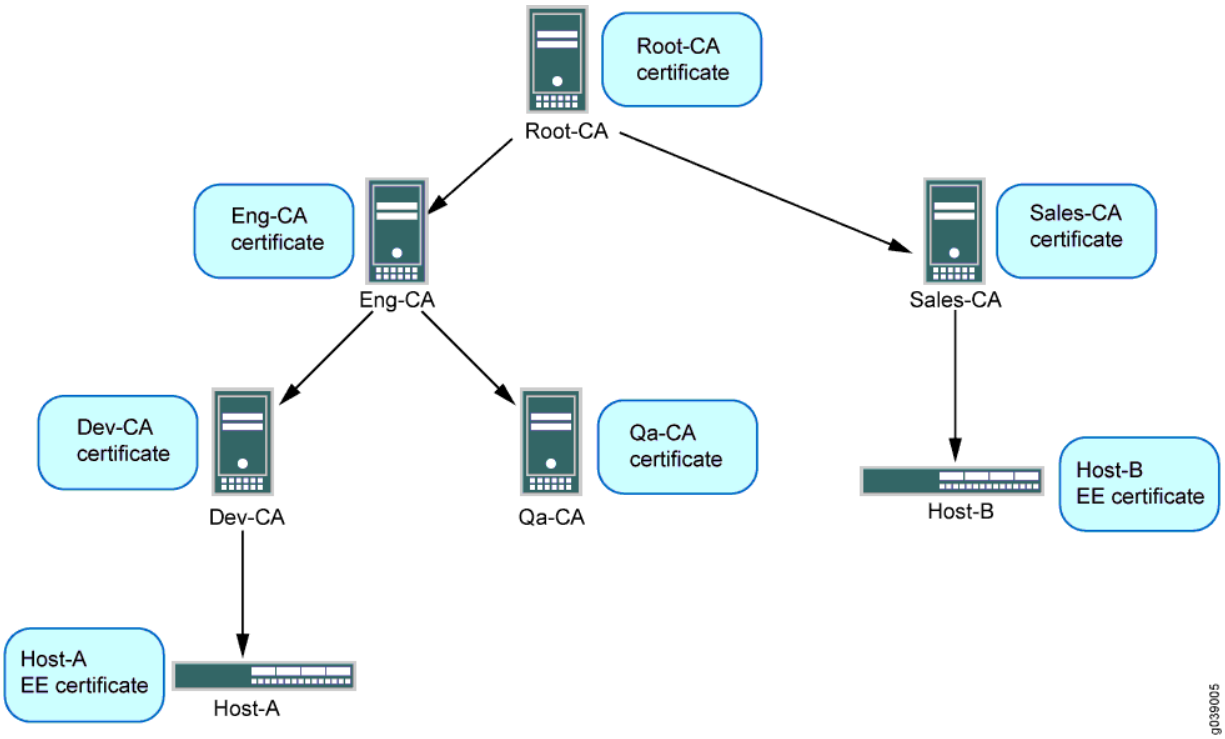
The network administrator needs to ensure that all peers participating in an IKE negotiation have at least one common trusted CA in their respective certificate chains. The common trusted CA does not have to be the root CA. The number of certificates in the chain, including certificates for EEs and the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

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. 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 19 on page 145, 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 19: 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

# IKE Protection from DDoS Attacks

IN THIS SECTION


- DDoS Vulnerabilities on IKE Implementations | 147
- Protection Against DDoS Attacks | 148
- Ways to Monitor DDoS Attacks | 149


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. The 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 refer to the session as a *half-open IKE session* until authentication happens with the IKE_AUTH message. After the peers establish an IKE security association (SA), the session is termed as 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 large amounts of SA_INIT messages (without IKE_AUTH messages) for which the attacker can create half-open IKE security association structures. This causes the device to utilize the resources and run out of memory.
- Send a large amount of junk IKE_AUTH packets with correct SPI_i and SPI_r on the initiator and the responder, respectively. This causes the device to run out of memory while trying to decrypt the packets.

- Send continuous SA_INIT packets. This causes the device to run out of memory while trying to generate keys for encrypted packets.
- Send large amounts of rekey requests per second during full open IKE sessions.
- Send large amounts of messages with distinct message IDs. It causes the device to queue all incoming IKE messages and run out of memory.

How to safeguard the IKE implementations

We provide robust infrastructure to mitigate and monitor the DDoS attacks for both IKEv1 and IKEv2 protocols. You can protect against the attacks on IKE implementations when your firewall runs the iked process (**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 do not support the client puzzle mechanism that is present in the RFC. While the IKEv2 DDoS protection is based on RFC 8019, we provide similar protection for IKEv1 when your firewall runs IPsec VPN service using the iked process.

## 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 security associations, and further managing the incoming exchange rates for the rekey requests. We provide the following measures to ensure protection:

- For half-open IKE SAs:
  - The responder does not allow configuration of half open IKE SA for a certain duration. You can set this limit so that the responder does not configure until it reaches the timeout duration. For more details, see *session (Security IKE)RLI41450 Review this entire topic* for the option `timeout`.
  - You can set the limit on the maximum allowed half open IKE SAs on the responder. When the total number of half open IKE SAs reach the maximum count, the responder rejects new connections for both IKEv1 and IKEv2 SAs. For more details, see *session (Security IKE)RLI41450 Review this entire topic* for the option `max-count`.
  - The responder enforces threshold values on the session count for half open IKE SAs. When the total number of half open IKE SAs reach 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 *session (Security IKE)RLI41450 Review this entire topic* for the options `thresholds`, `send-cookie` and `reduce-timeout`.

- The responder can discard duplicate sessions. For more details, see *session (Security IKE)RLI41450 Review this entire topic* for the option `discard-duplicate`.
- You can set backoff timeouts for authentication failure and initiation failure phases. For more details, see *session (Security IKE)RLI41450 Review this entire topic* 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 *session (Security IKE)RLI41450 Review this entire topic* for the option `incoming-exchange-max-rates`.
- The responder can block an incoming IKE session from peers based on the peer IKE ID. For more details, see *blocklists (Security IKE)RLI41450 Review this entire topic*.
- For dynamic gateways, you can set a limit for the number of connections at 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 150.

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 peersRLI41450*. 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 peersRLI41450*.
- 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 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 | **151**
- Configure the IKE Session for Full Open IKE SAs | **154**
- Configure the IKE Session Blocklists | **155**


### Prerequisites

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

- SRX Series Firewall that supports `junos-ike` package to run the IPsec VPN service using the iked process.
- SRX Series 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 | **151**
- Configuration | **151**


### 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 explicIty 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.

  - 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.

## Configure the IKE Session for Full Open IKE SAs

**IN THIS SECTION**


- Overview | 154
- Configuration | 154


### 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 | 155
- Configuration | 156


### 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 | 158
- Overview | 158
- Configuration | 159

- Verification | **167**
- IKE and IPsec SA Failure for a Revoked Certificate | **169**


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**

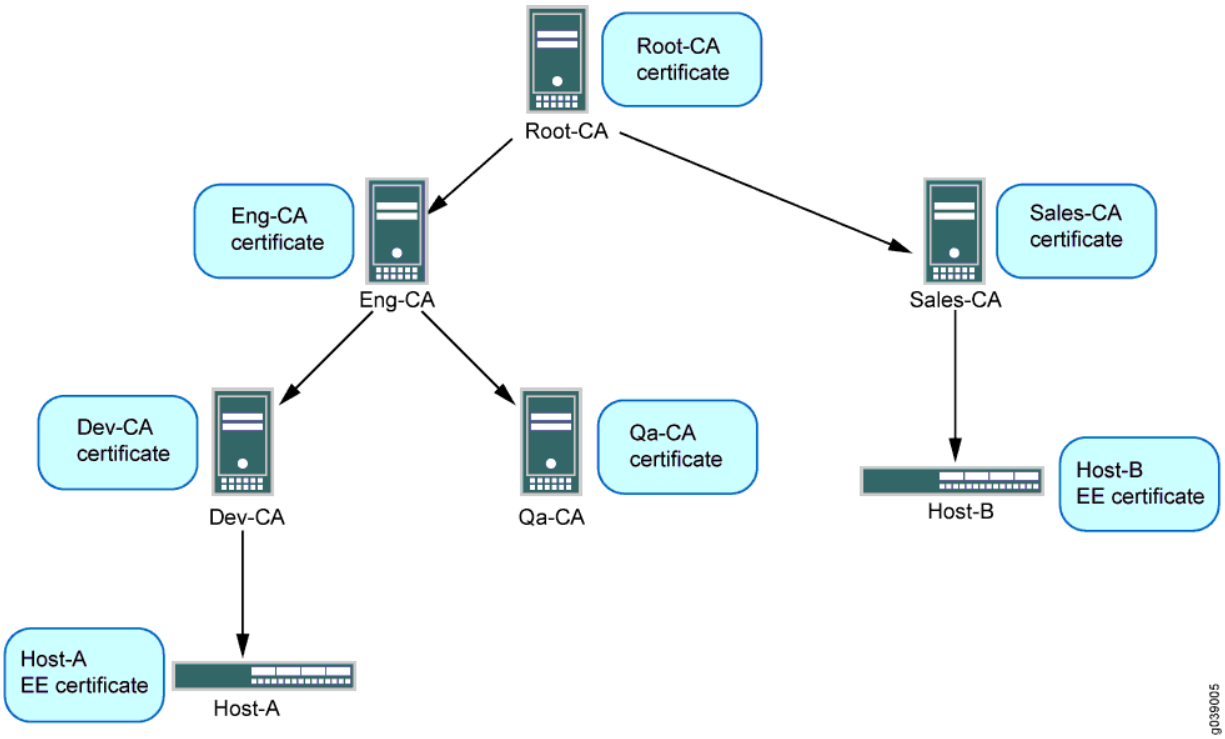

- Topology | **158**


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 20 on page 159. 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 20: 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 211 for examples of configuring peer devices for VPNs.

## Configuration

**IN THIS SECTION**


- Configure CA Profiles | **160**
- Enroll Certificates | **162**
- Configure IPsec VPN Options | **165**


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 | **168**
- Verifying IPsec Phase 2 Status | **168**

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 | 169


### 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

PKI Components In Junos OS | 33

Understanding Certificate Authority Profiles | 46

# IKEv2 Fragmentation

IN THIS SECTION


- Message Fragmentation | 171
- Configuration | 171
- Caveats | 172


## 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 SRX Series 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 SRX Series 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**

Understanding Certificate Authority Profiles | 46

## 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**

Understanding Certificate Authority Profiles | 46

## 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.

Starting in Junos OS Release 19.1R1, on SRX5000 line, the establish tunnels option supports the `responder-only` and `responder-only-no-rekey` values under the `[edit security ipsec vpn vpn-name]` hierarchy-level.

The `responder-only` and `responder-only-no-rekey` options are supported on the SRX5000 line with an SPC3 card only if the `junos-ike-package` is installed. 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 20.

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.

**Release History Table**

| Release | Description |
|---|---|
| 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 | 43

# IPsec VPN Overview

IN THIS SECTION


- IPsec VPN Topologies on SRX Series Firewalls | 177
- Comparing Policy-Based and Route-Based VPNs | 177
- Comparison of Policy-Based VPNs and Route-Based VPNs | 180
- Understanding IKE and IPsec Packet Processing | 181
- Distribution of IKE and IPsec Sessions Across SPUs | 184
- VPN Support for Inserting Services Processing Cards | 186
- Cryptographic acceleration support on SRX5K-SPC3 Card, SRX mid-range platforms and vSRX Virtual Firewall | 188
- IPsec VPN Feature Support on SRX5000 Line with SRX5K-SPC3 and vSRX Virtual Firewall Instances with New Package | 190
- Routing Protocols Support on IPsec VPN Tunnels | 191
- Anti-Replay Window | 197
- Understanding Hub-and-Spoke VPNs | 198


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 IP Security (IPsec) tunnel.

The term *tunnel* does not denote tunnel mode (see "Packet Processing in Tunnel Mode" on page 181). Instead, it refers to the IPsec connection.

## 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*.

**SEE ALSO**

Example: Configuring a Hub-and-Spoke VPN | 221

## Comparing Policy-Based and Route-Based VPNs

It is important to understand the differences between policy-based and route-based VPNs and why one might be preferable to the other.

Table 6 on page 177 lists the differences between route-based VPNs and policy-based VPNs.

**Table 6: Differences Between Route-Based VPNs and Policy-Based VPNs**

| Route-Based VPNs | Policy-Based VPNs |
|---|---|
| With route-based VPNs, a policy does not specifically reference a VPN tunnel. | With policy-based VPN tunnels, a tunnel is treated as an object that, together with source, destination, application, and action, constitutes a tunnel policy that permits VPN traffic. |

**Table 6: Differences Between Route-Based VPNs and Policy-Based VPNs *(Continued)***

| Route-Based VPNs | Policy-Based VPNs |
|---|---|
| The policy references a destination address. | In a policy-based VPN configuration, a tunnel policy specifically references a VPN tunnel by name. |
| The number of route-based VPN tunnels that you create is limited by the number of route entries or the number of st0 interfaces that the device supports, whichever number is lower. | The number of policy-based VPN tunnels that you can create is limited by the number of policies that the device supports. |
| Route-based VPN tunnel configuration is a good choice when you want to conserve tunnel resources while setting granular restrictions on VPN traffic. | 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 security association (SA) with the remote peer. Each SA counts as an individual VPN tunnel. |
| With a route-based approach to VPNs, the regulation of traffic is not coupled to the means of its delivery. You can configure dozens of policies to regulate traffic flowing through a single VPN tunnel between two sites, and only one IPsec SA is at work. Also, a route-based VPN configuration allows you to create policies referencing a destination reached through a VPN tunnel in which the action is deny. | In a policy-based VPN configuration, the action must be permit and must include a tunnel. |
| 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. | The exchange of dynamic routing information is not supported in policy-based VPNs. |
| Route-based configurations are used for hub-and-spoke topologies. | Policy-based VPNs cannot be used for hub-and-spoke topologies. |
| With route-based VPNs, a policy does not specifically reference a VPN tunnel. | 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, a policy-based tunnel is the best choice. |

**Table 6: Differences Between Route-Based VPNs and Policy-Based VPNs *(Continued)***

| Route-Based VPNs | Policy-Based VPNs |
|---|---|
| Route-based VPNs do not support remote-access (dial-up) VPN configurations. | Policy-based VPN tunnels are required for remote-access (dial-up) VPN configurations. |
| Route-based VPNs might not work correctly with some third-party vendors. | Policy-based VPNs might be required if the third party requires separate SAs for each remote subnet. |
| 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 via a secure tunnel interface (`st0`) , which is bound to a specific VPN tunnel.<br><br>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. | With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy. |
| Route-based VPNs support NAT for st0 interfaces. | Policy-based VPNs cannot be used if NAT is required for tunneled 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.

Policy-based VPNs are only supported on SRX5400, SRX5600, and SRX5800 line. Platform support depends on the Junos OS release in your installation.

## SEE ALSO

Example: Configuring a Route-Based VPN | **407**

Example: Configuring a Policy-Based VPN | **280**

## Comparison of Policy-Based VPNs and Route-Based VPNs

Table 7 on page 180 summarizes the differences between policy-based VPNs and route-based VPNs.

**Table 7: 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. |
| If you need more granularity than a route can provide to specify the traffic sent to a tunnel, using a policy-based VPN with security policies is the best choice. | Route-based VPNs uses routes to specify the traffic sent to a tunnel; a policy does not specifically reference a VPN tunnel. |

**Table 7: Comparison Between Policy-Based VPNs and Route-Based VPNs *(Continued)***

| Policy-Based VPNs | Route-Based VPNs |
|---|---|
| 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.<br><br>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. |

## Understanding IKE and IPsec Packet Processing

**IN THIS SECTION**


- Packet Processing in Tunnel Mode | **181**


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" on page 20.) 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.

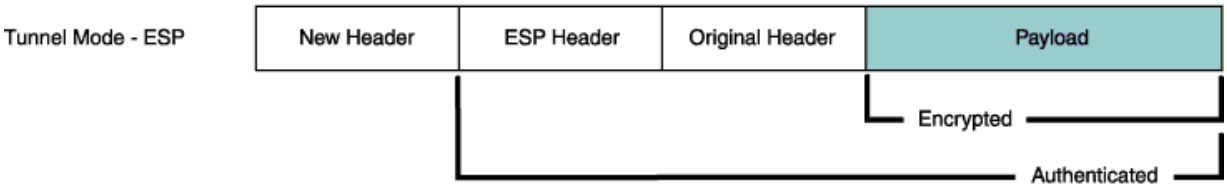
This topic includes the following sections:

### 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.

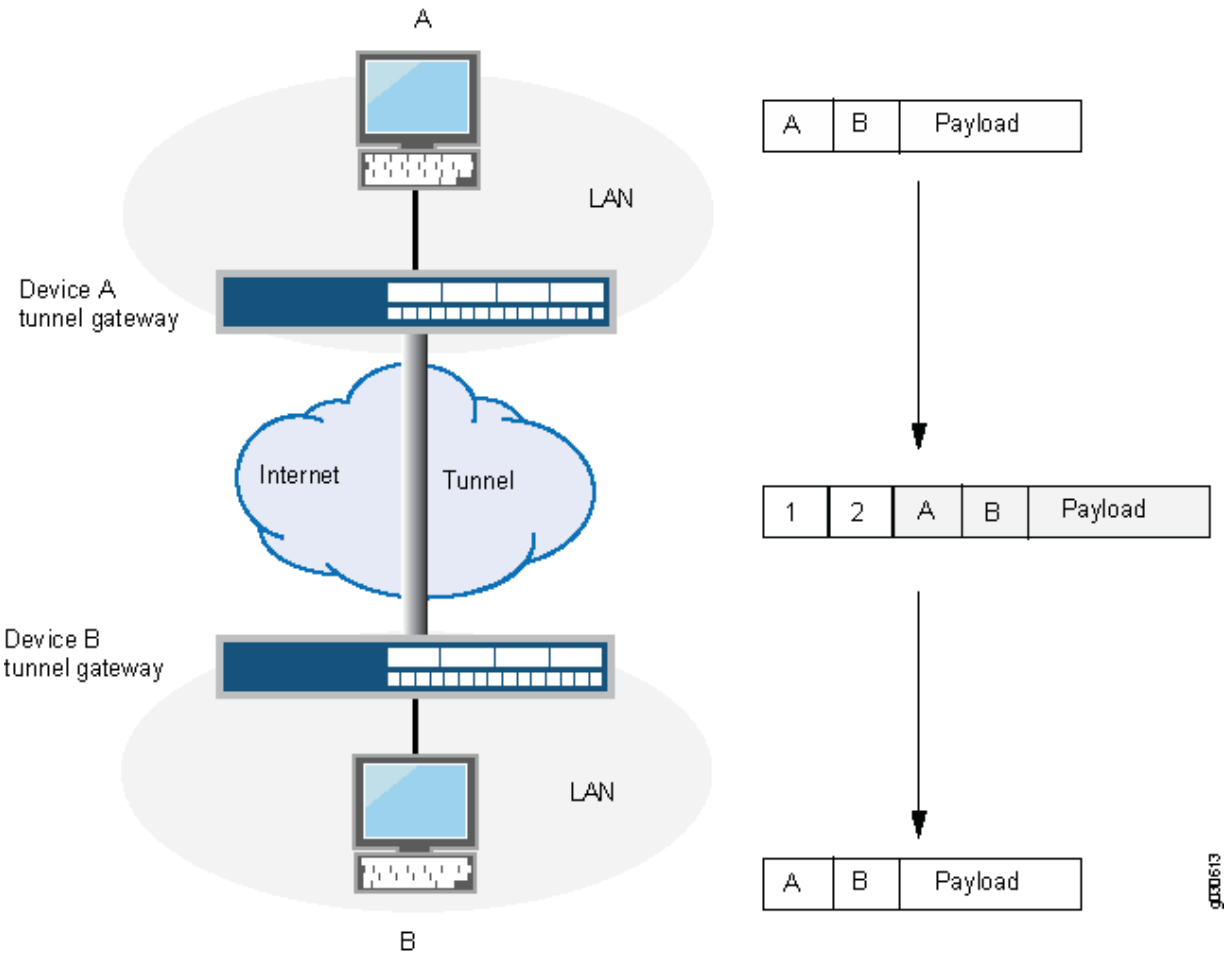
In tunnel mode, the entire original IP packet—payload and header—is encapsulated within another IP payload, and a new header is appended to it, as shown in Figure 21 on page 182. The entire original packet can be encrypted, authenticated, or both. With the Authentication Header (AH) protocol, the AH and new headers are also authenticated. With the Encapsulating Security Payload (ESP) protocol, the ESP header can also be authenticated.

**Figure 21: 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 22 on page 183.

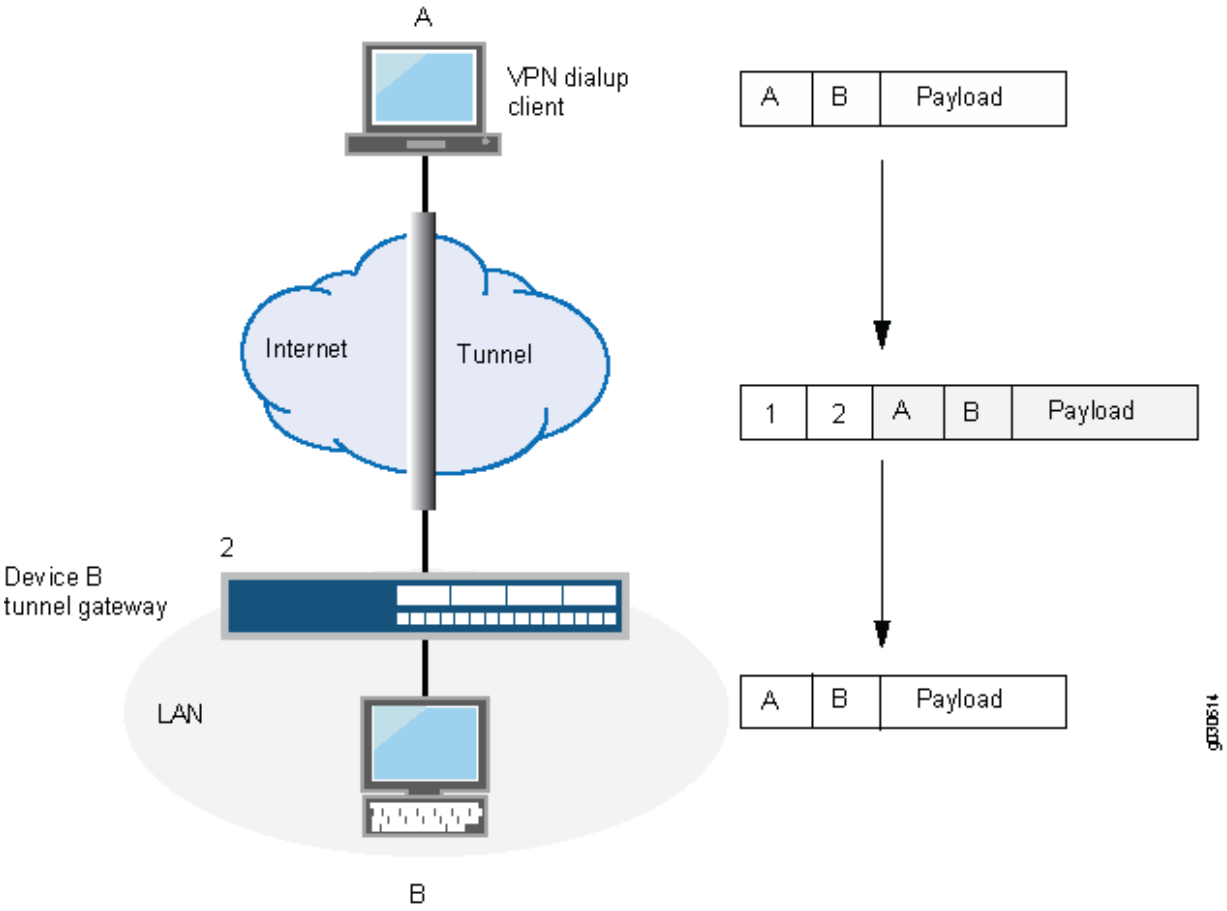
**Figure 22: Site-to-Site VPN in Tunnel Mode**




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 23 on page 184). 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 23: Dial-Up VPN in Tunnel Mode**




**SEE ALSO**

Example: Configuring a Policy-Based VPN | 280

Route-Based IPsec VPNs | 406

## Distribution of IKE and IPsec Sessions Across SPUs

In the SRX5400, SRX5600, and SRX5800 devices, 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 8 on page 185 shows an example of an SRX5000 line with three SPUs running seven IPsec tunnels over three IKE gateways.

**Table 8: 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

SRX5400, SRX5600, and SRX5800 devices 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.

In an SRX5400, SRX5600, or SRX5800 chassis cluster, you can insert new SPCs on the devices without affecting or disrupting the traffic on the existing IKE or IPsec VPN tunnels. When you insert a new SPC in each chassis of the cluster, the existing tunnels are not affected and traffic continues to flow without disruption.

Starting in Junos OS Release 19.4R1, on all SRX5000 line chassis cluster, you can insert a new SRX5K-SPC3 (SPC3) or SRX5K-SPC-4-15-320 (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. You must reboot the node after the inserting SPC3 to activate the card. After the node reboot is complete, IPsec tunnels are distributed to the cards.

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.

Starting in Junos OS Release 18.2R2 and 18.4R1, all the existing IPsec VPN features that are currently supported on SRX5K-SPC3 (SPC3) only will be supported on SRX5400, SRX5600, and SRX5800 devices when SRX5K-SPC-4-15-320 (SPC2) and SPC3 cards are installed and operating on the device in a chassis cluster mode or in a standalone mode.

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.
- On SRX5800 chassis cluster, you must not insert the SPC3 card in the highest slot (slot no. 11) due to the power and heat distribution limit.
- We do not support SPC3 hot removal.

Table 9 on page 187 summarizes the SRX5000 line with SPC2 or SPC3 card that supports `kmd` or `iked` process:

**Table 9: `kmd/iked` Process Support on SRX5000 Line**

| SRX5000 Line | Support for `kmd` or `iked` Process |
|---|---|
| SRX5000 line with only SPC2 card installed | Supports `kmd` process. |
| SRX5000 line with only SPC3 card installed | Supports `iked` process. |
| SRX5000 line with both SPC2 and SPC3 card installed | Supports `iked` process. |

## SEE ALSO

*show security ike tunnel-map*

## Cryptographic acceleration support on SRX5K-SPC3 Card, SRX mid-range platforms and vSRX Virtual Firewall

SRX5000 line with SRX5K-SPC3 card (Services Processing Card), SRX mid-range platforms (SRX4100, SRX4200, SRX1500 and SRX4600 Series Firewalls) and vSRX Virtual Firewall requires `junos-ike` package as the control plane software to install and enable IPsec VPN features.

- On SRX5000 line with RE3, by default, `junos-ike` package is installed in Junos OS Releases 20.1R2, 20.2R2, 20.3R2, 20.4R1, and later. As a result, **iked** and **ikemd** process runs on the routing engine by default instead of IPsec key management daemon (kmd). The SRX5000 line with SRX5K-SPC3 offloads the cryptographic operations to the hardware cryptographic engine.
- The SRX mid-range platforms covering SRX1500, SRX4100, SRX4200 and SRX4600 Series Firewalls, offloads the DH, RSA and ECDSA cryptographic operations to the hardware cryptographic engine with devices running `junos-ike` software. This feature is available from Junos OS Release 23.2R1 with devices having `junos-ike` package installed. The devices that continue to run legacy **iked** software (kmd process) do not support this feature.
- On vSRX Virtual Firewall, the data plane CPU thread offloads the DH, RSA and ECDSA operations. Hardware acceleration is not available on these devices. This feature is available from Junos OS Release 23.2R1 with `junos-ike` package installed on the device.

The Table 10 on page 188 describes the hardware acceleration support for various ciphers:

**Table 10: Cryptographic Acceleration Support**

| Ciphers | SRX1500 | | SRX4100/ SRX4200 | | SRX4600 | | SRX5K - SPC3 | vSRX3.0 | |
|---|---|---|---|---|---|---|---|---|---|
| | KMD | IKED | KMD | IKED | KMD | IKED | IKED | KMD | IKED |
| DH (Groups 1, 2, 5, 14) | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| DH (Groups 19, 20) | No | Yes | No | Yes | No | Yes | Yes | Yes | Yes |
| DH (Groups 15, 16) | No | Yes | No | Yes | No | Yes | Yes | Yes | Yes |
| DH Group 21 | No | Yes | No | Yes | No | Yes | Yes | Yes | Yes |

**Table 10: Cryptographic Acceleration Support *(Continued)***

| Ciphers | SRX1500 | | SRX4100/ SRX4200 | | SRX4600 | | SRX5K - SPC3 | vSRX3.0 | |
|---|---|---|---|---|---|---|---|---|---|
| DH Group 24 | Yes | Yes | Yes | Yes | Yes | Yes | No | No | No |
| RSA | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| ECDSA (256, 384, 521) | No | Yes | No | Yes | No | Yes | Yes | Yes | Yes |

To install the Junos IKE package on your 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 use `kmd` process in order to enable IPsec VPN features on SRX5000 line without an SPC3 card, you must run the `request system software delete junos-ike` command. After running the command, 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}
```

SEE ALSO

*show security ipsec security-associations*

*show security ipsec tunnel-distribution*

# IPsec VPN Feature Support on SRX5000 Line with SRX5K-SPC3 and vSRX Virtual Firewall Instances with New Package

IN THIS SECTION


- IPsec VPN Features Not Supported | **190**


This topic provides you a summary of IPsec VPN features and configurations that are not supported of SRX5000 line with SRX5K-SPC3 and on vSRX Virtual Firewall instances.

IPsec VPN feature is supported by two processes, **iked** and **ikemd** on SRX5K-SPC3 and vSRX Virtual Firewall instances. A single instance of **iked** and **ikemd** will run on the Routing Engine at a time.

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, and both the **iked** and **ikemd** process runs on the routing engine.

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

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

If you want to use `kmd` process to enable IPsec VPN features on SRX5000 line without an SRX5K-SPC3 card, you must run the `request system software delete junos-ike` command. After running the command, you must reboot the device.

## IPsec VPN Features Not Supported

To determine if a feature is supported by a specific platform or Junos OS release, refer Feature Explorer.

Table 11 on page 191 summarizes the non-supported IPsec VPN features on SRX Series Firewalls and vSRX Virtual Firewall running iked process:

**Table 11: IPsec VPN Features Not Supported on SRX Series Firewalls and vSRX Virtual Firewall Instances**

| Features | Support on SRX5000 line with SRX5K-SPC3 and vSRX Virtual Firewall Instances |
|---|---|
| AutoVPN Protocol Independent Multicast (PIM) point-to-multipoint mode. | No |
| Configuring forwarding class on IPsec VPNs. | No |
| Group VPN. | No |
| Packet size configuration for IPsec datapath verification. | No |
| Policy-based IPsec VPN. | No |

## Routing Protocols Support on IPsec VPN Tunnels

We support routing protocols like, OSPF, BGP, PIM, RIP, and BFD to run on IPsec tunnels on SRX Series Firewalls and MX Series routers running `kmd` or `iked` process. The protocol support varies based on the IP addressing scheme or the type of the st0 interface, point-to-point (P2P) or point-to-multipoint (P2MP).

Table 12 on page 192 summarizes OSPF protocol support on SRX Series Firewalls and MX routers.

**Table 12: OSPF Protocol Support on IPsec VPN Tunnels**

| OSPF | | | | |
|---|---|---|---|---|
| **Devices** | **P2P** | | **P2MP** | |
| | **IPv4** | **IPv6** | **IPv4** | **IPv6** |
| SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2 | Yes | No | Yes | No |
| SRX5K-SPC3 | Yes | No | Yes | No |
| SRX5K in mixed-mode (SPC3 + SPC2) | Yes | No | Yes | No |
| vSRX Virtual Firewall 3.0 | Yes | No | Yes | No |
| MX-SPC3 | Yes | No | No | No |

Table 13 on page 193 summarizes OSPFv3 protocol support on SRX Series Firewalls and MX routers.

**Table 13: OSPFv3 Protocol Support on IPsec VPN Tunnels**

| OSPFv3 | | | | |
|---|---|---|---|---|
| **Devices** | **P2P** | | **P2MP** | |
| | **IPv4** | **IPv6** | **IPv4** | **IPv6** |
| SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2 | No | Yes | No | Yes |
| SRX5K-SPC3 | No | Yes | No | Yes |
| SRX5K in mixed-mode (SPC3 + SPC2) | No | Yes | No | Yes |
| vSRX Virtual Firewall 3.0 | No | Yes | No | Yes |
| MX-SPC3 | No | Yes | No | No |

Table 14 on page 194 summarizes BGP protocol support on SRX Series Firewalls and MX routers.

**Table 14: BGP Protocol Support on IPsec VPN Tunnels**

| BGP | | | | |
|---|---|---|---|---|
| **Devices** | **P2P** | | **P2MP** | |
| | **IPv4** | **IPv6** | **IPv4** | **IPv6** |
| SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2 | Yes | Yes | Yes | Yes |
| SRX5K-SPC3 | Yes | Yes | Yes | Yes |
| SRX5K in mixed-mode (SPC3 + SPC2) | Yes | Yes | Yes | Yes |
| vSRX Virtual Firewall 3.0 | Yes | Yes | Yes | Yes |
| MX-SPC3 | Yes | Yes | No | No |

Table 15 on page 195 summarizes PIM protocol support on SRX Series Firewalls and MX routers.

**Table 15: PIM Protocol Support on IPsec VPN Tunnels**

| PIM | | | | |
|---|---|---|---|---|
| **Devices** | **P2P** | | **P2MP** | |
| | **IPv4** | **IPv6** | **IPv4** | **IPv6** |
| SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, SRX550 HM, SRX650, SRX1400, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2 | Yes | No | No | No |
| SRX300, SRX320, SRX340, SRX345, SRX380, and SRX1500 | Yes | No | Yes | No |
| SRX5K-SPC3 | Yes | No | No | No |
| SRX5K in mixed-mode (SPC3 + SPC2) | Yes | No | No | No |
| vSRX Virtual Firewall | Yes | No | Yes<br>**NOTE**: Multithread is not supported. | No |
| MX-SPC3 | Yes | No | No | No |

Table 16 on page 196 summarizes RIP protocol support on SRX Series Firewalls and MX routers.

**Table 16: RIP Protocol Support on IPsec VPN Tunnels**

| RIP | | | | |
|---|---|---|---|---|
| **Devices** | **P2P** | | **P2MP** | |
| | **IPv4** | **IPv6** | **IPv4** | **IPv6** |
| SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2 | Yes | Yes | No | No |
| SRX5K-SPC3 | Yes | Yes | No | No |
| SRX5K in mixed-mode (SPC3 + SPC2) | Yes | Yes | No | No |
| vSRX Virtual Firewall 3.0 | Yes | Yes | No | No |
| MX-SPC3 | Yes | Yes | No | No |

Table 17 on page 197 summarizes BFP protocol support on SRX Series Firewalls and MX routers.

**Table 17: BFD Protocol Support on IPsec VPN Tunnels**

| BFD | | | | |
|---|---|---|---|---|
| Devices | P2P | | P2MP | |
| | IPv4 | IPv6 | IPv4 | IPv6 |
| SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2 | Yes | Yes | Yes | Yes |
| SRX5K-SPC3 | Yes | Yes | Yes | Yes |
| SRX5K in mixed-mode (SPC3 + SPC2) | Yes | Yes | Yes | Yes |
| vSRX Virtual Firewall 3.0 | Yes | Yes | Yes | Yes |
| MX-SPC3 | Yes | Yes | No | No |

## Anti-Replay Window

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

On the SRX Series 5000 line with SPC3 cards installed, you can configure the `anti-replay-window` size in the range of 64 to 8192 (power of 2). 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*

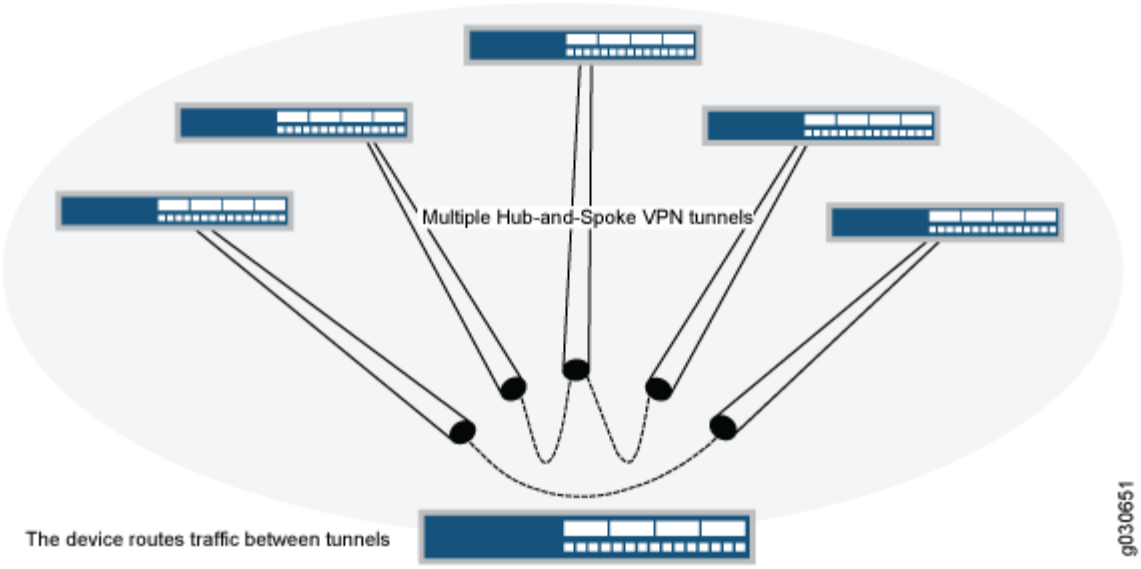
## Understanding Hub-and-Spoke VPNs

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 24 on page 199.)

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 24: Multiple Tunnels in a Hub-and-Spoke VPN Configuration**




## SEE ALSO

Example: Configuring a Hub-and-Spoke VPN | 221

**Release History Table**

| Release | Description |
|---|---|
| 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 | **406**

Policy-Based IPsec VPNs | **279**

5

CHAPTER

# VPN Configuration Overview

IPsec VPN Configuration Overview | 202

Comparing Policy-Based and Route-Based VPNs | 272

Chassis Cluster HA Control Link Encryption | 275

# IPsec VPN Configuration Overview

IN THIS SECTION


- IPsec VPN with Autokey IKE Configuration Overview | 202
- Recommended Configuration Options for Site-to-Site VPN with Static IP Addresses | 203
- Recommended Configuration Options for Site-to-Site or Dialup VPNs with Dynamic IP Addresses | 205
- Understanding IPsec VPNs with Dynamic Endpoints | 207
- Understanding IKE Identity Configuration | 209
- Configuring Remote IKE IDs for Site-to-Site VPNs | 211
- Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | 211
- Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Firewall | 214
- Configuring IPsec VPN Using the VPN Wizard | 220
- Example: Configuring a Hub-and-Spoke VPN | 221


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 or policy-based IPsec VPN using autokey IKE (preshared keys or certificates).

To configure a route-based or policy-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. Configure a security policy to permit traffic from the source zone to the destination zone.

   (For policy-based VPNs) Specify the security policy action `tunnel ipsec-vpn` with the name of the IPsec VPN tunnel that you configured.

5. Update your global VPN settings.

**SEE ALSO**

Understanding Route-Based IPsec VPNs | **406**

Understanding Policy-Based IPsec VPNs | **279**

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

Table 18 on page 204 lists the configuration options for a generic site-to-site VPN between two security devices with static IP addresses. The VPN can be either route-based or policy-based.

**Table 18: 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. |
| 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. |

**Table 18: Recommended Configuration for Site-to-Site VPN with Static IP Addresses *(Continued)***

| Configuration Option | Comment |
|---|---|
| Anti-replay protection | Enabled by default. Disabling this feature might resolve compatibility issues with third-party peers. |

**SEE ALSO**

IPsec Overview | 20

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

Table 19 on page 205 lists the configuration options for a generic site-to-site or dialup VPN, where the peer devices have dynamic IP addresses.

**Table 19: 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. |
| 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. |

**Table 19: Recommended Configuration for Site-to-Site or Dialup VPNs with Dynamic IP Addresses *(Continued)***

| Configuration Option | Comment |
|---|---|
| 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 | 20

# Understanding IPsec VPNs with Dynamic Endpoints

IN THIS SECTION


- Overview | 207
- IKE Identity | 207
- Aggressive Mode for IKEv1 Policy | 208
- IKE Policies and External Interfaces | 208
- NAT | 208
- Group and Shared IKE IDs | 208


## 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. Starting with Junos OS Release 15.1X49-D80, 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

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. 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 | 690

# Understanding IKE Identity Configuration

IN THIS SECTION


- IKE ID Types | 209
- Remote IKE IDs and Site-to-Site VPNs | 210
- Remote IKE IDs and Dynamic Endpoint VPNs | 210
- Local IKE ID of the SRX Series Firewall | 210


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.

  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.

- 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 | 1039

## 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 | 605

Example: Configuring a Route-Based VPN with the Responder behind a NAT Device | 606

Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device | 643

## 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 | 20

# Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Firewall

IN THIS SECTION


- Requirements | 214
- Overview | 214
- Configuration | 215
- Verification | 219


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 20 on page 214 lists the parameters and values configured for the manual SA in this example.

**Table 20: Manual SA for IPsec OSPF Interface Authentication**

| Parameter | Value |
|---|---|
| SA name | sa1 |

**Table 20: Manual SA for IPsec OSPF Interface Authentication *(Continued)***

| Parameter | Value |
|---|---|
| Mode | transport |
| Direction | bidirectional |
| Protocol | AH |
| SPI | 256 |
| Authentication algorithm<br>Key | hmac-md5-96<br>(ASCII) 123456789012abc |
| Encryption algorithm<br>Key | des<br>(ASCII) cba210987654321 |

## Configuration

**IN THIS SECTION**


- Configuring a Manual SA | **215**
- Enabling IPsec Authentication for an OSPF Interface | **218**


### 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 | 220
- Verifying the IPsec Security Association on the OSPF Interface | 220


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 | **727**

## 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 | 20

Internet Key Exchange | 10

## Example: Configuring a Hub-and-Spoke VPN

**IN THIS SECTION**

- Requirements | 221
- Overview | 222
- Configuration | 233
- Verification | 264

This example shows how to configure a hub-and-spoke IPsec VPN for an enterprise-class deployment. For site-to-site IPSec VPN with IKEv1 and IKEv2, see Route-Based IPsec VPN with IKEv1 and Route-Based IPsec VPN with IKEv1 respectively.

### Requirements

This example uses the following hardware:

- SRX240 device
- SRX5800 device
- SSG140 device

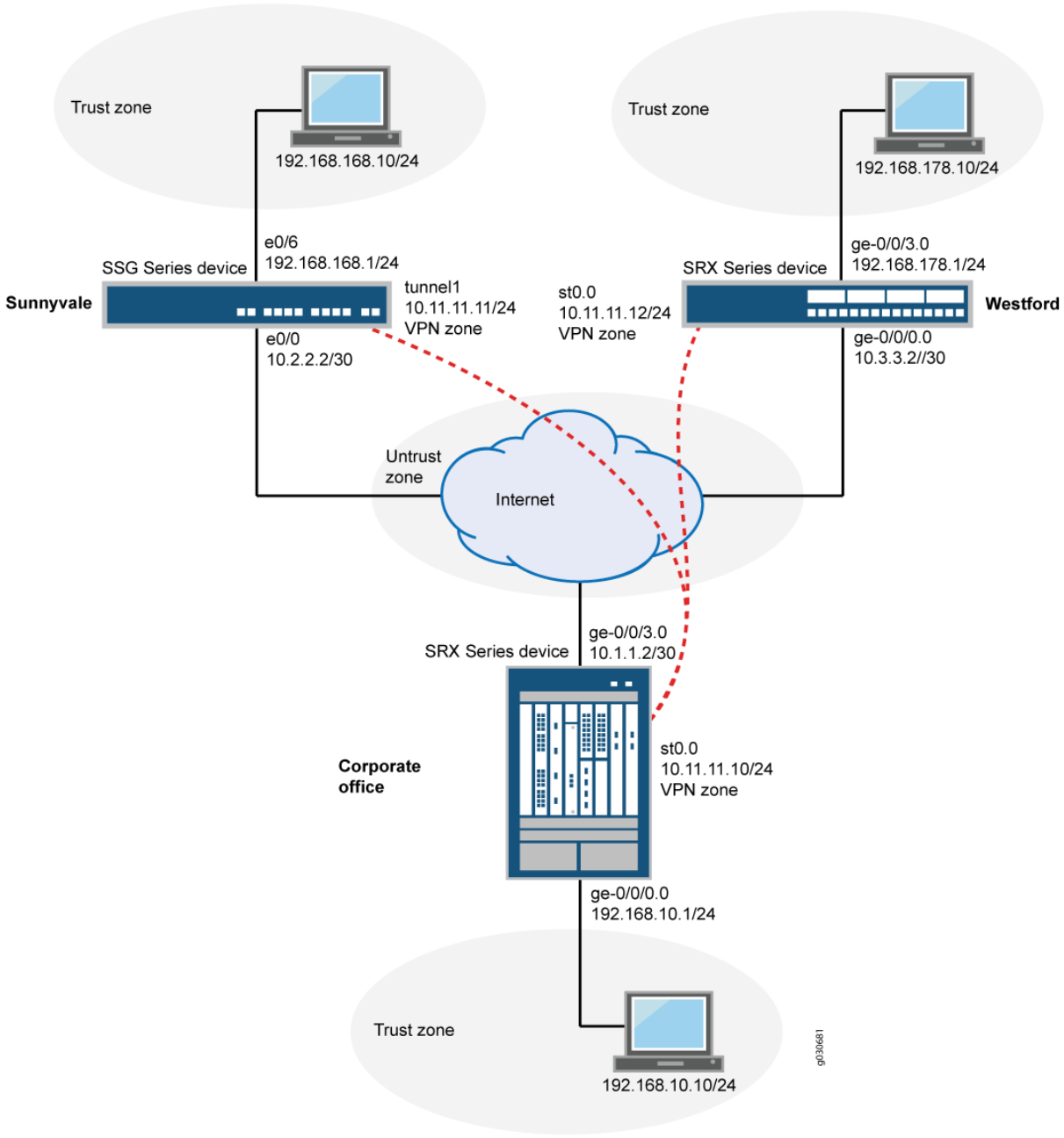
Before you begin, read "IPsec Overview" on page 20.

## Overview

This example describes how to configure a hub-and-spoke VPN typically found in branch deployments. The hub is the corporate office, and there are two spokes—a branch office in Sunnyvale, California, and a branch office in Westford, Massachusetts. Users in the branch offices will use the VPN to securely transfer data with the corporate office.

Figure 25 on page 223 shows an example of a hub-and-spoke VPN topology. In this topology, an SRX5800 device is located at the corporate office. An SRX Series Firewall is located at the Westford branch, and an SSG140 device is located at the Sunnyvale branch.

**Figure 25: Hub-and-Spoke VPN Topology**



In this example, you configure the corporate office hub, the Westford spoke, and the Sunnyvale spoke. First you configure interfaces, IPv4 static and default routes, security zones, and address books. Then you configure IKE Phase 1 and IPsec Phase 2 parameters, and bind the st0.0 interface to the IPsec VPN. On the hub, you configure st0.0 for multipoint and add a static NHTB table entry for the Sunnyvale spoke. Finally, you configure security policy and TCP-MSS parameters. See Table 21 on page 224 through Table 25 on page 232 for specific configuration parameters used in this example.

**Table 21: Interface, Security Zone, and Address Book Information**

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| Hub | Interfaces | ge-0/0/0.0 | 192.168.10.1/24 |
| | | ge-0/0/3.0 | 10.1.1.2/30 |
| | | st0 | 10.11.11.10/24 |
| Spoke | Interfaces | ge-0/0/0.0 | 10.3.3.2/30 |
| | | ge-0/0/3.0 | 192.168.178.1/24 |
| | | st0 | 10.11.11.12/24 |
| Hub | Security zones | trust | • All system services are allowed.<br>• The ge-0/0/0.0 interface is bound to this zone. |
| | | untrust | • IKE is the only allowed system service.<br>• The ge-0/0/3.0 interface is bound to this zone. |
| | | vpn | The st0.0 interface is bound to this zone. |

**Table 21: Interface, Security Zone, and Address Book Information *(Continued)***

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| Spoke | Security zones | trust | • All system services are allowed.<br>• The ge-0/0/3.0 interface is bound to this zone. |
| | | untrust | • IKE is the only allowed system service.<br>• The ge-0/0/0.0 interface is bound to this zone. |
| | | vpn | The st0.0 interface is bound to this zone. |
| Hub | Address book entries | local-net | • This address is for the trust zone's address book.<br>• The address for this address book entry is 192.168.10.0/24. |
| | | sunnyvale-net | • This address book is for the vpn zone's address book.<br>• The address for this address book entry is 192.168.168.0/24. |

**Table 21: Interface, Security Zone, and Address Book Information *(Continued)***

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| | | westford-net | • This address is for the vpn zone's address book.<br>• The address for this address book entry is 192.168.178.0/24. |
| Spoke | Address book entries | local-net | • This address is for the trust zone's address book.<br>• The address for this address book entry is 192.168.168.178.0/24. |
| | | corp-net | • This address is for the vpn zone's address book.<br>• The address for this address book entry is 192.168.10.0/24. |
| | | sunnyvale-net | • This address is for the vpn zone's address book.<br>• The address for this address book entry is 192.168.168.0/24. |

**Table 22: IKE Phase 1 Configuration Parameters**

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| Hub | Proposal | ike-phase1-proposal | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: sha1<br>• Encryption algorithm: aes-128-cbc |
| | Policy | ike-phase1-policy | • Mode: main<br>• Proposal reference: ike-phase1-proposal<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |
| | Gateway | gw-westford | • IKE policy reference: ike-phase1-policy<br>• External interface: ge-0/0/3.0<br>• Gateway address: 10.3.3.2 |

**Table 22: IKE Phase 1 Configuration Parameters *(Continued)***

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| | | gw-sunnyvale | • IKE policy reference: ike-phase1-policy<br>• External interface: ge-0/0/3.0<br>• Gateway address: 10.2.2.2 |
| Spoke | Proposal | ike-phase1-proposal | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: sha1<br>• Encryption algorithm: aes-128-cbc |
| | Policy | ike-phase1-policy | • Mode: main<br>• Proposal reference: ike-phase1-proposal<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |

**Table 22: IKE Phase 1 Configuration Parameters *(Continued)***

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| | Gateway | gw-corporate | • IKE policy reference: ike-phase1-policy<br>• External interface: ge-0/0/0.0<br>• Gateway address: 10.1.1.2 |

**Table 23: IPsec Phase 2 Configuration Parameters**

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| Hub | Proposal | ipsec-phase2-proposal | • Protocol: esp<br>• Authentication algorithm: hmac-sha1-96<br>• Encryption algorithm: aes-128-cbc |
| | Policy | ipsec-phase2-policy | • Proposal reference: ipsec-phase2-proposal<br>• PFS: Diffie-Hellman group2 |
| | VPN | vpn-sunnyvale | • IKE gateway reference: gw-sunnyvale<br>• IPsec policy reference: ipsec-phase2-policy<br>• Bind to interface: st0.0 |
| | | vpn-westford | • IKE gateway reference: gw-westford<br>• IPsec policy reference: ipsec-phase2-policy<br>• Bind to interface: st0.0 |

**Table 23: IPsec Phase 2 Configuration Parameters *(Continued)***

| Hub or Spoke | Feature | Name | Configuration Parameters |
|---|---|---|---|
| Spoke | Proposal | ipsec-phase2-proposal | • Protocol: esp<br>• Authentication algorithm: hmac-sha1-96<br>• Encryption algorithm: aes-128-cbc |
| | Policy | ipsec-phase2-policy | • Proposal reference: ipsec-phase2-proposal<br>• PFS: Diffie-Hellman group2 |
| | VPN | vpn-corporate | • IKE gateway reference: gw-corporate<br>• IPsec policy reference: ipsec-phase2-policy<br>• Bind to interface: st0.0 |

**Table 24: Security Policy Configuration Parameters**

| Hub or Spoke | Purpose | Name | Configuration Parameters |
|---|---|---|---|
| Hub | The security policy permits traffic from the trust zone to the vpn zone. | local-to-spokes | • Match criteria:<br>  • source-address local-net<br>  • destination-address sunnyvale-net<br>  • destination-address westford-net<br>  • application any |

**Table 24: Security Policy Configuration Parameters *(Continued)***

| Hub or Spoke | Purpose | Name | Configuration Parameters |
|---|---|---|---|
| | The security policy permits traffic from the vpn zone to the trust zone. | spokes-to-local | Match criteria:<br>• source-address sunnyvale-net<br>• source-address westford-net<br>• destination-address local-net<br>• application any |
| | The security policy permits intrazone traffic. | spoke-to-spoke | Match criteria:<br>• source-address any<br>• destination-address any<br>• application any |
| Spoke | The security policy permits traffic from the trust zone to the vpn zone. | to-corp | • Match criteria:<br>  • source-address local-net<br>  • destination-address corp-net<br>  • destination-address sunnyvale-net<br>  • application any |
| | The security policy permits traffic from the vpn zone to the trust zone. | from-corp | Match criteria:<br>• source-address corp-net<br>• source-address sunnyvale-net<br>• destination-address local-net<br>• application any |

**Table 24: Security Policy Configuration Parameters *(Continued)***

| Hub or Spoke | Purpose | Name | Configuration Parameters |
|---|---|---|---|
| | The security policy permits traffic from the untrust zone to the trust zone. | permit-any | Match criteria:<br>• source-address any<br>• source-destination any<br>• application any<br>• Permit action: source-nat interface<br>By specifying `source-nat interface`, the SRX Series Firewall translates the source IP address and port for outgoing traffic, using the IP address of the egress interface as the source IP address and a random high-number port for the source port. |

**Table 25: TCP-MSS Configuration Parameters**

| Purpose | Configuration Parameters |
|---|---|
| TCC-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 results in increased use of bandwidth and device resources.<br>The value of 1350 is a recommended 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 Basic Network, Security Zone, and Address Book Information for the Hub | 233
- Configuring IKE for the Hub | 238
- Configuring IPsec for the Hub | 242
- Configuring Security Policies for the Hub | 245
- Configuring TCP-MSS for the Hub | 248
- Configuring Basic Network, Security Zone, and Address Book Information for the Westford Spoke | 249
- Configuring IKE for the Westford Spoke | 254
- Configuring IPsec for the Westford Spoke | 257
- Configuring Security Policies for the Westford Spoke | 260
- Configuring TCP-MSS for the Westford Spoke | 262
- Configuring the Sunnyvale Spoke | 263


### Configuring Basic Network, Security Zone, and Address Book Information for 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-0/0/0 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 10.11.11.11
 set routing-options static route 192.168.178.0/24 next-hop 10.11.11.12
 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 interfaces st0.0
```

```
 set security address-book book1 address local-net 192.168.10.0/24
 set security address-book book1 attach zone trust
 set security address-book book2 address sunnyvale-net 192.168.168.0/24
set security address-book book2 address westford-net 192.168.178.0/24
set security address-book book2 attach zone vpn
```

### 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 basic network, security zone, and address book information for the hub:

1. Configure Ethernet interface information.

```
[edit]
user@hub# set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.1/24
user@hub# set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
user@hub# set interfaces st0 unit 0 family inet address 10.11.11.10/24
```

2. Configure static route information.

```
[edit]
user@hub# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
user@hub# set routing-options static route 192.168.168.0/24 next-hop 10.11.11.11
user@hub# set routing-options static route 192.168.178.0/24 next-hop 10.11.11.12
```

3. Configure the untrust security zone.

```
[edit ]
user@hub# set security zones security-zone untrust
```

4. Assign an interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@hub# set interfaces ge-0/0/3.0
```

5. Specify allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@hub# set host-inbound-traffic system-services ike
```

6. Configure the trust security zone.

```
[edit]
user@hub# edit security zones security-zone trust
```

7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@hub# set interfaces ge-0/0/0.0
```

8. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@hub# set host-inbound-traffic system-services all
```

9. Create an address book and attach a zone to it.

```
[edit security address-book book1]
user@hub# set address local-net 10.10.10.0/24
user@hub# set attach zone trust
```

10. Configure the vpn security zone.

```
[edit]
user@hub# edit security zones security-zone vpn
```

11. Assign an interface to the vpn security zone.

```
[edit security zones security-zone vpn]
user@hub# set interfaces st0.0
```

12. Create another address book and attach a zone to it.

```
[edit security address-book book2]
user@hub# set address sunnyvale-net 192.168.168.0/24
user@hub# set address westford-net 192.168.178.0/24
user@hub# set attach zone vpn
```

### Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, and `show security address-book` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# 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@hub# show routing-options
```

```
static {
    route 0.0.0.0/0 next-hop 10.1.1.1;
    route 192.168.168.0/24 next-hop 10.11.11.11;
    route 192.168.178.0/24 next-hop 10.11.11.12;
}
```

```
[edit]
user@hub# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn {
    host-inbound-traffic {
    }
    interfaces {
        st0.0;
    }
}
[edit]
user@hub# show security address-book
book1 {
    address local-net 10.10.10.0/24;
    attach {
        zone trust;
```

```
    }
}
    book2 {
        address sunnyvale-net 192.168.168.0/24;
        address westford-net 192.168.178.0/24;
        attach {
            zone vpn;
        }
    }
```

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

#### Configuring IKE for 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 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 mode main
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-westford external-interface ge-0/0/3.0
set security ike gateway gw-westford ike-policy ike-phase1-policy
set security ike gateway gw-westford address 10.3.3.2
set security ike gateway gw-sunnyvale external-interface ge-0/0/3.0
set security ike gateway gw-sunnyvale ike-policy ike-phase1-policy
set security ike gateway gw-sunnyvale address 10.2.2.2
```

### 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 IKE for the hub:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
user@hub# set proposal ike-phase1-proposal
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@hub# set policy ike-phase1-policy
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike-phase1-policy]
user@hub# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@hub# set proposals ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike-phase1-policy]
user@hub# set pre-shared-key ascii-text “$ABC123”
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@hub# set gateway gw-westford external-interface ge-0/0/3.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike]
user@hub# set gateway gw-westford ike-policy ike-phase1-policy
```

12. Define the IKE Phase 1 gateway address.

```
[edit security ike]
user@hub# set  gateway gw-westford address 10.3.3.2
```

13. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@hub# set gateway gw-sunnyvale external-interface ge-0/0/3.0
```

14. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway]
user@hub# set gateway gw-sunnyvale ike-policy ike-phase1-policy
```

15. Define the IKE Phase 1 gateway address.

```
[edit security ike gateway]
user@hub# set gateway gw-sunnyvale address 10.2.2.2
```

### 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@hub# 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 {
    mode main;
    proposals ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway gw-sunnyvale {
    ike-policy ike-phase1-policy;
    address 10.2.2.2;
    external-interface ge-0/0/3.0;
}
gateway gw-westford {
    ike-policy ike-phase1-policy;
    address 10.3.3.2;
    external-interface ge-0/0/3.0;
}
```

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

### Configuring IPsec for 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 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 vpn-westford ike gateway gw-westford
set security ipsec vpn vpn-westford ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-westford bind-interface st0.0
set security ipsec vpn vpn-sunnyvale ike gateway gw-sunnyvale
set security ipsec vpn vpn-sunnyvale ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-sunnyvale bind-interface st0.0
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.11 ipsec-vpn vpn-sunnyvale
set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.12 ipsec-vpn vpn-westford
```

## 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 for the hub:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@hub# set security ipsec proposal ipsec-phase2-proposal
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@hub# set policy ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec-phase2-policy]
user@hub# 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 gateways.

```
[edit security ipsec]
user@hub# set vpn vpn-westford ike gateway gw-westford
user@hub# set vpn vpn-sunnyvale ike gateway gw-sunnyvale
```

9. Specify the IPsec Phase 2 policies.

```
[edit security ipsec]
user@hub# set vpn vpn-westford ike ipsec-policy ipsec-phase2-policy
user@hub# set vpn vpn-sunnyvale ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@hub# set vpn vpn-westford bind-interface st0.0
user@hub# set vpn vpn-sunnyvale bind-interface st0.0
```

11. Configure the st0 interface as multipoint.

```
[edit]
user@hub# set interfaces st0 unit 0 multipoint
```

12. Add static NHTB table entries for the Sunnyvale and Westford offices.

```
[edit]
user@hub# set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.11 ipsec-vpn vpn-
sunnyvale
user@hub# set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.12 ipsec-vpn vpn-
westford
```

### 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@hub# 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 vpn-sunnyvale {
    bind-interface st0.0;
    ike {
        gateway gw-sunnyvale;
        ipsec-policy ipsec-phase2-policy;
    }
}
vpn vpn-westford {
    bind-interface st0.0;
    ike {
        gateway gw-westford;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

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

#### Configuring Security Policies for 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 policies from-zone trust to-zone vpn policy local-to-spokes match source-address
local-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match destination-
address sunnyvale-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match destination-
address westford-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match application any
set security policies from-zone trust to-zone vpn policy local-to-spokes then permit
set security policies from-zone vpn to-zone trust policy spokes-to-local match source-address
sunnyvale-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match source-address
```

```
westford-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match destination-
address local-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match application any
set security policies from-zone vpn to-zone trust policy spokes-to-local then permit
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match source-address any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match destination-address
any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match application any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke 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 for the hub:

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@hub# set policy local-to-spokes match source-address local-net
user@hub# set policy local-to-spokes match destination-address sunnyvale-net
user@hub# set policy local-to-spokes match destination-address westford-net
user@hub# set policy local-to-spokes match application any
user@hub# set policy local-to-spokes 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@hub# set policy spokes-to-local match source-address sunnyvale-net
user@hub# set policy spokes-to-local match source-address westford-net
user@hub# set policy spokes-to-local match destination-address local-net
user@hub# set policy spokes-to-local match application any
user@hub# set policy spokes-to-local then permit
```

3. Create the security policy to permit intrazone traffic.

```
[edit security policies from-zone vpn to-zone vpn]
user@hub# set policy spoke-to-spoke match source-address any
```

```
user@hub# set policy spoke-to-spoke match destination-address any
user@hub# set policy spoke-to-spoke match application any
user@hub# set policy spoke-to-spoke 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@hub# show security policies
from-zone trust to-zone vpn {
    policy local-to-spokes {
        match {
            source-address local-net;
            destination-address [ sunnyvale-net westford-net ];
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy spokes-to-local {
        match {
            source-address [ sunnyvale-net westford-net ];
            destination-address local-net;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone vpn {
    policy spoke-to-spoke {
        match {
            source-address any;
            destination-address any;
```

```
            application any;
        }
        then {
            permit;
        }
    }
}
```

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

#### Configuring TCP-MSS for 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 flow tcp-mss ipsec-vpn mss 1350
```

### Step-by-Step Procedure

To configure TCP-MSS information for the hub:

1. Configure TCP-MSS information.

```
[edit]
user@hub# 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@hub# show security flow
tcp-mss {
    ipsec-vpn {
```

```
        mss 1350;
    }
}
```

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

#### Configuring Basic Network, Security Zone, and Address Book Information for the Westford 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-0/0/0 unit 0 family inet address 10.3.3.2/30
set interfaces ge-0/0/3 unit 0 family inet address 192.168.178.1/24
set interfaces st0 unit 0 family inet address 10.11.11.12/24
set routing-options static route 0.0.0.0/0 next-hop 10.3.3.1
set routing-options static route 10.10.10.0/24 next-hop 10.11.11.10
set routing-options static route 192.168.168.0/24 next-hop 10.11.11.10
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/3.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone vpn interfaces st0.0
set security address-book book1 address local-net 192.168.178.0/24
set security address-book book1 attach zone trust
set security address-book book2 address corp-net 10.10.10.0/24
set security address-book book2 address sunnyvale-net 192.168.168.0/24
set security address-book book2 attach zone vpn
```

### 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 basic network, security zone, and address book information for the Westford spoke:

1. Configure Ethernet interface information.

```
[edit]
user@spoke# set interfaces ge-0/0/0 unit 0 family inet address 10.3.3.2/30
user@spoke# set interfaces ge-0/0/3 unit 0 family inet address 192.168.178.1/24
user@spoke# set interfaces st0 unit 0 family inet address 10.11.11.12/24
```

2. Configure static route information.

```
[edit]
user@spoke# set routing-options static route 0.0.0.0/0 next-hop 10.3.3.1
user@spoke# set routing-options static route 10.10.10.0/24 next-hop 10.11.11.10
user@spoke# set routing-options static route 192.168.168.0/24 next-hop 10.11.11.10
```

3. Configure the untrust security zone.

```
[edit]
user@spoke# set security zones security-zone untrust
```

4. Assign an interface to the security zone.

```
[edit security zones security-zone untrust]
user@spoke# set interfaces ge-0/0/0.0
```

5. Specify allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@spoke# set host-inbound-traffic system-services ike
```

6. Configure the trust security zone.

```
[edit]
user@spoke# edit security zones security-zone trust
```

7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@spoke# set interfaces ge-0/0/3.0
```

8. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@spoke# set host-inbound-traffic system-services all
```

9. Configure the vpn security zone.

```
[edit]
user@spoke# edit security zones security-zone vpn
```

10. Assign an interface to the vpn security zone.

```
[edit security zones security-zone vpn]
user@spoke# set interfaces st0.0
```

11. Create an address book and attach a zone to it.

```
[edit security address-book book1]
user@spoke# set address local-net 192.168.178.0/24
user@spoke# set attach zone trust
```

12. Create another address book and attach a zone to it.

```
[edit security address-book book2]
user@spoke# set address corp-net 10.10.10.0/24
user@spoke# set address sunnyvale-net 192.168.168.0/24
user@spoke# set attach zone vpn
```

### Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, **show security zones**, and **show security address-book** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.3.3.2/30;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.178.1/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 10.11.11.10/24;
        }
    }
}
```

```
[edit]
user@spoke# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.3.3.1;
    route 192.168.168.0/24 next-hop 10.11.11.10;
```

```
    route 10.10.10.0/24 next-hop 10.11.11.10;
}
```

```
[edit]
user@spoke# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone vpn {
    interfaces {
        st0.0;
    }
}
[edit]
user@spoke# show security address-book
book1 {
    address corp-net 10.10.10.0/24;
    attach {
        zone trust;
    }
}
    book2 {
        address local-net 192.168.178.0/24;
        address sunnyvale-net 192.168.168.0/24;
```

```
        attach {
            zone vpn;
        }
    }
```

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

### Configuring IKE for the Westford 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 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 mode main
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-corporate external-interface ge-0/0/0.0
set security ike gateway gw-corporate ike-policy ike-phase1-policy
set security ike gateway gw-corporate address 10.1.1.2
```

## 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 IKE for the Westford spoke:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
user@spoke# set proposal ike-phase1-proposal
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@spoke# set policy ike-phase1-policy
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike-phase1-policy]
user@spoke# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@spoke# set proposals ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike-phase1-policy]
user@spoke# set pre-shared-key ascii-text “$ABC123”
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@spoke# set gateway gw-corporate external-interface ge-0/0/0.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike]
user@spoke# set gateway gw-corporate ike-policy ike-phase1-policy
```

12. Define the IKE Phase 1 gateway address.

```
[edit security ike]
user@spoke# set gateway gw-corporate address 10.1.1.2
```

### 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@spoke# 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 {
    mode main;
    proposals ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
```

```
}
gateway gw-corporate {
    ike-policy ike-phase1-policy;
    address 10.1.1.2;
    external-interface ge-0/0/0.0;
}
```

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

#### Configuring IPsec for the Westford 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 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 vpn-corporate ike gateway gw-corporate
set security ipsec vpn vpn-corporate ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-corporate 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 for the Westford spoke:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@spoke# set security ipsec proposal ipsec-phase2-proposal
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@spoke# set policy ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec-phase2-policy]
user@spoke# 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@spoke# set vpn vpn-corporate ike gateway gw-corporate
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate 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@spoke# 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 vpn-corporate {
    bind-interface st0.0;
    ike {
        gateway gw-corporate;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

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

### Configuring Security Policies for the Westford 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 security policies from-zone trust to-zone vpn policy to-corporate match source-address local-net
set security policies from-zone trust to-zone vpn policy to-corporate match destination-address corp-net
set security policies from-zone trust to-zone vpn policy to-corporate match destination-address sunnyvale-net
set security policies from-zone trust to-zone vpn policy to-corporate application any
set security policies from-zone trust to-zone vpn policy to-corporate then permit
set security policies from-zone vpn to-zone trust policy from-corporate match source-address corp-net
set security policies from-zone vpn to-zone trust policy from-corporate match source-address sunnyvale-net
set security policies from-zone vpn to-zone trust policy from-corporate match destination-address local-net
set security policies from-zone vpn to-zone trust policy from-corporate application any
set security policies from-zone vpn to-zone trust policy from-corporate 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 for the Westford spoke:

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@spoke# set policy to-corp match source-address local-net
user@spoke# set policy to-corp match destination-address corp-net
user@spoke# set policy to-corp match destination-address sunnyvale-net
user@spoke# set policy to-corp match application any
user@spoke# set policy to-corp 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@spoke# set policy spokes-to-local match source-address corp-net
user@spoke# set policy spokes-to-local match source-address sunnyvale-net
user@spoke# set policy spokes-to-local match destination-address local-net
user@spoke# set policy spokes-to-local match application any
user@spoke# set policy spokes-to-local 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@spoke# show security policies
from-zone trust to-zone vpn {
    policy to-corp {
        match {
            source-address local-net;
            destination-address [ sunnyvale-net westford-net ];
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy spokes-to-local {
        match {
            source-address [ sunnyvale-net westford-net ];
            destination-address local-net;
            application any;
        }
        then {
            permit;
        }
```

```
    }
}
```

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

#### Configuring TCP-MSS for the Westford 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 security flow tcp-mss ipsec-vpn mss 1350
```

### Step-by-Step Procedure

To configure TCP-MSS for the Westford spoke:

1. Configure TCP-MSS information.

```
[edit]
user@spoke# 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@spoke# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
    }
}
```

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

### Configuring the Sunnyvale Spoke

## CLI Quick Configuration

This example uses an SSG Series device for the Sunnyvale spoke. For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts and 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"
set interface ethernet0/6 zone "Trust"
set interface "tunnel.1" zone "VPN"
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" "sunnyvale-net" 192.168.168.0 255.255.255.0
set address "VPN" "corp-net" 10.10.10.0 255.255.255.0
set address "VPN" "westford-net" 192.168.178.0 255.255.255.0
set ike gateway "corp-ike" address 10.1.1.2 Main outgoing-interface ethernet0/0 preshare
"395psksecr3t" sec-level standard
set vpn corp-vpn monitor optimized rekey
set vpn "corp-vpn" bind interface tunnel.1
set vpn "corp-vpn" gateway "corp-ike" replay tunnel idletime 0 sec-level standard
set policy id 1 from "Trust" to "Untrust" "ANY" "ANY" "ANY" nat src permit
set policy id 2 from "Trust" to "VPN" "sunnyvale-net" "corp-net" "ANY" permit
set policy id 2
exit
set dst-address "westford-net"
exit
set policy id 3 from "VPN" to "Trust" "corp-net" "sunnyvale-net" "ANY" permit
set policy id 3
set src-address "westford-net"
exit
set route 10.10.10.0/24 interface tunnel.1
```

```
set route 192.168.178.0/24 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 | 264
- Verifying the IPsec Phase 2 Status | 266
- Verifying Next-Hop Tunnel Bindings | 268
- Verifying Static Routes for Remote Peer Local LANs | 269
- Reviewing Statistics and Errors for an IPsec Security Association | 270
- Testing Traffic Flow Across the VPN | 270


To confirm that the configuration is working properly, perform these tasks:

### 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 and 192.168.178/24 networks to bring the tunnels up. For route-based VPNs, you can send traffic initiated from the SRX Series Firewall through the tunnel. We recommend that when testing IPsec tunnels, you send test traffic 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@hub> show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
```

```
6       10.3.3.2         UP     94906ae2263bbd8e  1c35e4c3fc54d6d3  Main
7       10.2.2.2         UP     7e7a1c0367dfe73c  f284221c656a5fbc  Main
```

```
user@hub> show security ike security-associations index 6 detail
IKE peer 10.3.3.2, Index 6,
  Role: Responder, State: UP
  Initiator cookie: 94906ae2263bbd8e,, Responder cookie: 1c35e4c3fc54d6d3
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.1.2:500, Remote: 10.3.3.2:500
  Lifetime: Expires in 3571 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : aes-cbc (128 bits)
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input bytes    :                1128
   Output bytes   :                 988
   Input packets  :                   6
   Output packets :                   5
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 1
    Negotiation type: Quick mode, Role: Responder, Message ID: 1350777248
    Local: 10.1.1.2:500, Remote: 10.3.3.2:500
    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)
    Flags: Caller notification sent, Waiting for done
```

### 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 information is 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 security association 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)
- Initiator and responder role information

  Troubleshooting is best performed on the peer using the responder role.
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

**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@hub> show security ipsec security-associations
  total configured sa: 4
```

```
  ID    Gateway          Port  Algorithm          SPI      Life:sec/kb  Mon vsys
  <16384 10.2.2.2         500   ESP:aes-128/sha1   b2fc36f8 3364/ unlim   -   0
  >16384 10.2.2.2         500   ESP:aes-128/sha1   5d73929e 3364/ unlim   -   0
  ID    Gateway          Port  Algorithm          SPI      Life:sec/kb  Mon vsys
  <16385 10.3.3.2         500   ESP:3des/sha1      70f789c6 28756/unlim   -   0
  >16385 10.3.3.2         500   ESP:3des/sha1      80f4126d 28756/unlim   -   0
```

```
user@hub> show security ipsec security-associations index 16385 detail
  Virtual-system: Root
  Local Gateway: 10.1.1.2, Remote Gateway: 10.3.3.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/24)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    DF-bit: clear
    Direction: inbound, SPI: 1895270854, AUX-SPI: 0
    Hard lifetime: Expires in 28729 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 28136 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: 2163479149, AUX-SPI: 0
    Hard lifetime: Expires in 28729 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 28136 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 16385. 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 28756/ unlim value indicates that the Phase 2 lifetime expires in 28756 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 16385 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 Next-Hop Tunnel Bindings

### Purpose

After Phase 2 is complete for all peers, verify the next-hop tunnel bindings.

### Action

From operational mode, enter the `show security ipsec next-hop-tunnels` command.

```
user@hub> show security ipsec next-hop-tunnels
Next-hop gateway  interface    IPSec VPN name               Flag
10.11.11.11       st0.0        sunnyvale-vpn                Static
10.11.11.12       st0.0        westford-vpn                 Auto
```

### Meaning

The next-hop gateways are the IP addresses for the st0 interfaces of all remote spoke peers. The next hop should be associated with the correct IPsec VPN name. If no NHTB entry exists, there is no way for the hub device to differentiate which IPsec VPN is associated with which next hop.

The Flag field has one of the following values:

- Static— NHTB was manually configured in the st0.0 interface configurations, which is required if the peer is not an SRX Series Firewall.
- Auto— NHTB was not configured, but the entry was automatically populated into the NHTB table during Phase 2 negotiations between two SRX Series Firewalls

There is no NHTB table for any of the spoke sites in this example. From the spoke perspective, the st0 interface is still a point-to-point link with only one IPsec VPN binding.

#### Verifying Static Routes for Remote Peer Local LANs

### Purpose

Verify that the static route references the spoke peer's st0 IP address.

### Action

From operational mode, enter the `show route` command.

```
user@hub> show route 192.168.168.10
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.168.0/24   *[Static/5] 00:08:33
                    > to 10.11.11.11 via st0.0
```

```
user@hub> show route 192.168.178.10
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.178.0/24   *[Static/5] 00:04:04
                    > to 10.11.11.12 via st0.0
```

The next hop is the remote peer's st0 IP address, and both routes point to st0.0 as the outgoing interface.

#### Reviewing 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` command.

```
user@hub> show security ipsec statistics index 16385
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 whether 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.

```
user@hub> 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@hub> 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
```

```
ssg-> ping 192.168.178.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.178.10, timeout is 1 seconds from
ethernet0/6
!!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=8/8/10 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

- Understanding Hub-and-Spoke VPNs | **198**
- Route-Based IPsec VPNs | **406**
- Example: Configuring a Policy-Based VPN | **280**

**Release History Table**

| 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 | **433**

# Comparing Policy-Based and Route-Based VPNs

It is important to understand the differences between policy-based and route-based VPNs and why one might be preferable to the other.

Table 26 on page 273 lists the differences between route-based VPNs and policy-based VPNs.

**Table 26: Differences Between Route-Based VPNs and Policy-Based VPNs**

| Route-Based VPNs | Policy-Based VPNs |
|---|---|
| With route-based VPNs, a policy does not specifically reference a VPN tunnel. | With policy-based VPN tunnels, a tunnel is treated as an object that, together with source, destination, application, and action, constitutes a tunnel policy that permits VPN traffic. |
| The policy references a destination address. | In a policy-based VPN configuration, a tunnel policy specifically references a VPN tunnel by name. |
| The number of route-based VPN tunnels that you create is limited by the number of route entries or the number of st0 interfaces that the device supports, whichever number is lower. | The number of policy-based VPN tunnels that you can create is limited by the number of policies that the device supports. |
| Route-based VPN tunnel configuration is a good choice when you want to conserve tunnel resources while setting granular restrictions on VPN traffic. | 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 security association (SA) with the remote peer. Each SA counts as an individual VPN tunnel. |
| With a route-based approach to VPNs, the regulation of traffic is not coupled to the means of its delivery. You can configure dozens of policies to regulate traffic flowing through a single VPN tunnel between two sites, and only one IPsec SA is at work. Also, a route-based VPN configuration allows you to create policies referencing a destination reached through a VPN tunnel in which the action is deny. | In a policy-based VPN configuration, the action must be permit and must include a tunnel. |
| 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. | The exchange of dynamic routing information is not supported in policy-based VPNs. |

**Table 26: Differences Between Route-Based VPNs and Policy-Based VPNs *(Continued)***

| Route-Based VPNs | Policy-Based VPNs |
|---|---|
| Route-based configurations are used for hub-and-spoke topologies. | Policy-based VPNs cannot be used for hub-and-spoke topologies. |
| With route-based VPNs, a policy does not specifically reference a VPN tunnel. | 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, a policy-based tunnel is the best choice. |
| Route-based VPNs do not support remote-access (dial-up) VPN configurations. | Policy-based VPN tunnels are required for remote-access (dial-up) VPN configurations. |
| Route-based VPNs might not work correctly with some third-party vendors. | Policy-based VPNs might be required if the third party requires separate SAs for each remote subnet. |
| 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 via a secure tunnel interface (`st0`) , which is bound to a specific VPN tunnel.<br><br>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. | With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy. |
| Route-based VPNs support NAT for st0 interfaces. | Policy-based VPNs cannot be used if NAT is required for tunneled 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.

Policy-based VPNs are only supported on SRX5400, SRX5600, and SRX5800 line. Platform support depends on the Junos OS release in your installation.

RELATED DOCUMENTATION

Example: Configuring a Route-Based VPN | 407

Example: Configuring a Policy-Based VPN | 280

# Chassis Cluster HA 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 SRX Series Chassis Cluster Configuration Overview.

1. After enabling the chassis cluster, in the device 1, configure HA link encryption as shown in sample configuration below, commit and reboot. Device 1 needs to be configured with both node0 and node1 HA link encryption configuration 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.

6
CHAPTER

# Policy Based VPN

Policy-Based IPsec VPNs | 279

Configure Policy-Based IPsec VPN with Certificates | 305

Configure IPsec VPN with OCSP for Certificate Revocation Status | 340

IPv6 IPsec VPNs | 361

# Policy-Based IPsec VPNs

IN THIS SECTION


- Understanding Policy-Based IPsec VPNs | 279
- Example: Configuring a Policy-Based VPN | 280


A policy-based VPN is a configuration in which an IPsec VPN tunnel created between two end points is specified within the policy itself with a policy action for the transit traffic that meets the policy's match criteria.

## Understanding Policy-Based IPsec VPNs

For policy-based IPsec VPNs, a security policy specifies as its action the VPN tunnel to be used for transit traffic that meets the policy's match criteria. A VPN is configured independent of a policy statement. The policy statement refers to the VPN by name to specify the traffic that is allowed access to the tunnel. For policy-based VPNs, each policy creates an individual IPsec security association (SA) with the remote peer, each of which counts as an individual VPN tunnel. For example, if a policy contains a group source address and a group destination address, whenever one of the users belonging to the address set attempts to communicate with any one of the hosts specified as the destination address, a new tunnel is negotiated and established. Because each tunnel requires its own negotiation process and separate pair of SAs, the use of policy-based IPsec VPNs can be more resource-intensive than route-based VPNs.

Examples of where policy-based VPNs can be used:

- You are implementing a dial-up VPN.
- Policy-based VPNs allow you to direct traffic based on firewall policies.

We recommend that you use route-based VPN when you want to configure a VPN between multiple remote sites. Route-based VPNs can provide the same capabilities as policy-based VPNs.

Limitations:

- Policy-based IPSec VPNs are not supported with IKEv2.

- Support for policy-based IPsec VPN is not available when using `junos-ike` package with your firewall running `iked` process for IPsec VPN service. With `junos-ike` package, remove any policy-based IPsec VPN configurations as they are ineffective. Note that in SRX5K-SPC3 with RE3, the `junos-ike` package is available by default. In platforms SRX1500 and higher, it's an optional package. See IPsec VPN Feature Support with New Package for more details.

**SEE ALSO**

IPsec Overview | 20

Example: Configuring a Route-Based VPN | 407

Example: Configuring a Hub-and-Spoke VPN | 221

## Example: Configuring a Policy-Based VPN

**IN THIS SECTION**

- Requirements | 280
- Overview | 281
- Configuration | 284
- Verification | 298

This example shows how to configure a policy-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 Policy-Based demonstration in Juniper Networks Virtual Labs and

reserve your free sandbox today! You'll find the IPsec VPN Policy-Based sandbox in the Security category.

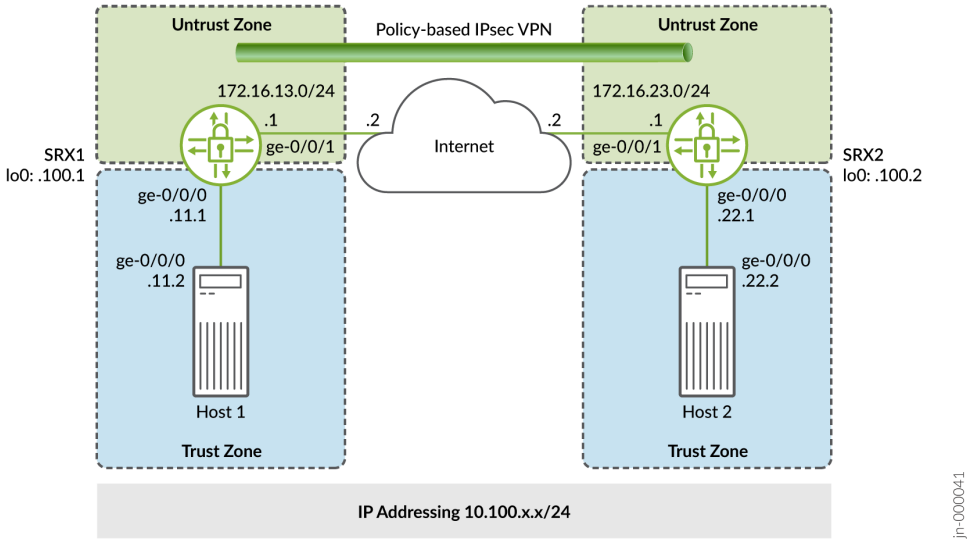
Before you begin, read "IPsec Overview" on page 20.

## Overview

In this example, you configure a policy-based VPN on SRX1 and SRX2. Host1 and Host2 use the VPN to send traffic securely over the Internet between both hosts.

Figure 26 on page 281 shows an example of a policy-based VPN topology.

**Figure 26: Policy-Based VPN Topology**




IKE IPsec tunnel 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. Just as there are two phases to tunnel negotiation, there are two phases to tunnel configuration.

In this example, you configure interfaces, an IPv4 default route, and security zones. Then you configure IKE Phase 1, IPsec Phase 2, security policy, and TCP-MSS parameters. See Table 27 on page 282 through Table 31 on page 284.

**Table 27: Interface, Static Route, and Security Zone Information for SRX1**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Interfaces | ge-0/0/0.0 | 10.100.11.1/24 |
| | ge-0/0/1.0 | 172.16.13.1/24 |
| 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. |
| Static routes | 0.0.0.0/0 | • The next hop is 172.16.13.2. |

**Table 28: IKE Phase 1 Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | standard | • Authentication method: pre-shared-keys |
| Policy | IKE-POL | • Mode: main<br>• Proposal reference: standard<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |
| Gateway | IKE-GW | • IKE policy reference: IKE-POL<br>• External interface: ge-0/0/1<br>• Gateway address: 172.16.23.1 |

**Table 29: IPsec Phase 2 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<br>• IPsec policy reference: IPSEC-POL<br>• establish-tunnels immediately |

**Table 30: Security Policy Configuration Parameters**

| Purpose | Name | Configuration Parameters |
|---|---|---|
| This security policy permits traffic from the trust zone to the untrust zone. | VPN-OUT | • Match criteria:<br>  • source-address Host1-Net<br>  • destination-address Host2-Net<br>  • application any<br>• Permit action: tunnel ipsec-vpn VPN-to-Host2 |
| This security policy permits traffic from the untrust zone to the trust zone. | VPN-IN | • Match criteria:<br>  • source-address Host2-Net<br>  • destination-address Host1-Net<br>  • application any<br>• Permit action: tunnel ipsec-vpn VPN-to-Host2 |

**Table 30: Security Policy Configuration Parameters *(Continued)***

| Purpose | Name | Configuration Parameters |
|---|---|---|
| This security policy permits all traffic from the trust zone to the untrust zone.<br><br>You must put the VPN-OUT policy before the default-permit security policy. Junos OS performs a security policy lookup starting at the top of the list. If the default-permit policy comes before the VPN-OUT policy, all traffic from the trust zone matches the default-permit policy and is permitted. Thus, no traffic will ever match the VPN-OUT policy. | default-permit | • Match criteria:<br>  • source-address any<br>  • source-destination any<br>  • application any<br>• Action: permit |

**Table 31: 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 maximum transmission unit (MTU) limits on a network. This is especially important for VPN traffic, as the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting Encapsulating Security Payload (ESP) packet to exceed the MTU of the physical interface, thus causing fragmentation. Fragmentation results in increased use of bandwidth and device resources.<br><br>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 Basic Network and Security Zone Information | **285**
- Configuring IKE | **288**
- Configuring IPsec | **290**

- Configuring Security Policies | **292**
- Configuring TCP-MSS | **295**
- Configuring SRX2 | **296**


### Configuring Basic Network and Security Zone Information

#### CLI Quick Configuration

To quickly configure this 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 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
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do this, 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
```

2. Configure the static routes.

```
[edit]
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
```

### 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;
        }
    }
}
```

```
[edit]
user@SRX1# show routing-options
static {
    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;
    }
}
```

### Configuring IKE

## CLI Quick Configuration

To quickly configure this 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 the 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 the 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 the IKE gateway and define its external interface.

```
[edit security ike gateway IKE-GW]
user@SRX1# set external-interface ge-0/0/1.0
```

8. Define the IKE gateway address.

```
[edit security ike gateway IKE-GW]
user@SRX1# address 172.16.23.1
```

9. Define the IKE policy reference.

```
[edit security ike gateway IKE-GW]
user@SRX1# set ike-policy IKE-POL
```

### 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 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 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 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 the 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@SRX1# set vpn VPN-to-Host2 ike ipsec-policy IPSEC-POL
```

6. Configure the tunnel to establish immediately.

```
[edit security ipsec]
user@SRX1# 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@SRX1# show security ipsec
proposal standard;
policy IPSEC-POL {
    proposals standard;
}
vpn VPN-to-Host2 {
    ike {
        gateway IKE-GW;
        ipsec-policy IPSEC-POL;
    }
    establish-tunnels immediately;
}
```



### CLI Quick Configuration

To quickly configure this 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 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 untrust
```

```
set security policies from-zone trust to-zone untrust policy VPN-OUT match source-address Host1-
Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match destination-address
Host2-Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match application any
set security policies from-zone trust to-zone untrust policy VPN-OUT then permit tunnel ipsec-
vpn VPN-to-Host2
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 untrust to-zone trust policy VPN-IN match source-address Host2-
Net
set security policies from-zone untrust to-zone trust policy VPN-IN match destination-address
Host1-Net
set security policies from-zone untrust to-zone trust policy VPN-IN match application any
set security policies from-zone untrust to-zone trust policy VPN-IN then permit tunnel ipsec-vpn
VPN-to-Host2
```

### 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 untrust
```

2. Create the security policy to match on traffic from Host1 in the trust zone to Host2 in the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
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 tunnel ipsec-vpn VPN-to-Host2
```

3. Create the security policy to permit all other traffic to the Internet from the trust zone to the untrust zone.

```
[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
```

4. Create a security policy to permit traffic from Host2 in the untrust zone to Host1 in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@SRX1# set policy VPN-IN match source-address Host2-Net
user@SRX1# set policy VPN-IN match destination-address Host1-Net
user@SRX1# set policy VPN-IN match application any
user@SRX1# set policy VPN-IN then permit tunnel ipsec-vpn VPN-to-Host2
```

### 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@SRX1# show security policies
from-zone trust to-zone untrust {
    policy VPN-OUT {
        match {
            source-address Host1-Net;
            destination-address Host2-Net;
            application any;
        }
        then {
            permit {
                tunnel {
```

```
                    ipsec-vpn VPN-to-Host2;
                }
            }
        }
    }
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone untrust to-zone trust {
    policy VPN-IN {
        match {
            source-address Host2-Net;
            destination-address Host1-Net;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn VPN-to-Host2;
                }
            }
        }
    }
}
```

**Configuring TCP-MSS**

## CLI Quick Configuration

To quickly configure this 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 flow tcp-mss ipsec-vpn mss 1350
```

### Step-by-Step Procedure

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 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 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 untrust
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 VPN-OUT match source-address Host2-
Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match destination-address
Host1-Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match application any
set security policies from-zone trust to-zone untrust policy VPN-OUT then permit tunnel ipsec-
vpn VPN-to-Host1
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 untrust to-zone trust policy VPN-IN match source-address Host1-
Net
set security policies from-zone untrust to-zone trust policy VPN-IN match destination-address
Host2-Net
set security policies from-zone untrust to-zone trust policy VPN-IN match application any
set security policies from-zone untrust to-zone trust policy VPN-IN then permit tunnel ipsec-vpn
VPN-to-Host1
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 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 routing-options static route 0.0.0.0/0 next-hop 172.16.23.2
```

## Verification

**IN THIS SECTION**


- Verifying the IKE Status | **298**
- Verifying the IPsec Phase 2 Status | **301**
- Test Traffic Flow Across the VPN | **303**
- Reviewing Statistics and Errors for an IPsec Security Association | **303**


To confirm that the configuration is working properly, perform these tasks:

### Verifying 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
1859361 UP     9788fa59c3ee2e2a  0b17e52f34b83aba  Main           172.16.23.1
```

```
user@SRX1> show security ike security-associations index 1859361 detail
IKE peer 172.16.23.1, Index 1859361, Gateway Name: IKE-GW
  Role: Responder, State: UP
  Initiator cookie: 9788fa59c3ee2e2a, Responder cookie: 0b17e52f34b83aba
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 172.16.13.1:500, Remote: 172.16.23.1:500
  Lifetime: Expires in 17567 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  :                 1740
   Output bytes  :                 1132
   Input  packets:                   15
   Output packets:                    7
   Input  fragmentated packets:       0
   Output fragmentated packets:       0
  IPSec security associations: 4 created, 4 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 Phase 1 security associations (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 1859361 detail` command lists additional information about the security association with an index number of 1859361:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Initiator and responder role information

  Troubleshooting is best performed on the peer using the responder role.
- Number of IPsec SAs created

- Number of Phase 2 negotiations in progress

**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@SRX1 show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <2      ESP:3des/sha1   ae5afc5a 921/ unlim   -   root 500   172.16.23.1
  >2      ESP:3des/sha1   6388a743 921/ unlim   -   root 500   172.16.23.1
```

```
user@SRX1> show security ipsec security-associations index 2 detail
ID: 2 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]=10.100.11.0/24)
  Remote Identity: ipv4_subnet(any:0,[0..7]=10.100.22.0/24)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled                              , Policy-name: VPN-OUT
  Port: 500, Nego#: 30, Fail#: 0, Def-Del#: 0 Flag: 0x600829
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Thu Jul 29 2021 14:29:22 -0700: IPSec SA negotiation successfully completed (29 times)
    Thu Jul 29 2021 12:00:30 -0700: IKE SA negotiation successfully completed (4 times)
    Wed Jul 28 2021 15:20:58
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Wed Jul 28 2021 15:05:13 -0700: IPSec SA negotiation successfully completed (1 times)
    Wed Jul 28 2021 15:05:13
    : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
    Wed Jul 28 2021 15:05:13 -0700: External interface's address received. Information updated
(1 times)
    Wed Jul 28 2021 15:05:13 -0700: External interface's zone received. Information updated (1
```

```
times)
    Wed Jul 28 2021 11:17:38
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (1 times)
    Wed Jul 28 2021 09:27:11 -0700: IKE SA negotiation successfully completed (19 times)
    Thu Jul 22 2021 16:34:17 -0700: Negotiation failed with INVALID_SYNTAX error (3 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: ae5afc5a, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 828 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 234 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: 6388a743, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 828 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 234 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 2. 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 921/ unlim value indicates that the Phase 2 lifetime expires in 921 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 (up) or D (down) is listed.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 2 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 reasons for a Phase 2 failure. For policy-based VPNs, the proxy ID is derived from the security policy. The local address and remote address are derived from the address book entries, and the service is derived from the application configured for the policy. If Phase 2 fails because of a proxy ID mismatch, you can use the policy to confirm which address book entries are configured. Verify that the addresses match the information being sent. Check the service to ensure that the ports match the information being sent.

#### 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
!!!
```

### 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 2
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, you can run the `show security ipsec statistics` command several times to confirm that the encrypted and decrypted packet counters are incrementing. You should also check if the other error counters are incrementing.

### SEE ALSO


- IPsec Overview | **20**
- Example: Configuring a Route-Based VPN | **407**
- IPsec Policy-Based VPN in Juniper Networks Virtual Labs


### RELATED DOCUMENTATION


- AutoVPN on Hub-and-Spoke Devices | **1033**

# Configure Policy-Based IPsec VPN with Certificates

IN THIS SECTION


- Requirements | 305
- Overview | 306
- Configuration | 309
- Verification | 321
- Troubleshooting IKE, PKI, and IPsec Issues | 328


This example shows how to configure, verify, and troubleshoot PKI. This topic includes the following sections:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.4 or later
- Juniper Networks security devices

Before you begin:

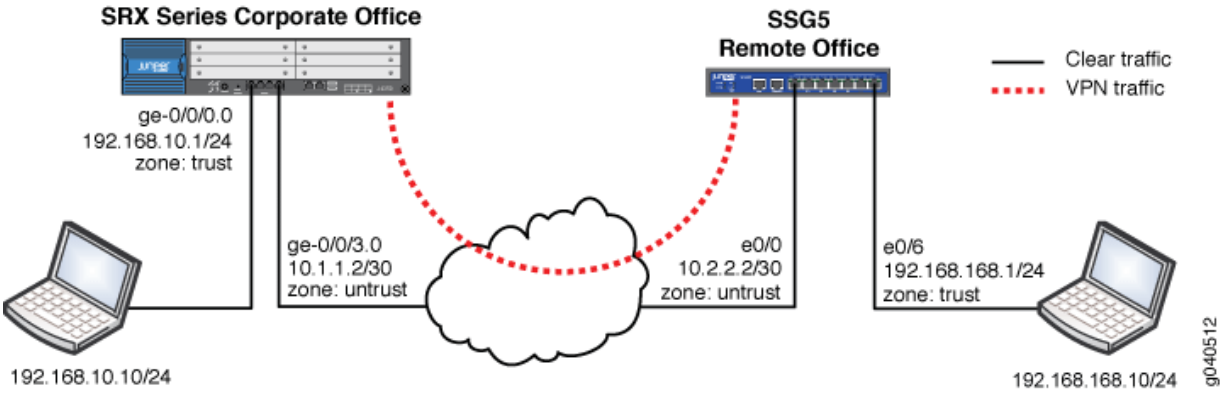
- Ensure that the internal LAN interface of the SRX Series Firewall is ge-0/0/0 in zone trust and has a private IP subnet.
- Ensure that the Internet interface of the device is ge-0/0/3 in zone untrust and has a public IP.
- Ensure that all traffic between the local and remote LANs is permitted, and traffic can be initiated from either side.
- Ensure that the SSG5 has been preconfigured correctly and loaded with a ready-to-use local certificate, CA certificate, and CRL.
- Ensure that the SSG5 device is configured to use the FQDN of ssg5.example.net (IKE ID).
- Ensure that PKI certificates with 1024-bit keys are used for the IKE negotiations on both sides.

- Ensure that the CA is a standalone CA at the domain example.com for both VPN peers.

## Overview

Figure 27 on page 306 shows the network topology used for this example to configure a policy-based IPsec VPN to allow data to be securely transferred between a corporate office and a remote office.

**Figure 27: Network Topology Diagram**




The PKI administration is the same for both policy-based VPNs and route-based VPNs.

In this example, the VPN traffic is incoming on interface ge-0/0/0.0 with the next hop of 10.1.1.1. Thus the traffic is outgoing on interface ge-0/0/3.0. Any tunnel policy must consider incoming and outgoing interfaces.

Optionally, you can use a dynamic routing protocol such as OSPF (not described in this document). When processing the first packet of a new session, the device running Junos OS first performs a route lookup. The static route, which is also the default route, dictates the zone for the outgoing VPN traffic.

Many CAs use hostnames (for example, FQDN) to specify various elements of the PKI. Because the CDP is usually specified using a URL containing an FQDN, you must configure a DNS resolver on the device running Junos OS.

The certificate request can be generated by the following methods:

- Creating a CA profile to specify the CA settings
- Generating the PKCS10 certificate request

The PKCS10 certificate request process involves generating a public or private key pair and then generating the certificate request itself, using the key pair.

Take note of the following information about the CA profile:

- The CA profile defines the attributes of a certificate authority.
- Each CA profile is associated with a CA certificate. If a new or renewed CA certificate needs to be loaded without removing the older CA certificate, a new profile must be created. This profile can also be used for online fetching of the CRL.
- There can be multiple such profiles present in the system created for different users.

If you specify a CA administrator e-mail address to send the certificate request to, then the system composes an e-mail from the certificate request file and forwards it to the specified e-mail address. The e-mail status notification is sent to the administrator.

The certificate request can be sent to the CA through an out-of-band method.

The following options are available to generate the PKCS10 certificate request:

- `certificate-id` — Name of the local digital certificate and the public/private key pair. This ensures that the proper key pair is used for the certificate request and ultimately for the local certificate.

  Starting in Junos OS Release 19.1R1, a commit check is added to prevent user from adding `.`, `/`, `%`, and space in a certificate identifier while generating a local or remote certificates or a key pair.

- `subject` — Distinguished name format that contains the common name, department, company name, state, and country:
  - CN — Common name
  - OU — Department
  - O — Company name
  - L — Locality
  - ST — State
  - C — Country
  - CN — Phone
  - DC — Domain component

    You are not required to enter all subject name components. Note also that you can enter multiple values of each type.

- `domain-name` — FQDN. The FQDN provides the identity of the certificate owner for IKE negotiations and provides an alternative to the subject name.

- `filename (path | terminal)` — (Optional) Location where the certificate request should be placed, or the login terminal.
- `ip-address` — (Optional) IP address of the device.
- `email` — (Optional) E-mail address of the CA administrator.

  You must use a domain-name, an ip-address, or an e-mail address.

The generated certificate request is stored in a specified file location. A local copy of the certificate request is saved in the local certificate storage. If the administrator reissues this command, the certificate request is generated again.

The PKCS10 certificate request is stored in a specified file and location, from which you can download it and send it to the CA for enrollment. If you have not specified the filename or location, you can get PKCS10 certificate request details by using the `show security pki certificate-request certificate-id <id-name>` command in the CLI. You can copy the command output and paste it into a Web front end for the CA server or into an e-mail.

The PKCS10 certificate request is generated and stored on the system as a pending certificate or certificate request. An e-mail notification is sent to the administrator of the CA (in this example, certadmin@example.com).

A unique identity called certificate-ID is used to name the generated key pair. This ID is also used in certificate enrollment and request commands to get the right key pair. The generated key pair is saved in the certificate store in a file with the same name as the certificate-ID. The file size can be 1024 or 2048 bits.

A default (fallback) profile can be created if intermediate CAs are not preinstalled in the device. The default profile values are used in the absence of a specifically configured CA profile.

In the case of a CDP, the following order is followed:

- Per CA profile
- CDP embedded in CA certificate
- Default CA profile

We recommend using a specific CA profile instead of a default profile.

The administrator submits the certificate request to the CA. The CA administrator verifies the certificate request and generates a new certificate for the device. The administrator for the Juniper Networks device retrieves it, along with the CA certificate and CRL.

The process of retrieving the CA certificate, the device's new local certificate, and the CRL from the CA depends on the CA configuration and software vendor in use.

Junos OS supports the following CA vendors:

- Entrust
- Verisign
- Microsoft

Although other CA software services such as OpenSSL can be used to generate certificates, these certificates are not verified by Junos OS.

## Configuration

**IN THIS SECTION**


- PKI Basic Configuration | 309
- Configuring a CA Profile | 311
- Generating a Public-Private Key Pair | 312
- Enrolling a Local Certificate | 312
- Loading CA and Local Certificates | 313
- Configuring the IPsec VPN with the Certificates | 317


### PKI Basic Configuration

#### 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 PKI:

1. Configure an IP address and protocol family on the Gigabit Ethernet interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 192.168.10.1/24
user@host# set ge-0/0/3 unit 0 family inet address 10.1.1.2/30
```

2. Configure a default route to the Internet next hop.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
```

3. Set the system time and date.

```
[edit]
user@host# set system time-zone PST8PDT
```

After the configuration is committed, verify the clock settings using the `show system uptime` command.

```
user@host> show system uptime
Current time: 2007-11-01 17:57:09 PDT
System booted: 2007-11-01 14:36:38 PDT (03:20:31 ago)
Protocols started: 2007-11-01 14:37:30 PDT (03:19:39 ago)
Last configured: 2007-11-01 17:52:32 PDT (00:04:37 ago) by root
5:57PM up 3:21, 4 users, load averages: 0.00, 0.00, 0.00
```

4. Set the NTP server address.

```
user@host> set date ntp 130.126.24.24
1 Nov 17:52:52 ntpdate[5204]: step time server 172.16.24.24 offset -0.220645 sec
```

5. Set the DNS configuration.

```
[edit]
user@host# set system name-server 172.31.2.1
user@host# set system name-server 172.31.2.2
```

## Configuring a CA Profile

### Step-by-Step Procedure

1. Create a trusted CA profile.

```
[edit]
user@host# set security pki ca-profile ms-ca ca-identity example.com
```

2. Create a revocation check to specify a method for checking certificate revocation.

   Set the refresh interval, in hours, to specify the frequency in which to update the CRL. The default values are next-update time in CRL, or 1 week, if no next-update time is specified.

```
[edit]
user@host# set security pki ca-profile ms-ca revocation-check crl refresh-interval 48
```

   In the `revocation-check` configuration statement, you can use the `disable` option to disable the revocation check or select the `crl` option to configure the CRL attributes. You can select the `disable on-download-failure` option to allow the sessions matching the CA profile, when CRL download failed for a CA profile. The sessions will be allowed only if no old CRL is present in the same CA profile.

3. Specify the location (URL) to retrieve the CRL (HTTP or LDAP). By default, the URL is empty and uses CDP information embedded in the CA certificate.

```
[edit]
user@host# set security pki ca-profile ms-ca revocation-check crl url http://srv1.example.com/
CertEnroll/EXAMPLE.crl
```

   Currently you can configure only one URL. Support for backup URL configuration is not available.

4. Specify an e-mail address to send the certificate request directly to a CA administrator.

```
user@host# set security pki ca-profile ms-ca administrator email-address certadmin@example.com
```

5. Commit the configuration:

```
user@host# commit and-quit
commit complete
Exiting configuration mode
```

## Generating a Public-Private Key Pair

### Step-by-Step Procedure

When the CA profile is configured, the next step is to generate a key pair on the Juniper Networks device. To generate the private and public key pair:

1. Create a certificate key pair.

```
user@host> request security pki generate-key-pair certificate-id ms-cert size 1024
```

### Results

After the public-private key pair is generated, the Juniper Networks device displays the following:

```
Generated key pair ms-cert, key size 1024 bits
```

## Enrolling a Local Certificate

### Step-by-Step Procedure

1. Generate a local digital certificate request in the PKCS-10 format. See *request security pki generate-certificate-request*.

```
user@host> request security pki generate-certificate-request certificate-id ms-cert subject
"CN=john doe,CN=10.1.1.2,OU=sales,O=example, L=Sunnyvale,ST=CA,C=US" email user@example.net
filename ms-cert-req
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIB3DCCAUUCAQAwbDERMA8GA1UEAxMIam9obiBkb2UxDjAMBgNVBAsTBXNhbGVz
MRkwFwYDVQQKExBKdW5pcGVyIE5ldHdvcmtzMRIwEAYDVQQHEwlTdW5ueXZhbGUx
```

```
CzAJBgNVBAgTAkNBMQswCQYDVQQGEwJVUzCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEA5EG6sgG/CTFzX6KC/hz6Czal0BxakUxfGxF7UWYWHaWFFYLqo6vXNO8r
OS5Yak7rWANAsMob3E2X/1adlQIRi4QFTjkBqGI+MTEDGnqFsJBqrB6oyqGtdcSU
u0qUivMvgKQVCx8hpx99J3EBTurfWL1pCNlBmZggNogb6MbwES0CAwEAAaAwMC4G
CSqGSIb3DQEJDjEhMB8wHQYDVR0RBBYwFIESInVzZXJAanVuaXBlci5uZXQiMA0G
CSqGSIb3DQEBBQUAA4GBAI6GhBaCsXk6/1lE2e5AakFFDhY7oqzHhgd1yMjiSUMV
djmf9JbDz2gM2UKpI+yKgtUjyCK/lV2ui57hpZMvnhAW4AmgwkOJg6mpR5rsxdLr
4/HHSHuEGOF17RHO6x0YwJ+KE1rYDRWj3Dtz447ynaLxcDF7buwd4IrMcRJJI9ws
-----END CERTIFICATE REQUEST-----
Fingerprint:
47:b0:e1:4c:be:52:f7:90:c1:56:13:4e:35:52:d8:8a:50:06:e6:c8 (sha1)
a9:a1:cd:f3:0d:06:21:f5:31:b0:6b:a8:65:1b:a9:87 (md5)
```

In the sample of the PKCS10 certificate, the request starts with and includes the BEGIN CERTIFICATE REQUEST line and ends with and includes the END CERTIFICATE REQUEST line. This portion can be copied and pasted to your CA for enrollment. Optionally, you can also offload the ms-cert-req file and send that to your CA.

2. Submit the certificate request to the CA, and retrieve the certificate.

## Loading CA and Local Certificates

### Step-by-Step Procedure

1. Load the local certificate, CA certificate, and CRL.

```
user@host> file copy ftp://192.168.10.10/certnew.cer certnew.cer /var/
tmp//...transferring.file.........crYdEC/100% of 1459 B 5864 kBps
user@host> file copy ftp:// 192.168.10.10/CA-certnew.cer CA-certnew.cer /var/
tmp//...transferring.file.........UKXUWu/100% of 1049 B 3607 kBps
user@host> file copy ftp:// 192.168.10.10/certcrl.crl certcrl.crl /var/
tmp//...transferring.file.........wpqnpA/100% of 401 B 1611 kBps
```

You can verify that all files have been uploaded by using the command `file list`.

2. Load the certificate into local storage from the specified external file.

You must also specify the certificate ID to keep the proper linkage with the private or public key pair. This step loads the certificate into the RAM cache storage of the PKI module, checks the associated private key, and verifies the signing operation.

```
user@host> request security pki local-certificate load certificate-id ms-cert filename
certnew.cer
Local certificate loaded successfully
```

3. Load the CA certificate from the specified external file.

   You must specify the CA profile to associate the CA certificate to the configured profile.

```
user@host> request security pki ca-certificate load ca-profile ms-ca filename CA-certnew.cer
Fingerprint:
1b:02:cc:cb:0f:d3:14:39:51:aa:0f:ff:52:d3:38:94:b7:11:86:30 (sha1)
90:60:53:c0:74:99:f5:da:53:d0:a0:f3:b0:23:ca:a3 (md5)
Do you want to load this CA certificate ? [yes,no] (no) yes
CA certificate for profile ms-ca loaded successfully
```

4. Load the CRL into the local storage.

   The maximum size of the CRL is 5 MB. You must specify the associated CA profile in the command.

```
user@host> request security pki crl load ca-profile ms-ca filename certcrl.crl
CRL for CA profile ms-ca loaded successfully
```

### Results

Verify that all local certificates are loaded.

```
user@host> show security pki local-certificate certificate-id ms-cert detail Certificate
identifier: ms-cert
Certificate version: 3
Serial number: 3a01c5a0000000000011
Issuer:
Organization: Example, Organizational unit: example, Country: US, State:
CA, Locality: Sunnyvale,
Common name: LAB
Subject:
```

```
Organization: Example, Organizational unit: example, Country: US,
State: CA, Locality: Sunnyvale,
Common name: john doe
Alternate subject: "user@example.net", fqdn empty, ip empty
Validity:
Not before: 11- 2-2007 22:54
Not after: 11- 2-2008 23:04
Public key algorithm: rsaEncryption(1024 bits)
30:81:89:02:81:81:00:e4:41:ba:b2:01:bf:09:31:73:5f:a2:82:fe
1c:fa:0b:36:a5:d0:1c:5a:91:4c:5f:1b:11:7b:51:66:16:1d:a5:85
15:82:ea:a3:ab:d7:34:ef:2b:39:2e:58:6a:4e:eb:58:03:40:b0:ca
1b:dc:4d:97:ff:56:9d:95:02:11:8b:84:05:4e:39:01:a8:62:3e:31
31:03:1a:7a:85:b0:90:6a:ac:1e:a8:ca:a1:ad:75:c4:94:bb:4a:94
8a:f3:2f:80:a4:15:0b:1f:21:a7:1f:7d:27:71:01:4e:ea:df:58:bd
69:08:d9:41:99:98:20:36:88:1b:e8:c6:f0:11:2d:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Distribution CRL:
ldap:///CN=LAB,CN=LABSRV1,CN=CDP,CN=Public%20Key%20Services,CN=Services,
CN=Configuration,DC=domain,DC=com?certificateRevocationList?base?
objectclass=cRLDistributionPoint
http://labsrv1.domain.com/CertEnroll/LAB.crl
Fingerprint:
c9:6d:3d:3e:c9:3f:57:3c:92:e0:c4:31:fc:1c:93:61:b4:b1:2d:58 (sha1)
50:5d:16:89:c9:d3:ab:5a:f2:04:8b:94:5d:5f:65:bd (md5)
```

You can display the individual certificate details by specifying certificate-ID in the command line.

Verify all CA certificates or the CA certificates of an individual CA profile (specified).

```
user@host> show security pki ca-certificate ca-profile ms-ca detail
Certificate identifier: ms-ca
Certificate version: 3
Serial number: 44b033d1e5e158b44597d143bbfa8a13
Issuer:
Organization: Example, Organizational unit: example, Country: US, State:
CA, Locality: Sunnyvale,
Common name: example
Subject:
Organization: Example, Organizational unit: example, Country: US, State:
CA, Locality: Sunnyvale,
Common name: example
Validity:
```

```
Not before: 09-25-2007 20:32
Not after: 09-25-2012 20:41
Public key algorithm: rsaEncryption(1024 bits)
30:81:89:02:81:81:00:d1:9e:6f:f4:49:c8:13:74:c3:0b:49:a0:56
11:90:df:3c:af:56:29:58:94:40:74:2b:f8:3c:61:09:4e:1a:33:d0
8d:53:34:a4:ec:5b:e6:81:f5:a5:1d:69:cd:ea:32:1e:b3:f7:41:8e
7b:ab:9c:ee:19:9f:d2:46:42:b4:87:27:49:85:45:d9:72:f4:ae:72
27:b7:b3:be:f2:a7:4c:af:7a:8d:3e:f7:5b:35:cf:72:a5:e7:96:8e
30:e1:ba:03:4e:a2:1a:f2:1f:8c:ec:e0:14:77:4e:6a:e1:3b:d9:03
ad:de:db:55:6f:b8:6a:0e:36:81:e3:e9:3b:e5:c9:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Distribution CRL:
ldap:///CN=LAB,CN=LABSRV1,CN=CDP,CN=Public%20Key%20Services,CN=Services,
CN=Configuration,DC=domain,DC=com?certificateRevocationList?base?
objectclass=cRLDistributionPoint
http://srv1.domain.com/CertEnroll/LAB.crl
Use for key: CRL signing, Certificate signing, Non repudiation
Fingerprint:
1b:02:cc:cb:0f:d3:14:39:51:aa:0f:ff:52:d3:38:94:b7:11:86:30 (sha1)
90:60:53:c0:74:99:f5:da:53:d0:a0:f3:b0:23:ca:a3 (md5)
```

Verify all loaded CRLs or the CRLs of the specified individual CA profile.

```
user@host> show security pki crl ca-profile ms-ca detail
CA profile: ms-ca
CRL version: V00000001
CRL issuer: emailAddress = certadmin@example.net, C = US, ST = CA,
L = Sunnyvale, O = Example, OU = example, CN = example
Effective date: 10-30-2007 20:32
Next update: 11- 7-2007 08:52
```

Verify the certificate path for the local certificate and the CA certificate.

```
user@host> request security pki local-certificate verify certificate-id ms-cert
Local certificate ms-cert verification success
user@host> request security pki ca-certificate verify ca-profile ms-ca
CA certificate ms-ca verified successfully
```

## Configuring the IPsec VPN with the Certificates

### Step-by-Step Procedure

To configure the IPsec VPN with the certificate, refer to the network diagram shown in Figure 27 on page 306

1. Configure security zones and assign interfaces to the zones.

   In this example packets are incoming on `ge-0/0/0`, and the ingress zone is the trust zone.

```
[edit security zones security-zone]
user@host# set trust interfaces ge-0/0/0.0
user@host# set untrust interfaces ge-0/0/3.0
```

2. Configure host-inbound services for each zone.

   Host-inbound services are for traffic destined for the Juniper Networks device. These settings include but are not limited to the FTP, HTTP, HTTPS, IKE, ping, rlogin, RSH, SNMP, SSH, Telnet, TFTP, and traceroute.

```
[edit security zones security-zone]
user@host# set trust host-inbound-traffic system-services all
user@host# set untrust host-inbound-traffic system-services ike
```

3. Configure the address book entries for each zone.

```
[edit security zones security-zone]
user@host# set trust address-book address local-net 192.168.10.0/24
user@host# set untrust address-book address remote-net 192.168.168.0/24
```

4. Configure the IKE (Phase 1) proposal to use RSA encryption.

```
[edit security ike proposal rsa-prop1]
user@host# set authentication-method rsa-signatures
user@host# set encryption-algorithm 3des-cbc
user@host# set authentication-algorithm sha1
user@host# set dh-group group2
```

5. Configure an IKE policy.

The phase 1 exchange can take place in either main mode or aggressive mode.

```
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals rsa-prop1
user@host# set certificate local-certificate ms-cert
user@host# set certificate peer-certificate-type x509- signature
user@host# set certificate trusted-ca use-all
```

6. Configure an IKE gateway.

   In this example, the peer is identified by an FQDN (hostname). Therefore the gateway IKE ID should be the remote peer domain name. You must specify the correct external interface or peer ID to properly identify the IKE gateway during Phase 1 setup.

```
[edit security ike gateway ike-gate]
user@host# set external-interface ge-0/0/3.0
user@host# set ike-policy ike-policy1
user@host# set dynamic hostname ssg5.example.net
```

7. Configure the IPsec policy.

   This example uses the Standard proposal set, which includes `esp-group2-3des-sha1` and `esp-group2-aes128-sha1` proposals. However, a unique proposal can be created and then specified in the IPsec policy if needed.

```
[edit security ipsec policy vpn-policy1]
user@host# set proposal-set standard
user@host# set perfect-forward-secrecy keys group2
```

8. Configure the IPsec VPN with an IKE gateway and IPsec policy.

   In this example, the ike-vpn VPN name must be referenced in the tunnel policy to create a security association. Additionally, if required, an idle time and a proxy ID can be specified if they are different from the tunnel policy addresses.

```
[edit security ipsec vpn ike-vpn ike]
user@host# set gateway ike-gate
user@host# set ipsec-policy vpn-policy1
```

9. Configure bidirectional tunnel policies for VPN traffic.

   In this example, traffic from the host LAN to the remote office LAN requires a from-zone trust to-zone untrust tunnel policy. However, if a session needs to originate from the remote LAN to the host LAN, then a tunnel policy in the opposite direction from from-zone untrust to-zone trust is also required. When you specify the policy in the opposite direction as the pair-policy, the VPN becomes bidirectional. Note that in addition to the permit action, you also need to specify the IPsec profile to be used. Note that for tunnel policies, the action is always permit. In fact, if you are configuring a policy with the deny action, you will not see an option for specifying the tunnel.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy tunnel-policy-out match source-address local-net
user@host# set policy tunnel-policy-out match destination-address remote-net
user@host# set policy tunnel-policy-out match application any
user@host# set policy tunnel-policy-out then permit tunnel ipsec-vpn ike-vpn pair-policy
tunnel-policy-in
user@host# top edit security policies from-zone untrust to-zone trust
user@host# set policy tunnel-policy-in match source-address remote-net
user@host# set policy tunnel-policy-in match destination-address local-net
user@host# set policy tunnel-policy-in match application any
user@host# set policy tunnel-policy-in then permit tunnel ipsec-vpn ike-vpn pair-policy
tunnel-policy-out
```

10. Configure a source NAT rule and a security policy for Internet traffic.

    The device uses the specified source-nat interface, and translates the source IP address and port for outgoing traffic, using the IP address of the egress interface as the source IP address and a random higher port for the source port. If required, more granular policies can be created to permit or deny certain traffic.

```
[edit security nat source rule-set nat-out]
user@host#set from zone trust
user@host#set to zone untrust
user@host#set rule interface-nat match source-address 192.168.10.0/24
user@host#set rule interface-nat match destination-address 0.0.0.0/0
user@host#set rule interface-nat then source-nat interface
```

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy any-permit match source-address any
user@host# set policy any-permit match destination-address any
```

```
user@host# set policy any-permit match application any
user@host# set policy any-permit then permit
```

11. Move the tunnel policy above the any-permit policy.

```
[edit security policies from-zone trust to-zone untrust]
user@host# insert policy tunnel-policy-out before policy any-permit
```

The security policy should be below the tunnel policy in the hierarchy because the policy list is read from top to bottom. If this policy were above the tunnel policy, then the traffic would always match this policy and would not continue to the next policy. Thus no user traffic would be encrypted.

12. Configure the tcp-mss setting for TCP traffic across the tunnel.

TCP-MSS is negotiated as part of the TCP 3-way handshake. It limits the maximum size of a TCP segment to accommodate the MTU limits on a network. This is very important for VPN traffic because 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, causing fragmentation. Because fragmentation increases the bandwidth and device resources usage, and in general it should be avoided.

The recommended value to use for tcp-mss is 1350 for most Ethernet-based networks with an MTU of 1500 or higher. This value might need to be altered if any device in the path has a lower value of MTU or if there is any added overhead such as PPP, Frame Relay, and so on. As a general rule, you might need to experiment with different tcp-mss values to obtain optimal performance.

```
user@host# set security flow tcp-mss ipsec-vpn mss mss-value
Example:
[edit]
user@host# set security flow tcp-mss ipsec-vpn mss 1350
user@host# commit and-quit
commit complete
Exiting configuration mode
```

# Verification

IN THIS SECTION


- Confirming IKE Phase 1 Status | 321
- Getting Details on Individual Security Associations | 322
- Confirming IPsec Phase 2 Status | 323
- Displaying IPsec Security Association Details | 324
- Checking IPsec SA Statistics | 326
- Testing Traffic Flow Across the VPN | 327
- Confirming the Connectivity | 327


Confirm that the configuration is working properly.

## Confirming IKE Phase 1 Status

### Purpose

Confirm the VPN status by checking any IKE Phase 1 security associations status.

PKI related to IPsec tunnels is formed during Phase 1 setup. Completion of Phase 1 indicates that PKI was successful.

### Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations


Index Remote Address State Initiator cookie Responder cookie Mode
 2010.2.2.2 UP af4f78bc135e4365 48a35f853ee95d21 Main
```

### Meaning

The output indicates that:

- The remote peer is 10.2.2.2 and the status is UP, which means the successful association of Phase 1 establishment.
- The remote peer IKE ID, IKE policy, and external interfaces are all correct.
- Index 20 is a unique value for each IKE security association. You can use this output details to get further details on each security association. See "Getting Details on Individual Security Associations" on page 322.

Incorrect output would indicate that:

- The remote peer status is Down.
- There are no IKE security associations .
- There are IKE policy parameters, such as the wrong mode type (Aggr or Main), PKI issues, or Phase 1 proposals (all must match on both peers). For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 328.
- External interface is invalid for receiving the IKE packets. Check the configurations for PKI-related issues, check the key management daemon (kmd) log for any other errors, or run trace options to find the mismatch. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 328.

## Getting Details on Individual Security Associations

### Purpose

Get details on individual IKE.

### Action

From operational mode, enter the **show security ike security-associations index 20 detail** command.

```
user@host> show security ike security-associations index 20 detail
IKE peer 10.2.2.2, Index 20,
Role: Responder, State: UP
Initiator cookie: af4f78bc135e4365, Responder cookie: 48a35f853ee95d21
Exchange type: Main, Authentication method: RSA-signatures
Local: 10.1.1.2:500, Remote: 10.2.2.2:500
Lifetime: Expires in 23282 seconds
```

```
Algorithms:
Authentication : sha1
Encryption : 3des-cbc
Pseudo random function: hmac-sha1
Traffic statistics:
Input bytes : 10249
Output bytes : 4249
Input packets: 10
Output packets: 9
Flags: Caller notification sent
IPsec security associations: 2 created, 1 deleted
Phase 2 negotiations in progress: 0
```

### Meaning

The output displays the details of the individual IKE SAs such as role (initiator or responder), status, exchange type, authentication method, encryption algorithms, traffic statistics, Phase 2 negotiation status, and so on.

You can use the output data to:

- Know the role of the IKE SA. Troubleshooting is easier when the peer has the responder role.
- Get the traffic statistics to verify the traffic flow in both directions.
- Get the number of IPsec security associations created or in progress.
- Get the status of any completed Phase 2 negotiations.

## Confirming IPsec Phase 2 Status

### Purpose

View IPsec (Phase 2) security associations.

When IKE Phase 1 is confirmed, view the IPsec (Phase 2) security associations.

### Action

From operational mode, enter the **show security ipsec security-associations** command.

```
user@host> show security ipsec security-associations
```

```
total configured sa: 2
ID Gateway Port Algorithm SPI Life:sec/kb Mon vsys
<2 10.2.2.2 500 ESP:3des/sha1 bce1c6e0 1676/ unlim - 0
>2 10.2.2.2 500 ESP:3des/sha1 1a24eab9 1676/ unlim - 0
```

### Meaning

The output indicates that:

- There is a configured IPsec SA pair available . The port number 500 indicates that a standard IKE port is used. Otherwise, it is Network Address Translation-Traversal (NAT-T), 4500, or random high port.
- The security parameter index (SPI) is used for both directions. The lifetime or usage limits of the SA is expressed either in seconds or in kilobytes. In the output, 1676/ unlim indicates Phase 2 lifetime is set to expire in 1676 seconds and there is no specified lifetime size.
- The ID number shows the unique index value for each IPsec SA.
- A hyphen (-) in the Mon column indicates that VPN monitoring is not enabled for this SA.
- The virtual system (vsys) is zero, which is the default value.

Phase 2 lifetime can be different from the Phase 1 lifetime because Phase 2 is not dependent on Phase 1 after the VPN is up.

## Displaying IPsec Security Association Details

### Purpose

Display the individual IPsec SA details identified by the index number.

### Action

From operational mode, enter the **show security ipsec security-associations index 2 detail** command.

```
user@host> show security ipsec security-associations index 2 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)
DF-bit: clear
Policy-name: tunnel-policy-out
Direction: inbound, SPI: bce1c6e0, AUX-SPI: 0
```

```
Hard lifetime: Expires in 1667 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 1093 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Anti-replay service: enabled, Replay window size: 32
Direction: outbound, SPI: 1a24eab9, AUX-SPI: 0
Hard lifetime: Expires in 1667 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 1093 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Anti-replay service: enabled, Replay window size: 32
```

### Meaning

The output displays the local Identity and the remote Identity.

Note that a proxy ID mismatch can cause Phase 2 completion to fail. The proxy ID is derived from the tunnel policy (for policy-based VPNs). The local address and remote address are derived from the address book entries, and the service is derived from the application configured for the policy.

If Phase 2 fails due to a proxy ID mismatch, verify which address book entries are configured in the policy and ensure that the correct addresses are sent. Also ensure that the ports are matching. Double-check the service to ensure that the ports match for the remote and local servers.

If multiple objects are configured in a tunnel policy for source address, destination address, or application, then the resulting proxy ID for that parameter is changed to zeroes.

For example, assume the following scenario for a tunnel policy:

- Local addresses of 192.168.10.0/24 and 10.10.20.0/24
- Remote address of 192.168.168.0/24
- Application as junos-http

The resulting proxy ID is local 0.0.0.0/0, remote 192.168.168.0/24, service 80.

The resulting proxy IDs can affect the interoperability if the remote peer is not configured for the second subnet. Also, if you are employing a third-party vendor's application, you might have to manually enter the proxy ID to match.

If IPsec fails to complete, then check the kmd log or use the `set traceoptions` command. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 328.

## Checking IPsec SA Statistics

### Purpose

Check statistics and errors for an IPsec SA.

For troubleshooting purpose, check the Encapsulating Security Payload/Authentication Header (ESP/AH) counters for any errors with a particular IPsec SA.

### Action

From operational mode, enter the **show security ipsec statistics index 2** command.

```
user@host> show security ipsec statistics index 2
ESP Statistics:
Encrypted bytes: 674784
Decrypted bytes: 309276
Encrypted packets: 7029
Decrypted packets: 7029
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

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. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 328.

## Testing Traffic Flow Across the VPN

### Purpose

Test traffic flow across the VPN after Phase 1 and Phase 2 have completed successfully. You can test traffic flow by using the `ping` command. You can ping from local host to remote host. You can also initiate pings from the Juniper Networks device itself.

This example shows how to initiate a ping request from the Juniper Networks device to the remote host. Note that when pings are initiated from the Juniper Networks device, the source interface must be specified to ensure that the correct route lookup takes place and the appropriate zones are referenced in the policy lookup.

In this example, the ge-0/0/0.0 interface resides in the same security zone as the local host and must be specified in the ping request so that the policy lookup can be from zone trust to zone untrust.

### Action

From operational mode, enter the **ping 192.168.168.10 interface ge-0/0/0 count 5** command.

```
user@host> 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
```

## Confirming the Connectivity

### Purpose

Confirm the connectivity between a remote host and a local host.

### Action

From operational mode, enter the **ping 192.168.10.10 from ethernet0/6** command.

```
ssg5-> 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

You can confirm end-to-end connectivity by using the `ping` command from the remote host to the local host. In this example, the command is initiated from the SSG5 device.

Failed end-to-end connectivity can indicate an issue with routing, policy, end host, or encryption/decryption of the ESP packets. To verify the exact causes of the failure:

- Check IPsec statistics for details on errors as described in "Checking IPsec SA Statistics" on page 326.
- Confirm end host connectivity by using the `ping` command from a host on the same subnet as the end host. If the end host is reachable by other hosts, then you can assume that the issue is not with the end host.
- Enable security flow trace options for troubleshooting the routing-related and policy-related issues.

## Troubleshooting IKE, PKI, and IPsec Issues

**IN THIS SECTION**


- Basic Troubleshooting Steps | **329**
- Checking the Free Disk Space on Your Device | **330**
- Checking the Log Files to Verify Different Scenarios and Uploading Log Files to an FTP | **330**
- Enabling IKE Trace Options to View Messages on IKE | **331**
- Enabling PKI Trace Options to View Messages on IPsec | **333**
- Setting up IKE and PKI Trace Options to Troubleshoot IKE Setup Issues with Certificates | **334**

- Analyzing the Phase 1 Success Message | 334
- Analyzing the Phase 1 Failure Message (Proposal Mismatch) | 335
- Analyzing the Phase 1 Failure Message (Authentication Failure) | 336
- Analyzing the Phase 1 Failure Message (Timeout Error) | 336
- Analyzing the Phase 2 Failure Message | 337
- Analyzing the Phase 2 Failure Message | 338
- Troubleshooting Common Problems Related to IKE and PKI | 339


Troubleshoot IKE, PKI, and IPsec issues.

## Basic Troubleshooting Steps

### Problem

The basic troubleshooting steps are as follows:

1. Identifying and isolating the problem.
2. Debugging the problem.

The common approach of starting troubleshooting is with the lowest layer of the OSI layers and working your way up the OSI stack to confirm the layer in which the failure occurs.

### Solution

Basic steps for troubleshooting IKE, PKI, and IPsec are as follows:

- Confirm the physical connectivity of the Internet link at the physical and data link levels.
- Confirm that the Juniper Networks device has connectivity to the Internet next hop and connectivity to the remote IKE peer.
- Confirm IKE Phase 1 completion.
- Confirm IKE Phase 2 completion if IKE Phase 1 completion is successful.
- Confirm the traffic flow across the VPN (if the VPN is up and active).

Junos OS includes the trace options feature. Using this feature, you can enable a trace option flag to write the data from the trace option to a log file, which can be predetermined or manually configured

and stored in flash memory. These trace logs can be retained even after a system reboot. Check the available flash storage before implementing trace options.

You can enable the trace options feature in configuration mode and commit the configuration to use the trace options feature. Similarly to disable trace options, you must deactivate trace options in configuration mode and commit the configuration.

## Checking the Free Disk Space on Your Device

### Problem

Check the statistics on the free disk space in your device file systems.

### Solution

From operational mode, enter the **show system storage** command.

```
user@host> show system storage
Filesystem Size Used Avail Capacity Mounted on
/dev/ad0s1a 213M 74M 137M 35% /
devfs 1.0K 1.0K 0B 100% /dev
devfs 1.0K 1.0K 0B 100% /dev/
/dev/md0 180M 180M 0B 100% /junos
/cf 213M 74M 137M 35% /junos/cf
devfs 1.0K 1.0K 0B 100% /junos/dev/
procfs 4.0K 4.0K 0B 100% /proc
/dev/bo0s1e 24M 13K 24M 0% /config
/dev/md1 168M 7.6M 147M 5% /mfs
/cf/var/jail 213M 74M 137M 35% /jail/var
```

The `/dev/ad0s1a` represents the onboard flash memory and is currently at 35 percent capacity.

## Checking the Log Files to Verify Different Scenarios and Uploading Log Files to an FTP

### Problem

View the log files to check security IKE debug messages, security flow debugs, and the state of logging to the syslog.

### Solution

From operational mode, enter the **show log kmd**, **show log pkid**, **show log security-trace**, and **show log messages** commands.

```
user@host> show log kmd
user@host> show log pkid
user@host> show log security-trace
user@host> show log messages
```

You can view a list of all logs in the /var/log directory by using the `show log` command.

Log files can also be uploaded to an FTP server by using the `file copy` command.

```
(operational mode):
user@host> file copy path/filename dest-path/filename
Example:
```

```
user@host> file copy /var/log/kmd ftp://192.168.10.10/kmd.log
```

```
ftp://192.168.10.10/kmd.log 100% of 35 kB 12 MBps
```

## Enabling IKE Trace Options to View Messages on IKE

### Problem

To view success or failure messages for IKE or IPsec, you can view the kmd log by using the `show log kmd` command. Because the kmd log displays some general messages, it can be useful to obtain additional details by enabling IKE and PKI trace options.

Generally, it is best practice to troubleshoot the peer that has the responder role. You must obtain the trace output from the initiator and responder to understand the cause of a failure.

Configure IKE tracing options.

### Solution

```
user@host> configure
Entering configuration mode

[edit]
user@host# edit security ike traceoptions
[edit security ike traceoptions]
```

```
user@host# set file ?
Possible completions:
<filename> Name of file in which to write trace information
files Maximum number of trace files (2..1000)
match Regular expression for lines to be logged
no-world-readable Don't allow any user to read the log file
size Maximum trace file size (10240..1073741824)
world-readable Allow any user to read the log file
```

```
[edit security ike traceoptions]
```

```
user@host# set flag ?
Possible completions:
all Trace everything
certificates Trace certificate events
database Trace security associations database events
general Trace general events
ike Trace IKE module processing
parse Trace configuration processing
policy-manager Trace policy manager processing
routing-socket Trace routing socket messages
timer Trace internal timer events
```

If you do not specify file names for the <filename> field, then all IKE trace options are written to the kmd log.

You must specify at least one flag option to write trace data to the log. For example:

- `file size` — Maximum size of each trace file, in bytes. For example, 1 million (1,000,000 ) can generate a maximum file size of 1 MB.

- `files` — Maximum number of trace files to be generated and stored in a flash memory device.

You must commit your configuration to start the trace.

## Enabling PKI Trace Options to View Messages on IPsec

### Problem

Enable PKI trace options to identify whether an IKE failure is related to the certificate or to a non-PKI issue.

### Solution

```
[edit security pki traceoptions]
```

```
user@host# set file ?
Possible completions:
<filename> Name of file in which to write trace information
files Maximum number of trace files (2..1000)
match Regular expression for lines to be logged
no-world-readable Don't allow any user to read the log file
size Maximum trace file size (10240..1073741824)
world-readable Allow any user to read the log file
```

```
[edit security pki traceoptions]
```

```
user@host# set flag ?
Possible completions:
all Trace with all flags enabled
certificate-verification PKI certificate verification tracing
online-crl-check PKI online crl tracing
```

## Setting up IKE and PKI Trace Options to Troubleshoot IKE Setup Issues with Certificates

### Problem

Configure the recommended settings for IKE and PKI trace options.

The IKE and PKI trace options use the same parameters, but the default filename for all PKI-related traces is found in the pkid log.

### Solution

```
user@host> configure
Entering configuration mode

[edit security ike traceoptions]
user@host# set file size 1m
user@host# set flag ike
user@host# set flag policy-manager
user@host# set flag routing-socket
user@host# set flag certificates

[edit security pki traceoptions]
user@host# set file size 1m
user@host# set flag all
user@host# commit and-quit
commit complete
Exiting configuration mode
```

## Analyzing the Phase 1 Success Message

### Problem

Understand the output of the `show log kmd` command when the IKE Phase 1 and Phase 2 conditions are successful.

### Solution

```
Nov 7 11:52:14 Phase-1 [responder] done for local=ipv4(udp:500,[0..3]=
10.1.1.2) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
```

```
Nov 7 11:52:14 Phase-2 [responder] done for
p1_local=ipv4(udp:500,[0..3]=10.1.1.2) p1_remote=fqdn(udp:500,[0..15]=ssg5.example.net)
p2_local=ipv4_subnet(any:0,[0..7]=192.168.10.0/24)
p2_remote=ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `ssg5.example.net` —Remote peer (hostname with FQDN).
- `udp: 500`—NAT-T was not negotiated.
- `Phase 1 [responder] done`—Phase 1 status, along with the role (initiator or responder).
- `Phase 2 [responder] done`—Phase 1 status, along with the proxy ID information.

  You can also confirm the IPsec SA status by using the verification commands mentioned in "Confirming IKE Phase 1 Status" on page 321.

## Analyzing the Phase 1 Failure Message (Proposal Mismatch)

### Problem

Understanding the output of the `show log kmd` command, where the IKE Phase 1 condition is a failure, helps in determining the reason for the VPN not establishing Phase 1.

### Solution

```
Nov 7 11:52:14 Phase-1 [responder] failed with error(No proposal chosen) for
local=unknown(any:0,[0..0]=) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
Nov 7 11:52:14 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { 011359c9 ddef501d - 2216ed2a bfc50f5f
[-
1] / 0x00000000 } IP; Error = No proposal chosen (14)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `ssg5.example.net` —Remote peer (hostname with FQDN).
- `udp: 500`—NAT-T was not negotiated.
- `Phase-1 [responder] failed with error (No proposal chosen)`—Phase 1 failure because of proposal mismatch.

To resolve this issue, ensure that the parameters for the IKE gateway Phase 1 proposals on both the responder and the initiator match. Also confirm that a tunnel policy exists for the VPN.

## Analyzing the Phase 1 Failure Message (Authentication Failure)

### Problem

Understand the output of the `show log kmd` command when the IKE Phase 1 condition is a failure. This helps in determining the reason for the VPN not establishing Phase 1.

### Solution

```
Nov 7 12:06:36 Unable to find phase-1 policy as remote peer:10.2.2.2 is not recognized.
Nov 7 12:06:36 Phase-1 [responder] failed with error(Authentication failed) for
local=ipv4(udp:500,[0..3]=10.1.1.2) remote=ipv4(any:0,[0..3]=10.2.2.2)
Nov 7 12:06:36 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { f725ca38 dad47583 - dab1ba4c ae26674b
[-
1] / 0x00000000 } IP; Error = Authentication failed (24)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `10.2.2.2`—Remote peer
- `Phase 1 [responder] failed with error (Authentication failed)`—Phase 1 failure due to the responder not recognizing the incoming request originating from a valid gateway peer. In the case of IKE with PKI certificates, this failure typically indicates that an incorrect IKE ID type was specified or entered.

To resolve this issue, confirm that the correct peer IKE ID type is specified on the local peer based on the following:

- How the remote peer certificate was generated
- Subject Alternative Name or DN information in the received remote peer certificate

## Analyzing the Phase 1 Failure Message (Timeout Error)

### Problem

Understand the output of the `show log kmd` command when the IKE Phase 1 condition is a failure.

### Solution

```
Nov 7 13:52:39 Phase-1 [responder] failed with error(Timeout) for local=unknown(any:0,[0..0]=)
remote=ipv4(any:0,[0..3]=10.2.2.2)
```

The sample output indicates:

- `10.1.1.2`—Llocal address.
- `10.2.2.2`—Remote peer.
- `Phase 1 [responder] failed with error(Timeout)`—Phase 1 failure.

  This error indicates that either the IKE packet is lost enroute to the remote peer or there is a delay or no response from the remote peer.

Because this timeout error is the result of waiting on a response from the PKI daemon, you must review the PKI trace options output to see whether there is a problem with PKI.

## Analyzing the Phase 2 Failure Message

### Problem

Understand the output of the `show log kmd` command when the IKE Phase 2 condition is a failure.

### Solution

```
Nov 7 11:52:14 Phase-1 [responder] done for local=ipv4(udp:500,[0..3]=
10.1.1.2) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
Nov 7 11:52:14 Failed to match the peer proxy ids
p2_remote=ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
p2_local=ipv4_subnet(any:0,[0..7]=10.10.20.0/24) for the remote peer:ipv4(udp:500,
[0..3]=10.2.2.2)
Nov 7 11:52:14 KMD_PM_P2_POLICY_LOOKUP_FAILURE: Policy lookup for Phase-2 [responder] failed for
p1_local=ipv4(udp:500,[0..3]=10.1.1.2) p1_remote=ipv4(udp:500,[0..3]=10.2.2.2)
p2_local=ipv4_subnet(any:0,[0..7]=10.10.20.0/24)
p2_remote=ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
Nov 7 11:52:14 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { 41f638eb cc22bbfe - 43fd0e85 b4f619d5
[0]
/ 0xc77fafcf } QM; Error = No proposal chosen (14)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `ssg5.example.net` —Remote peer (IKE ID type hostname with FQDN).
- `Phase 1 [responder] done`—Phase 1 success.
- `Failed to match the peer proxy ids`—The Incorrect proxy IDs are received. In the previous sample, the two proxy IDs received are 192.168.168.0/24 (remote) and 10.10.20.0/24 (local) (for service=any). Based on the configuration given in this example, the expected local address is 192.168.10.0/24. This shows that there is a mismatch of configurations on the local peer, resulting in the failure of proxy ID match.

  To resolve this issue, correct the address book entry or configure the proxy ID on either peer so that it matches the other peer.

  The output also indicates the reason for failure is `No proposal chosen`. However in this case you also see the message `Failed to match the peer proxy ids`.

## Analyzing the Phase 2 Failure Message

### Problem

Understand the output of the `show log kmd` command when the IKE Phase 2 condition is a failure.

### Solution

```
Nov 7 11:52:14 Phase-1 [responder] done for local=ipv4(udp:500,[0..3]=
10.1.1.2) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
Nov 7 11:52:14 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { cd9dff36 4888d398 - 6b0d3933 f0bc8e26
[0]
/ 0x1747248b } QM; Error = No proposal chosen (14)
```

The sample output indicates:

- `10.1.1.2` —Local address.
- `fqdn(udp:500,[0..15]=ssg5.example.net`—Remote peer.
- `Phase 1 [responder] done`—Phase 1 success.
- `Error = No proposal chosen`—No proposal was chosen during Phase 2. This issue is due to proposal mismatch between the two peers.

  To resolve this issue, confirm that the Phase 2 proposals match on both peers.

## Troubleshooting Common Problems Related to IKE and PKI

### Problem

Troubleshoot common problems related to IKE and PKI.

Enabling the trace options feature helps you to gather more information on the debugging issues than is obtainable from the normal log entries. You can use the trace options log to understand the reasons for IKE or PKI failures.

### Solution

Methods for troubleshooting the IKE -and-PKI-related issues:

- Ensure that the clock, date, time zone, and daylight savings settings are correct. Use NTP to keep the clock accurate.
- Ensure that you use a two-letter country code in the "C=" (country) field of the DN.

  For example: use “US” and not “USA” or “United States.” Some CAs require that the country field of the DN be populated, allowing you to enter the country code value only with a two-letter value.
- Ensure that if a peer certificate is using multiple OU=or CN= fields, you are using the distinguished name with container method (the sequence must be maintained and is case- sensitive).
- If the certificate is not valid yet, check the system clock and, if required, adjust the system time zone or just add a day in the clock for a quick test.
- Ensure that a matching IKE ID type and value are configured.
- PKI can fail due to a revocation check failure. To confirm this, temporarily disable revocation checking and see whether IKE Phase 1 is able to complete.

  To disable revocation checking, use the following command in configure mode:

  ```
  set security pki ca-profile <ca-profile> revocation-check disable
  ```

RELATED DOCUMENTATION

IPsec Overview | 20

PKI Components In Junos OS | 33

# Configure IPsec VPN with OCSP for Certificate Revocation Status

IN THIS SECTION


- Requirements | 340
- Overview | 340
- Configuration | 344
- Verification | 355


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 | 343

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 32 on page 341 shows the Phase 1 options used in this example.

**Table 32: 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 |

**Table 32: Phase 1 Options for OCSP Configuration Example *(Continued)***

| Option | Peer A | Peer B |
|---|---|---|
| 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 |
| Version | v2 | v2 |

Table 33 on page 342 shows the Phase 2 options used in this example.

**Table 33: 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 |

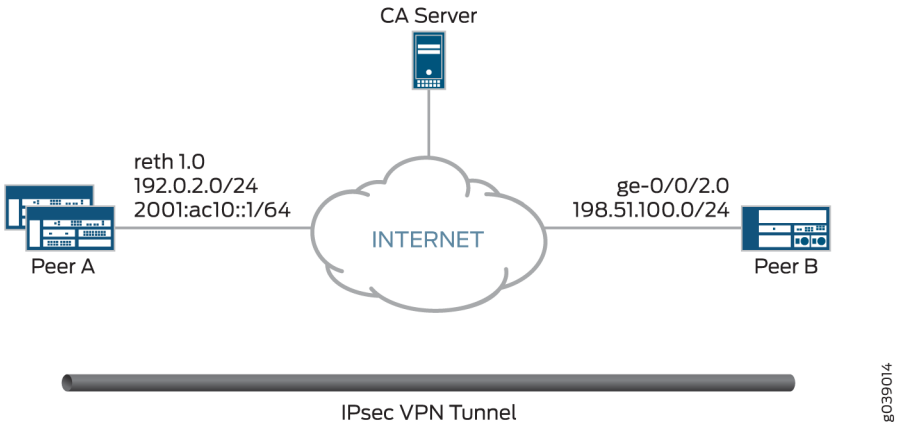
**Table 33: Phase 2 Options for OCSP Configuration Example *(Continued)***

| Option | Peer A | Peer B |
|---|---|---|
| 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 |
| Policy | ipsec_policy | ipsec_policy |
| Establish tunnels | - | immediately |

## Topology

Figure 28 on page 344 shows the peer devices that are configured in this example.

**Figure 28: OCSP Configuration Example**




## Configuration

**IN THIS SECTION**


- Configuring Peer A | 344
- Configuring Peer B | 350


### 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 | 355
- Verifying Local Certificates | 357
- Verifying IKE Phase 1 Status | 359
- Verifying IPsec Phase 2 Status | 360


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

PKI Components In Junos OS | 33

# IPv6 IPsec VPNs

IN THIS SECTION


- VPN Feature Support for IPv6 Addresses | 362
- Understanding IPv6 IKE and IPsec Packet Processing | 368
- IPv6 IPsec Configuration Overview | 375
- Example: Configuring an IPv6 IPsec Manual VPN | 375
- Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN | 379

Juniper Networks supports manual and autokey IKE with preshared keys configurations for IPv6 IPsec VPN.

## 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 34 on page 362 defines the support for IPv6 addresses in VPN features.

**Table 34: 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 | – |
| Policy-based VPN | Yes | IPv6 policy-based VPNs are not supported on SRX Series Firewalls in chassis cluster configurations. IPv6 policy-based VPNs are only supported with IPv6-in-IPv6 tunnels on standalone SRX300, SRX320, SRX340, SRX345, and SRX550HM devices. |
| 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 | – |

**Table 34: IPv6 Address Support in VPN Features *(Continued)***

| Feature | Supported | Exceptions |
|---|---|---|
| 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 does not support IPv6 traffic. |
| Group VPN | No | – |
| Point-to-point tunnel interfaces | Yes | – |
| Point-to-multipoint tunnel interfaces | No | – |
| Hub-and-spoke scenario for site-to-site VPNs | Yes | – |
| 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 | – |

**Table 34: IPv6 Address Support in VPN Features *(Continued)***

| Feature | Supported | Exceptions |
|---|---|---|
| Chassis cluster | Yes | IPsec VPN with active-active mode is supported only on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices for route-based IPv6 tunnels. IPsec VPN with active-active mode is not supported on SRX5400, SRX5600, and SRX5800 devices. |
| 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 | – |
| 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 | – |

**Table 34: IPv6 Address Support in VPN Features *(Continued)***

| Feature | Supported | Exceptions |
|---|---|---|
| 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 supported in Junos OS Release 12.1X45-D10 release for SRX Series Firewalls. | 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. |
| 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 | – |

**Table 34: IPv6 Address Support in VPN Features *(Continued)***

| Feature | Supported | Exceptions |
|---|---|---|
| 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 | – |
| 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 | – |

**Table 34: IPv6 Address Support in VPN Features *(Continued)***

| Feature | Supported | Exceptions |
|---|---|---|
| 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 | – |
| Automatic or manual revocation over IPv6 | No | – |
| IPv6 addresses within PKI certificate fields | No | – |

## SEE ALSO

Understanding VPN Tunnel Modes | 508

IPsec Overview | 20

# Understanding IPv6 IKE and IPsec Packet Processing

IN THIS SECTION


- IPv6 IKE Packet Processing | 368
- IPv6 IPsec Packet Processing | 370


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 35 on page 368 lists the ID types and their assigned values in the identification payload.

  **Table 35: ISAKMP ID Types and Their Values**

| ID Type | Value |
|---|---|
| RESERVED | 0 |
| ID_IPV4_ADDR | 1 |
| ID_FQDN | 2 |

**Table 35: ISAKMP ID Types and Their Values *(Continued)***

| ID Type | Value |
|---|---|
| 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 |
| 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.

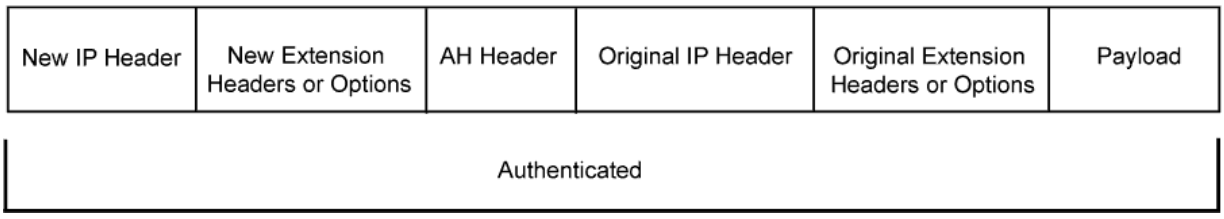
Devices with IPv6 addressing do not perform fragmentation. IPv6 hosts should either perform path MTU discovery or send packets smaller than the IPv6 minimum MTU size of 1280 bytes.

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 29 on page 370.

**Figure 29: IPv6 AH Tunnel Mode**



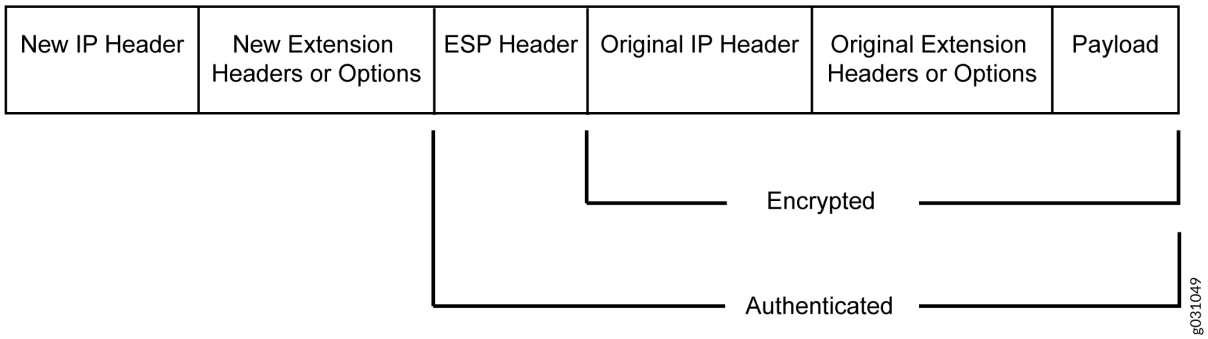

Unlike ESP, the AH authentication algorithm covers the outer header as well as any new extension headers and options.

AH tunnel mode on SRX Series Firewalls does not support IPv4 mutable options or IPv6 mutable extension headers. See Table 36 on page 371.

### 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 30 on page 371.

**Figure 30: 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 SRX Series Firewalls. Table 36 on page 371 shows the IPv4 options or IPv6 extension headers that are supported with the ESP or AH protocol on SRX Series Firewalls. If an unsupported IPsec packet is received, ICV calculation fails and the packet is dropped.

**Table 36: Support for IPv4 Options or IPv6 Extension Headers**

| Options or Extension Headers | SRX300, SRX320, SRX340, SRX345, and SRX550HM Devices | SRX5400, SRX5600, and SRX5800 Devices |
|---|---|---|
| ESP with IPv4 options | Supported | Supported |

**Table 36: Support for IPv4 Options or IPv6 Extension Headers *(Continued)***

| Options or Extension Headers | SRX300, SRX320, SRX340, SRX345, and SRX550HM Devices | SRX5400, SRX5600, and SRX5800 Devices |
|---|---|---|
| ESP with IPv6 extension headers | Supported | Supported |
| AH with IPv4 immutable options | Supported | Supported |
| AH with IPv6 immutable extension headers | Supported | Supported |
| AH with IPv4 mutable options | Not supported | Not supported |
| AH with IPv6 mutable extension headers | Not supported | Not supported |

### 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 37 on page 373 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 37: 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 38 on page 373 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 38: IPv4 Header Construction for IPv6-in-IPv4 and IPv4-in-IPv4 Tunnel Modes**

| Header Fields | Outer Header | Inner Header |
|---|---|---|
| version | 4. | No change. |

**Table 38: IPv4 Header Construction for IPv6-in-IPv4 and IPv4-in-IPv4 Tunnel Modes *(Continued)***

| Header Fields | Outer Header | Inner Header |
|---|---|---|
| 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 | 20

IPv6 IPsec Configuration Overview | 375

## 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 10

**SEE ALSO**

IPsec VPN with Autokey IKE Configuration Overview | 202

Example: Configuring an IPv6 address as the Source Address for a CA Profile | 50

## Example: Configuring an IPv6 IPsec Manual VPN

**IN THIS SECTION**

Requirements | 376

- Overview | 376
- Configuration | 376
- Verification | 379


This example shows how to configure an IPv6 IPsec manual VPN.

## Requirements

Before you begin:

- Understand how VPNs work. See "IPsec Overview" on page 20.
- Understand IPv6 IPsec packet processing. See "Understanding IPv6 IKE and IPsec Packet Processing" on page 368.

## 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 | 377

**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 | 379

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 | 375

# Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN

IN THIS SECTION

- Requirements | 380
- Overview | 380
- Configuration | 385
- Verification | 400

This example shows how to configure a policy-based IPv6 AutoKey IKE VPN to allow IPv6 data to be securely transferred between the branch office and the corporate office.

IPv6 policy-based VPNs are supported only on standalone SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

## Requirements

This example uses the following hardware:

- SRX300 device

Before you begin:

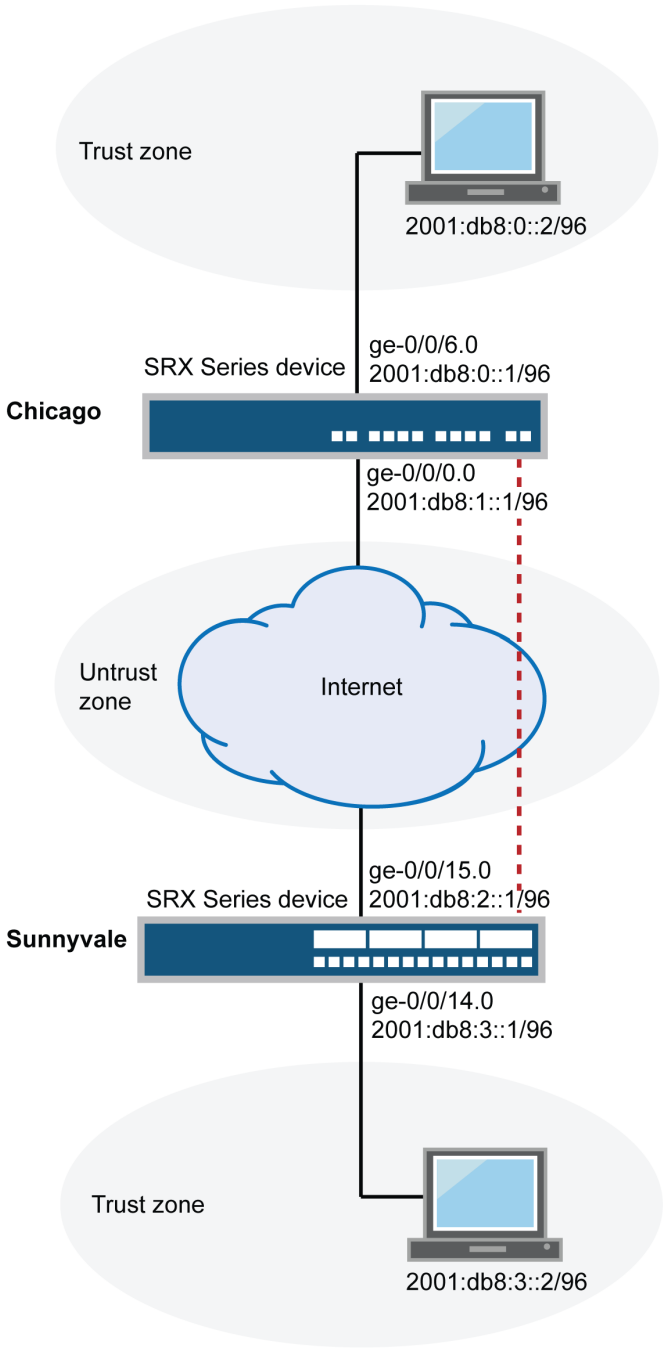
- Understand how VPNs work. See "IPsec Overview" on page 20.
- Understand IPv6 IKE and IPsec packet processing. See "Understanding IPv6 IKE and IPsec Packet Processing" on page 368.

## Overview

In this example, you configure an IPv6 IKE policy-based VPN for a branch office in Chicago, Illinois, 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, California.

Figure 31 on page 381 shows an example of an IPv6 IKE policy-based VPN topology. In this topology, one SRX Series Firewall is located in Sunnyvale, and another SRX Series Firewall (this can be a second SRX Series Firewall or a third-party device) is located in Chicago.

**Figure 31: IPv6 IKE Policy-Based VPN Topology**




In this example, you configure interfaces, an IPv6 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 39 on page 382 through Table 43 on page 385.

**Table 39: Interface, Security Zone, and Address Book Information**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Interfaces | ge-0/0/14.0 | 2001:db8:3::1/96 |
| | ge-0/0/15.0 | 2001:db8:0:2::1/96 |
| Security zones | Trust | • All system services are allowed.<br>• The ge-0/0/14.0 interface is bound to this zone. |
| | Untrust | • IKE is the only allowed system service.<br>• The ge-0/0/15.0 interface is bound to this zone. |
| Address book entries | Sunnyvale | • This address is for the Trust zone's address book.<br>• The address for this address book entry is 2001:db8:3::2/96. |
| | Chicago | • This address is for the Untrust zone's address book.<br>• The address for this address book entry is 2001:db8:0::2/96. |

**Table 40: IPv6 IKE Phase 1 Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipv6-ike-phase1-proposal | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: sha1<br>• Encryption algorithm: aes-128-cbc |
| Policy | ipv6-ike-phase1-policy | • Mode: Aggressive<br>• Proposal reference: ipv6-ike-phase1-proposal<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |
| Gateway | gw-Chicago | • IKE policy reference: ipv6-ike-phase1-policy<br>• External interface: ge-0/0/15.0<br>• Gateway address: 2001:db8:1::1/96 |

**Table 41: IPv6 IPsec Phase 2 Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipv6-ipsec-phase2-proposal | • Protocol: esp<br>• Authentication algorithm: hmac-sha1-96<br>• Encryption algorithm: aes-128-cbc |
| Policy | ipv6-ipsec-phase2-policy | • Proposal reference: ipv6-ipsec-phase2-proposal<br>• PFS: Diffie-Hellman group2 |

**Table 41: IPv6 IPsec Phase 2 Configuration Parameters *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| VPN | ipv6-ike-vpn-chicago | • IKE gateway reference: gw-chicago<br>• IPsec policy reference: ipv6-ipsec-phase2-policy |

**Table 42: Security Policy Configuration Parameters**

| Purpose | Name | Configuration Parameters |
|---|---|---|
| This security policy permits traffic from the Trust zone to the Untrust zone. | ipv6-vpn-tr-untr | • Match criteria:<br>  • source-address Sunnyvale<br>  • destination-address Chicago<br>  • application any<br>• Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago<br>• Permit action: tunnel pair-policy ipv6-vpn-untr-tr |
| This security policy permits traffic from the Untrust zone to the Trust zone. | ipv6-vpn-untr-tr | • Match criteria:<br>  • source-address Chicago<br>  • destination-address Sunnyvale<br>  • application any<br>• Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago<br>• Permit action: tunnel pair-policy ipv6-vpn-tr-untr |

**Table 42: Security Policy Configuration Parameters *(Continued)***

| Purpose | Name | Configuration Parameters |
|---|---|---|
| This security policy permits all traffic from the Trust zone to the Untrust zone.<br><br>You must put the ipv6-vpn-tr-untr policy before the permit-any security policy. Junos OS performs a security policy lookup starting at the top of the list. If the permit-any policy comes before the ipv6-vpn-tr-untr policy, all traffic from the Trust zone will match the permit-any policy and be permitted. Thus, no traffic will ever match the ipv6-vpn-tr-untr policy. | permit-any | • Match criteria:<br>  • source-address any<br>  • source-destination any<br>  • application any<br>• Action: permit |

**Table 43: 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. This is especially important for VPN traffic, as 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, thus causing fragmentation. Fragmentation results in increased use of bandwidth and device resources.<br><br>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 Basic Network, Security Zone, and Address Book Information | **386**
- Configuring IKE | **390**
- Configuring IPsec | **393**

Configuring Security Policies | 395
Configuring TCP-MSS | 399

### Configuring Basic Network, 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-0/0/14 unit 0 family inet6 address 2001:db8:3::1/96
set interfaces ge-0/0/15 unit 0 family inet6 address 2001:db8:2::1/96
set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
set security zones security-zone Untrust interfaces ge-0/0/15.0
set security zones security-zone Untrust host-inbound-traffic system-services ike
set security zones security-zone Trust interfaces ge-0/0/14.0
set security zones security-zone Trust host-inbound-traffic system-services all
set security address-book book1 address Sunnyvale 2001:db8:3::2/96
set security address-book book1 attach zone Trust
set security address-book book2 address Chicago 2001:db8:0::2/96
set security address-book book2 attach zone Untrust
```

#### 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 basic network, security zone, and address book information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/14 unit 0 family inet6 address 2001:db8:3::1/96
user@host# set interfaces ge-0/0/15 unit 0 family inet6 address 2001:db8:2::1/96
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
```

3. Configure the Untrust security zone.

```
[edit]
user@host# edit security zones security-zone Untrust
```

4. Assign an interface to the Untrust security zone.

```
[edit security zones security-zone Untrust]
user@host# set interfaces ge-0/0/15.0
```

5. Specify allowed system services for the Untrust 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-0/0/14.0
```

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. Create an address book and attach a zone to it.

```
[edit security address-book book1]
user@host# set address Sunnyvale 2001:db8:3::2/96
user@host# set attach zone Trust
```

10. Create another address book and attach a zone to it.

```
[edit security address-book book2]
user@host# set address Chicago 2001:db8:0::2/96
user@host# set attach zone Untrust
```

### Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, and `show security address-book` 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-0/0/14 {
    unit 0 {
        family inet6 {
            address 2001:db8:3::1/96;
        }
    }
}
ge-0/0/15 {
    unit 0 {
        family inet6 {
            address 2001:db8:2::1/96;
        }
    }
}
```

```
[edit]
user@host# show routing-options
static {
```

```
    route 0.0.0.0/0 next-hop 10.1.1.1;
}
```

```
[edit]
user@host# show security zones
security-zone Untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/15.0;
    }
}
security-zone Trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/14.0;
    }
}
[edit]
user@host# show security address-book
book1 {
    address Sunnyvale 2001:db8:3::2/96;
    attach {
        zone Trust;
    }
}
    book2 {
        address Chicago 2001:db8:0::2/96;
        attach {
            zone Untrust;
        }
    }
```

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

## 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 ipv6-ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ipv6-ike-phase1-proposal dh-group group2
set security ike proposal ipv6-ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ipv6-ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ipv6-ike-phase1-policy mode aggressive
set security ike policy ipv6-ike-phase1-policy proposals ipv6-ike-phase1-proposal
set security ike policy ipv6-ike-phase1-policy pre-shared-key ascii-text 111111111111111
set security ike gateway gw-chicago external-interface ge-0/0/15.0
set security ike gateway gw-chicago ike-policy ipv6-ike-phase1-policy
set security ike gateway gw-chicago address 2001:db8:0:1::1/96
```

## 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 ipv6-ike-phase1-proposal
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ipv6-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 ipv6-ike-phase1-policy
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ipv6-ike-phase1-policy]
user@host# set mode aggressive
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ipv6-ike-phase1-policy]
user@host# set proposals ipv6-ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ipv6-ike-phase1-policy]
user@host# set pre-shared-key ascii-text 1111111111111111
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@host# set gateway gw-chicago external-interface ge-0/0/15.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gw-chicago]
user@host# set ike-policy ipv6-ike-phase1-policy
```

12. Assign an IP address to the IKE Phase 1 gateway.

```
[edit security ike gateway gw-chicago]
user@host# set address 2001:db8:1::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@host# show security ike
proposal ipv6-ike-phase1-proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
policy ipv6-ike-phase1-policy {
    mode ;
    proposals ipv6-ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway gw-chicago {
    ike-policy ipv6-ike-phase1-policy;
    address 2001:db8:1::1;
```

```
    external-interface ge-0/0/15.0;
}
```

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 ipv6-ipsec-phase2-proposal protocol esp
set security ipsec proposal ipv6-ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipv6-ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipv6-ipsec-phase2-policy proposals ipv6-ipsec-phase2-proposal
set security ipsec policy ipv6-ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ipv6-ike-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ipv6-ike-vpn-chicago ike ipv6-ipsec-policy ipsec-phase2-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 CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@host# set security ipsec proposal ipv6-ipsec-phase2-proposal
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipv6- ipsec-phase2-proposal]
user@host# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipv6-ipsec-phase2-proposal]
user@host# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipv6-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 ipv6-ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipv6-ipsec-phase2-policy]
user@host# set proposals ipv6-ipsec-phase2-proposal
```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

```
[edit security ipsec policy ipv6-ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn ipv6-ike-vpn-chicago ike gateway gw-chicago
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn ipv6-ike-vpn-chicago ike ipsec-policy ipv6-ipsec-phase2-policy
```

### 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 ipv6-ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
policy ipv6-ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipv6-ipsec-phase2-proposal;
}
vpn ipv6-ike-vpn-chicago {
    ike {
        gateway gw-chicago;
        ipsec-policy ipv6-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 Untrust policy ipv6-vpn-tr-untr match source-
address Sunnyvale
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr match destination-
address Chicago
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr match application
```

```
any
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr then permit tunnel
ipsec-vpn ipv6-ike-vpn-chicago
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr then permit tunnel
pair-policy ipv6-vpn-untr-tr
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr match source-
address Chicago
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr match destination-
address Sunnyvale
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr match application
any
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr then permit tunnel
ipsec-vpn ipv6-ike-vpn-chicago
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr then permit tunnel
pair-policy ipv6-vpn-tr-untr
set security policies from-zone Trust to-zone Untrust policy permit-any match source-address any
set security policies from-zone Trust to-zone Untrust policy permit-any match destination-
address any
set security policies from-zone Trust to-zone Untrust policy permit-any match application any
set security policies from-zone Trust to-zone Untrust policy permit-any then permit
insert security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr before policy
permit-any
```

### 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 Untrust zone.

```
[edit security policies from-zone Trust to-zone Untrust]
user@host# set policy ipv6-vpn-tr-untr match source-address Sunnyvale
user@host# set policy ipv6-vpn-tr-untr match destination-address Chicago
user@host# set policy ipv6-vpn-tr-untr match application any
user@host# set policy ipv6-vpn-tr-untr then permit tunnel ipsec-vpn ipv6-ike-vpn-chicago
user@host# set policy ipv6-vpn-tr-untr then permit tunnel pair-policy ipv6-vpn-untr-tr
```

2. Create the security policy to permit traffic from the Untrust zone to the Trust zone.

```
[edit security policies from-zone Untrust to-zone Trust]
user@host# set policy ipv6-vpn-untr-tr match source-address Sunnyvale
user@host# set policy ipv6-vpn-untr-tr match destination-address Chicago
user@host# set policy ipv6-vpn-untr-tr match application any
user@host# set policy ipv6-vpn-untr-tr then permit tunnel ipsec-vpn ipv6-ike-vpn-chicago
user@host# set policy ipv6-vpn-untr-tr then permit tunnel pair-policy ipv6-vpn-tr-untr
```

3. Create the security policy to permit traffic from the Trust zone to the Untrust zone.

```
[edit security policies from-zone Trust to-zone Untrust]
user@host# set policy permit-any match source-address any
user@host# set policy permit-any match destination-address any
user@host# set policy permit-any match application any
user@host# set policy permit-any then permit
```

4. Reorder the security policies so that the vpn-tr-untr security policy is placed above the permit-any security policy.

```
[edit security policies from-zone Trust to-zone Untrust]
user@host# insert policy ipv6-vpn-tr-untr before policy permit-any
```

### 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 Untrust {
    policy ipv6-vpn-tr-untr {
        match {
            source-address Sunnyvale;
            destination-address Chicago;
            application any;
        }
        then {
```

```
            permit {
                tunnel {
                    ipsec-vpn ipv6-ike-vpn-chicago;
                    pair-policy ipv6-vpn-untr-tr;
                }
            }
        }
    }
    policy permit-any {
        match {
            source-address any;
            destination-address any;
            application any;

        }
        then {
            permit
        }
    }
}
from-zone Untrust to-zone Trust {
    policy ipv6-vpn-untr-tr {
        match {
            source-address Chicago;
            destination-address Sunnyvale;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn ipv6-ike-vpn-chicago;
                    pair-policy ipv6-vpn-tr-untr;
                }
            }
        }
    }
}
```

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

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.

## Verification

**IN THIS SECTION**


- Verifying the IKE Phase 1 Status | **400**
- Verifying the IPsec Phase 2 Status | **402**


To confirm that the configuration is working properly, perform these tasks:

### 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 Sunnyvale to a host in Chicago. For policy-based VPNs, a separate host must generate the traffic; traffic initiated from the SRX Series Firewall will not match the VPN policy. We recommend that the test traffic be from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate ping from 2001:db8:3::2/96 to 2001:db8:0::2/96.

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
5       2001:db8:1::1        UP     e48efd6a444853cf  0d09c59aafb720be  Aggressive
```

```
user@host> show security ike security-associations index 5 detail
IKE peer 2001:db8:1::1, Index 5,
  Role: Initiator, State: UP
  Initiator cookie: e48efd6a444853cf, Responder cookie: 0d09c59aafb720be
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 2001:db8:2::1:500, Remote: 2001:db8:1::1:500
```

```
Lifetime: Expires in 19518 seconds
Peer ike-id: not valid
Xauth assigned IP: 0.0.0.0
Algorithms:
 Authentication        : sha1
 Encryption            : aes-128-cbc
 Pseudo random function: hmac-sha1
Traffic statistics:
 Input  bytes  :                 1568
 Output bytes  :                 2748
 Input  packets:                    6
 Output packets:                   23
Flags: Caller notification sent
IPSec security associations: 5 created, 0 deleted
Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Initiator, Message ID: 2900338624
  Local: 2001:db8:2::1:500, Remote: 2001:db8:1::1:500
  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)
  Flags: Caller notification sent, Waiting for done
```

### Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 security associations (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_number`* `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 5 detail` command lists additional information about the security association with an index number of 5:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Initiator and responder role information

  Troubleshooting is best performed on the peer using the responder role.
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

**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    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
```

```
  2     ESP:aes-128/sha1 14caf1d9 3597/ unlim   -   root 500   2001:db8:1::1
  2     ESP:aes-128/sha1 9a4db486 3597/ unlim   -   root 500   2001:db8:1::1
```

```
user@host> show security ipsec security-associations index 2 detail
  Virtual-system: Root
  Local Gateway: 2001:db8:2::1, Remote Gateway: 2001:db8:1::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)
    DF-bit: clear
    Direction: inbound, SPI: 14caf1d9, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: 9a4db486, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 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 ID number is 2. 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 3597/unlim value indicates that the Phase 2 lifetime expires in 3597 seconds, and that no lifesize has been specified, which indicates that the lifetime 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 (up) or D (down) is listed.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 2 detail` command lists the following information:

- The local and remote identities make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common reasons for a Phase 2 failure. For policy-based VPNs, the proxy ID is derived from the security policy. The local and remote addresses are derived from the address book entries, and the service is derived from the application configured for the policy. If Phase 2 fails because of a proxy ID mismatch, you can use the policy to confirm which address book entries are configured. Verify that the addresses match the information being sent. Check the service to ensure that the ports match the information being sent.

  For some third-party vendors, the proxy ID must be manually entered to match.

## SEE ALSO

Internet Key Exchange | 10

## RELATED DOCUMENTATION

IPsec VPN Configuration Overview | 202

CHAPTER 7

# Route Based VPN


Route-Based IPsec VPNs | 406

Route-Based VPN with IKEv2 | 433

Secure Tunnel Interface in a Virtual Router | 495

Dual Stack Tunnels over an External Interface | 507

IPsec VPN Tunnels with Chassis Clusters | 525

Traffic Selectors in Route-Based VPNs | 536

# Route-Based IPsec VPNs

IN THIS SECTION


- Understanding Route-Based IPsec VPNs | 406
- Example: Configuring a Route-Based VPN | 407


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 | **20**

Example: Configuring a Hub-and-Spoke VPN | **221**

Example: Configuring a Policy-Based VPN | **280**


## Example: Configuring a Route-Based VPN

**IN THIS SECTION**


- Requirements | **407**
- Overview | **408**
- Configuration | **411**
- Verification | **426**


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.

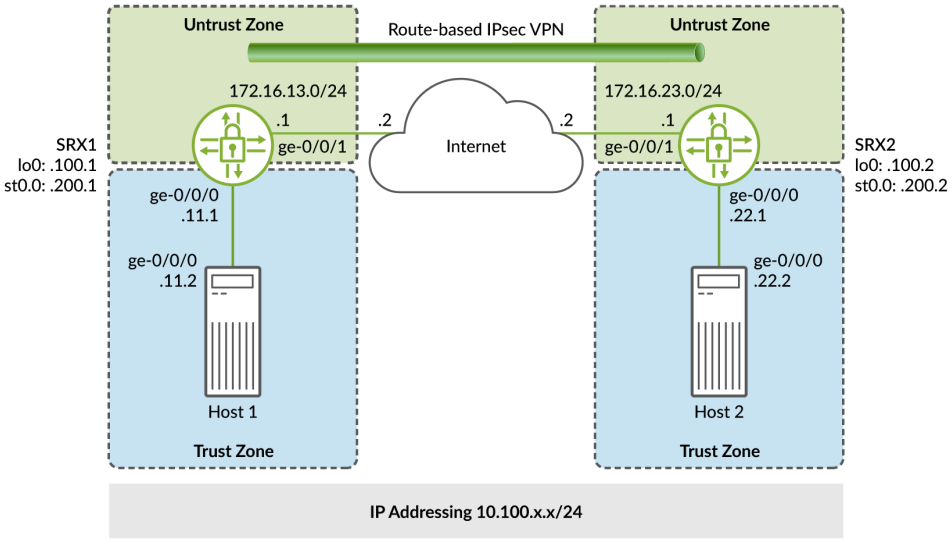
Before you begin, read "IPsec Overview" on page 20.

## 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 32 on page 408 shows an example of a route-based VPN topology.

**Figure 32: 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 44 on page 409 through Table 48 on page 411 for specific configuration parameters used in this example.

**Table 44: 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 |
| | 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<br>0.0.0.0/0 | The next hop is st0.0.<br>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 45: IKE Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | standard | • Authentication method: pre-shared-keys |
| Policy | IKE-POL | • Mode: main<br>• Proposal reference: standard<br>• IKE policy authentication method: pre-shared-keys |

**Table 45: IKE Configuration Parameters *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Gateway | IKE-GW | • IKE policy reference: IKE-POL<br>• External interface: ge-0/0/1<br>• Gateway address: 172.16.23.1 |

**Table 46: 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<br>• IPsec policy reference: IPSEC-POL<br>• Bind to interface: st0.0<br>• establish-tunnels immediately |

**Table 47: 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:<br>  • source-address Host1-Net<br>  • destination-address Host2-Net<br>  • application any<br>• Action: permit |

**Table 47: 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:<br>  • source-address Host2-Net<br>  • destination-address Host1-Net<br>  • application any<br>• Action: permit |

**Table 48: 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.<br><br>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 | **412**
- Configuring IKE | **416**

- Configuring IPsec | **418**
- Configuring Security Policies | **420**
- Configuring TCP-MSS | **424**
- Configuring SRX2 | **425**


**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 | **427**
- Verify the IPsec Status | **429**
- Test Traffic Flow Across the VPN | **431**
- Review Statistics and Errors for an IPsec Security Association | **432**


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 | 20

Example: Configuring a Policy-Based VPN | 280

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 | 434
- Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload | 460
- IKE Policy with a Trusted CA | 493

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 49 on page 433 describes the IPsec Radius xAuth or CP values.

**Table 49: 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 | 88 | Framed pool | NA | NA | Name | Text |
| Standard | 100 | Framed IPv6 pool | NA | NA | Name | Text |
| Vendor | 26 | Primary DNS | 4874 (Juniper ERX) | 4 | IP address | IPv4 address |

Table 49: IPsec Radius xAuth or CP values *(Continued)*

| Radius Attribute | Attribute ID | Attribute Name | Vendor ID (Dictionary) | Vendor Attribute ID | Attribute Value | Type |
|---|---|---|---|---|---|---|
| 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 | 435
- Overview | 435
- Configuration | 439
- Verification | 454


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 20.

## 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 50 on page 435 through Table 54 on page 438 for specific configuration parameters used in this example.

**Table 50: 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.<br>• The ge-0/0/0.0 interface is bound to this zone. |

**Table 50: Interface, Static Route, Security Zone, and Address Book Information *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| | untrust | • IKE is the only allowed system service.<br>• 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.<br>• The address for this address book entry is 192.168.10.0/24. |
| | chicago | • This address is for the untrust zone's address book.<br>• The address for this address book entry is 192.168.168.0/24. |

**Table 51: IKE Phase 1 Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ike-phase1-proposal | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: sha1<br>• Encryption algorithm: aes-128-cbc |

**Table 51: IKE Phase 1 Configuration Parameters *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Policy | ike-phase1-policy | • Mode: main<br>• Proposal reference: ike-phase1-proposal<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |
| Gateway | gw-chicago | • IKE policy reference: ike-phase1-policy<br>• External interface: ge-0/0/3.0<br>• Gateway address: 10.2.2.2 |

**Table 52: IPsec Phase 2 Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipsec-phase2-proposal | • Protocol: esp<br>• Authentication algorithm: hmac-sha1-96<br>• Encryption algorithm: aes-128-cbc |
| Policy | ipsec-phase2-policy | • Proposal reference: ipsec-phase2-proposal<br>• PFS: Diffie-Hellman group2 |
| VPN | ipsec-vpn-chicago | • IKE gateway reference: gw-chicago<br>• IPsec policy reference: ipsec-phase2-policy<br>• Bind to interface: st0.0 |

**Table 53: 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:<br>  • source-address sunnyvale<br>  • destination-address chicago<br>  • application any<br>• Action: permit |
| The security policy permits traffic from the vpn-chicago zone to the trust zone. | vpn-chi-tr | • Match criteria:<br>  • source-address chicago<br>  • destination-address sunnyvale<br>  • application any<br>• Action: permit |

**Table 54: 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.<br><br>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 | 439
- Configuring IKE | 444
- Configuring IPsec | 447
- Configuring Security Policies | 450
- Configuring TCP-MSS | 452
- Configuring the SSG Series Device | 453


### 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 {
    host-inbound-traffic {
        address-book {
            address chicago 192.168.168.0/24;
        }
    }
    interfaces {
        st0.0;
    }
}
```

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

## 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.

## 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.

## 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 | **454**
- Verifying the IPsec Phase 2 Status | **456**
- Reviewing Statistics and Errors for an IPsec Security Association | **458**
- Testing Traffic Flow Across the VPN | **459**


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 | 20

Example: Configuring a Hub-and-Spoke VPN | 221

Example: Configuring a Policy-Based VPN

# Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload

IN THIS SECTION

Requirements | 460

Overview | 460

Configuration | 466

Verification | 487

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

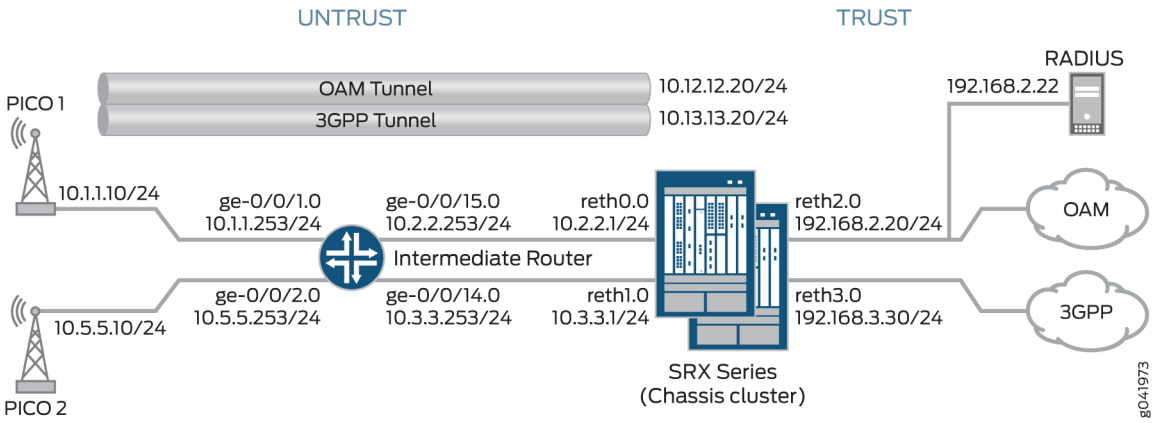
In this example, an SRX Series uses the IKEv2 configuration payload feature to propagate provisioning information to a series of pico cells. The pico cells ship from the factory with a standard configuration

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 33 on page 461 shows a topology in which the SRX Series supports pico cell provisioning using the IKEv2 configuration payload feature.

**Figure 33: 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.

Table 55 on page 463 shows the Phase 1 and Phase 2 options configured on the SRX Series, including information for establishing both OAM and 3GPP tunnels.

**Table 55: 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 55: 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 55: 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 55: 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 | **466**
- Configuring the Intermediate Router | **479**
- Configuring the Pico Cell (Sample Configuration) | **483**
- Configuring the RADIUS Server (Sample Configuration using a FreeRADIUS) | **485**


### 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 | **488**
- Verifying IPsec Security Associations for the SRX Series | **490**


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 | 20

Introduction to PKI in Junos OS | 31

## 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**

Understanding Certificate Authority Profiles | 46

# Secure Tunnel Interface in a Virtual Router

**IN THIS SECTION**


- Understanding Virtual Router Support for Route-Based VPNs | 495
- Example: Configuring an st0 Interface in a Virtual Router | 497


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 | 496


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. In the Junos OS 10.4 release, the support is enabled to place st0 interfaces in a routing

instance, where each unit is configured 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
- Policy-based VPN inside VR

## 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 | 20

## Example: Configuring an st0 Interface in a Virtual Router

**IN THIS SECTION**


- Requirements | 497
- Overview | 497
- Configuration | 500
- Verification | 506


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

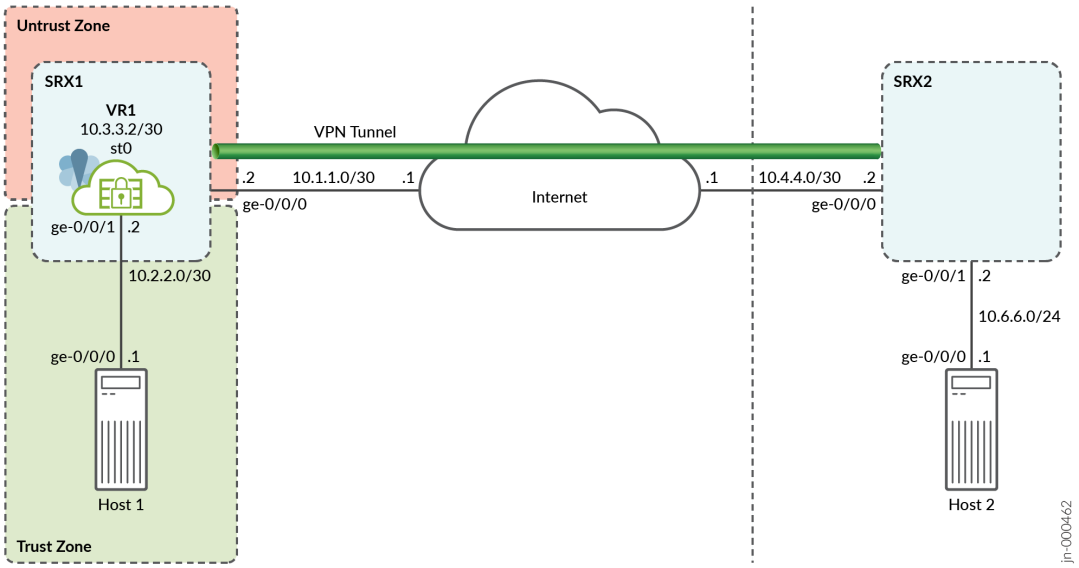
In this example, you perform the following operations:

- 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 34 on page 498 shows the topology used in this example.

**Figure 34: Secure Tunnel Interface in a Virtual Router**




Following tables show the configuration parameters.

**Table 56: 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 |

**Table 56: Interface, Routing Instance, Static Route, and Security Zone Information for SRX1 *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Routing instance (Virtual Router) | VR1 | ge-0/0/1.0<br>st0.0 |
| 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.<br>• The st0.0 interface is bound to this zone. |

**Table 57: IKE Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | first_ikeprop | • Authentication method: pre-shared-keys |
| Policy | first_ikepol | • Mode: main<br>• Proposal reference: first_ikeprop<br>• IKE policy authentication method: pre-shared-keys |
| Gateway | first | • IKE policy reference: first_ikepol<br>• External interface: ge-0/0/0.0<br>• Gateway address: 10.4.4.2 |

**Table 58: IPsec Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | first_ipsecprop | • protocol: esp<br>• authentication-algorithm: hmac-md5-96<br>• encryption-algorithm: 3des-cbc |
| Policy | first_ipsecpol | • IPsec proposal reference: first_ipsecprop |
| VPN | first_vpn | • IKE gateway reference: first<br>• IPsec policy reference: first_ipsecpol<br>• Bind to interface: st0.0<br>• establish-tunnels immediately |

## Configuration

**IN THIS SECTION**


- Procedure | 500


### 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 | **507**

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 | 495

#### RELATED DOCUMENTATION

Route-Based IPsec VPNs | 406

# Dual Stack Tunnels over an External Interface

**IN THIS SECTION**


- Understanding VPN Tunnel Modes | 508
- Example: Configuring Dual-Stack Tunnels over an External Interface | 511


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

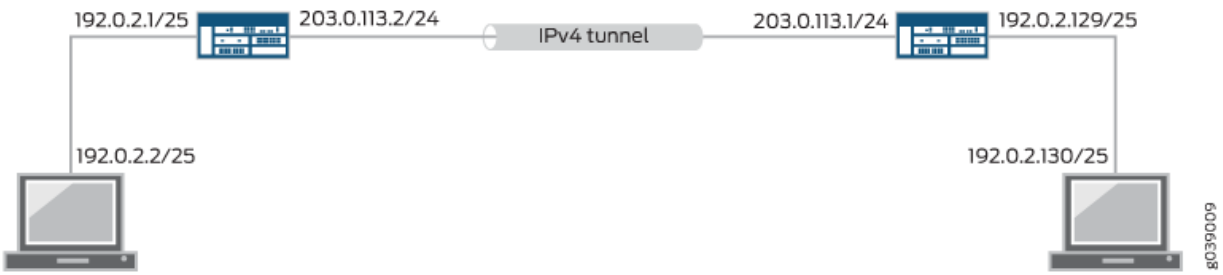
IN THIS SECTION


- Understanding Dual-Stack Tunnels over an External Interface | 510


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. SRX Series Firewalls support four tunnel modes for route-based site-to-site VPNs.

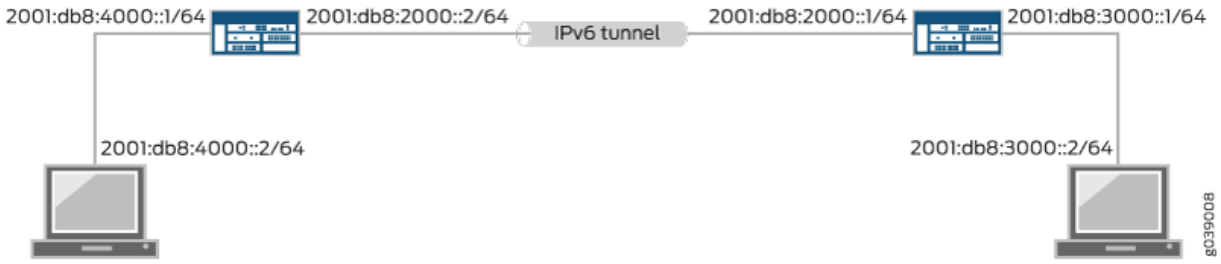
IPv4-in-IPv4 tunnels encapsulate IPv4 packets inside IPv4 packets, as shown in Figure 35 on page 508. The protocol fields for both the outer and the inner headers are IPv4.

**Figure 35: IPv4-in-IPv4 Tunnel**



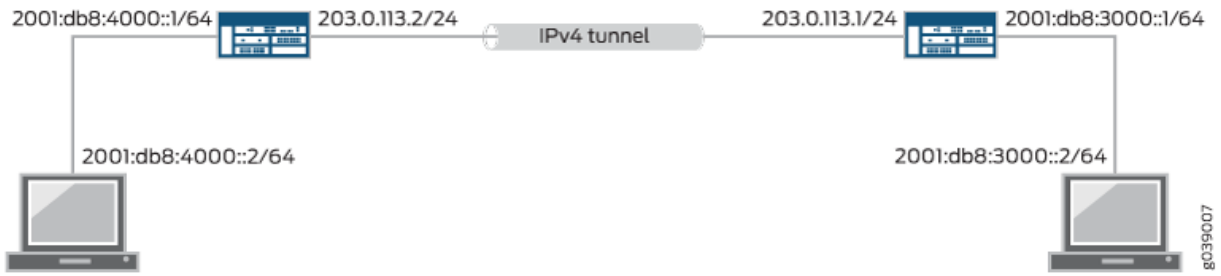

IPv6-in-IPv6 tunnels encapsulate IPv6 packets inside IPv6 packets, as shown in Figure 36 on page 509. The protocol fields for both the outer and inner headers are IPv6.

**Figure 36: IPv6-in-IPv6 Tunnel**



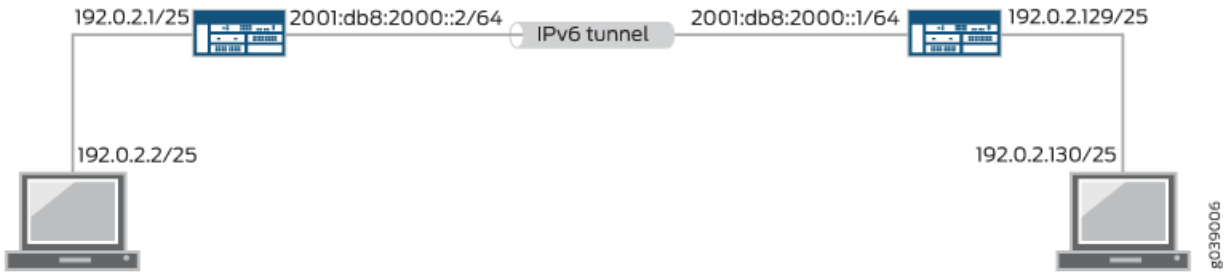

IPv6-in-IPv4 tunnels encapsulate IPv6 packets inside IPv4 packets, as shown in Figure 37 on page 509. The protocol field for the outer header is IPv4 and the protocol field for the inner header is IPv6.

**Figure 37: IPv6-in-IPv4 Tunnel**




IPv4-in-IPv6 tunnels encapsulate IPv4 packets inside IPv6 packets, as shown in Figure 38 on page 509. The protocol field for the outer header is IPv6 and the protocol field for the inner header is IPv4.

**Figure 38: 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`.

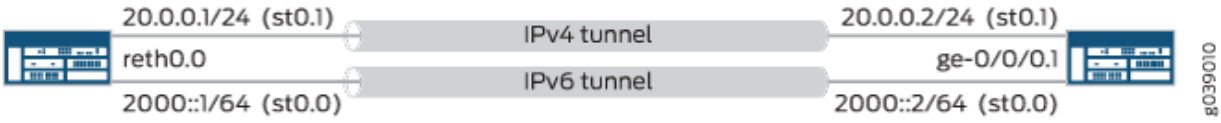
A physical interface configured with both IPv4 and IPv6 addresses can be used as the external interface for parallel IPv4 and IPv6 tunnels to a peer in a route-based site-to-site VPN. This feature is known as *dual-stack tunnels* and requires separate st0 interfaces for each tunnel.

For policy-based VPNs, IPv6-in-IPv6 is the only tunnel mode supported and it is only supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

## 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 39 on page 510, the physical interfaces reth0.0 and ge-0/0/0.1 support parallel IPv4 and IPv6 tunnels between two devices.

**Figure 39: Dual-Stack Tunnels**




In Figure 39 on page 510, 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 SRX Series 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 | 362

Understanding IPv6 IKE and IPsec Packet Processing | 368

VPN Feature Support for IPv6 Addresses | 362

## Example: Configuring Dual-Stack Tunnels over an External Interface

**IN THIS SECTION**

- Requirements | 511
- Overview | 512
- Configuration | 515
- Verification | 521

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 508.

The configuration shown in this example is only supported with route-based site-to-site VPNs.

## Overview

IN THIS SECTION

- Topology | 515

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 59 on page 512 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 59: Phase 1 Options for Dual-Stack Tunnel Configuration**

| Option | Value |
|---|---|
| IKE proposal | ike_proposal |
| Authentication method | Preshared keys |
| Authentication algorithm | MD5 |
| Encryption algorithm | 3DES CBC |
| Lifetime | 3600 seconds |
| IKE policy | ike_policy |

**Table 59: Phase 1 Options for Dual-Stack Tunnel Configuration *(Continued)***

| Option | Value |
|---|---|
| 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 |
| External interface | reth1.0 |

Table 60 on page 514 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 60: 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 |
| 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.

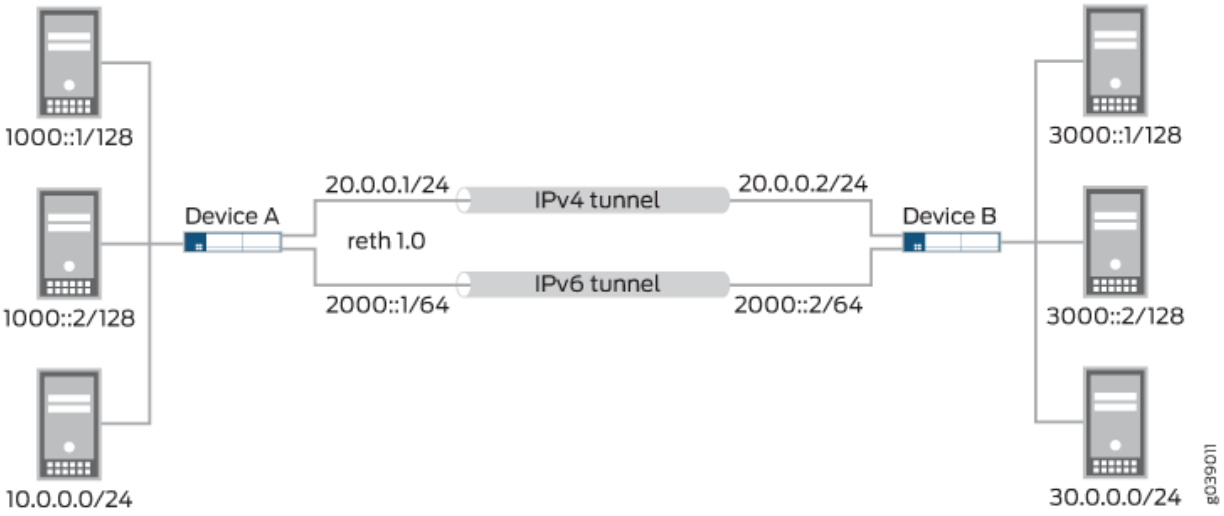
A static route is configured in the default (IPv4) routing table to route IPv4 traffic to 30.0.0.0/24 through st0.1.

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 40 on page 515, the SRX Series Firewall 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 40: Dual-Stack Tunnel Example**




## Configuration

**IN THIS SECTION**

- Procedure | 516

**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 | **522**
- Verifying IPsec Phase 2 Status | **522**
- Verifying Routes | **523**


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 | 361

IPsec VPN Overview | 176

# IPsec VPN Tunnels with Chassis Clusters

IN THIS SECTION

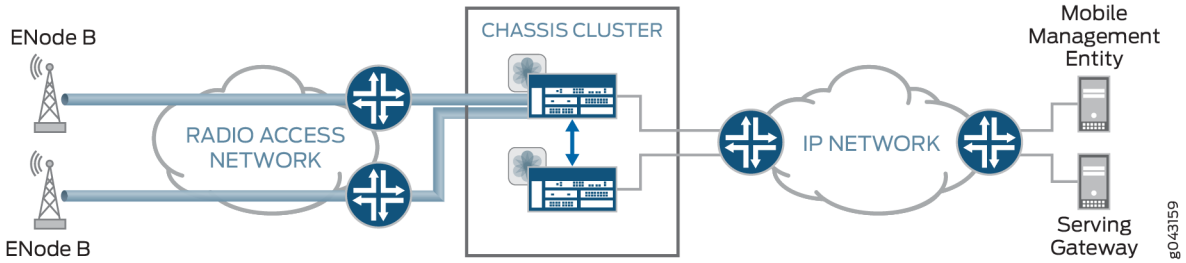
- Understanding Dual Active-Backup IPsec VPN Chassis Clusters | 525
- Example: Configuring Redundancy Groups for Loopback Interfaces | 527

SRX Series Firewall support IPsec VPN tunnels in a chassis cluster setup. In an active/passive chassis cluster, 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

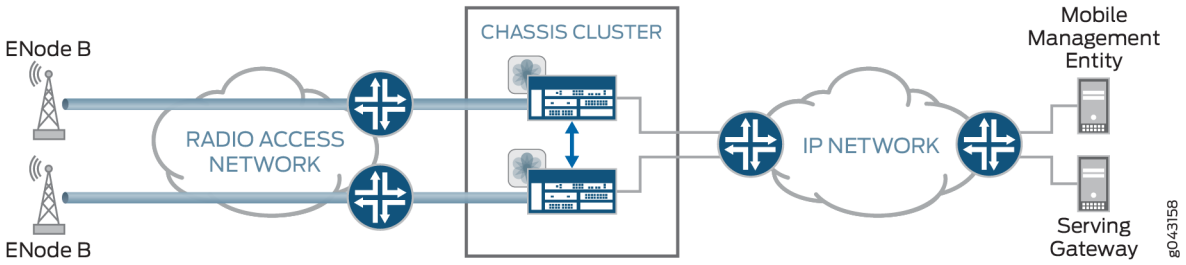
In an active/passive chassis cluster, all VPN tunnels terminate on the same node, as shown in Figure 41 on page 526.

**Figure 41: 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 42 on page 526. This deployment is known as *dual active-backup IPsec VPN chassis clusters*.

**Figure 42: 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 186.
- 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 SRX Series Devices

## Example: Configuring Redundancy Groups for Loopback Interfaces

**IN THIS SECTION**


- Requirements | 528
- Overview | 528
- Configuration | 530
- Verification | 534


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 supported chassis cluster SRX Series Firewall
- An SSG140 device or equivalent
- Two switches
- Junos OS Release 12.1x44-D10 or later for SRX Series Firewall

Before you begin:

Understand chassis cluster redundant Ethernet interfaces. See Chassis Cluster User Guide for SRX Series Devices.

## Overview

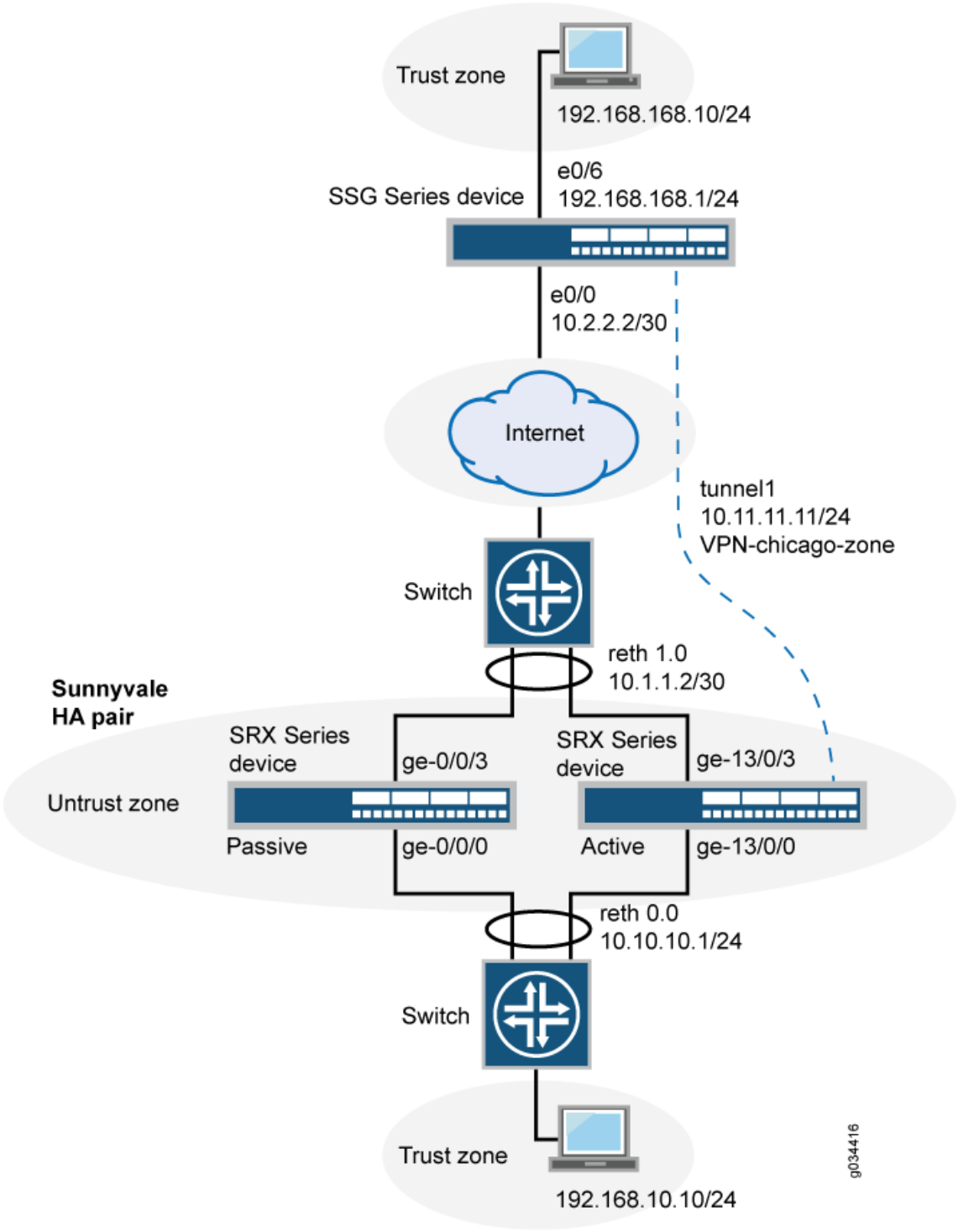
An Internet Key Exchange (IKE) gateway needs an external interface to communicate with a peer device. In a chassis cluster setup, the node on which the external interface is active selects a Services Processing Unit (SPU) to support the VPN tunnel. IKE and IPsec packets are processed on that SPU. Therefore, 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 43 on page 529 shows an example of a loopback chassis cluster VPN topology. In this topology, the SRX Series Firewall chassis cluster device is located in Sunnyvale, California. The SRX Series Firewall 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 SRX chassis cluster and it helps to build a VPN tunnel.

**Figure 43: Loopback Interface for Chassis Cluster VPN**




| | |
|---|---|
| lo0.1<br>10.3.3.3/30<br>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<br>10.11.11.10/24<br>vpn-chicago-zone | A logical interface on the secure tunnel interface for IPSec VPN tunnel. |

## Configuration

**IN THIS SECTION**

- Procedure | 530

### 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 | 534


### 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 | **1349**

RELATED DOCUMENTATION

Chassis Cluster User Guide for SRX Series Devices

# Traffic Selectors in Route-Based VPNs

IN THIS SECTION


- Understanding Traffic Selectors in Route-Based VPNs | 536
- Example: Configuring Traffic Selectors in a Route-Based VPN | 543


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).

## Understanding Traffic Selectors in Route-Based VPNs

IN THIS SECTION


- Traffic Selector Configuration | 537
- Understanding Auto Route Insertion | 538
- Understanding Traffic Selectors and Overlapping IP Addresses | 538


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.

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. Starting with Junos OS Release 15.1X49-D100, traffic selectors can be configured with IKEv2 site-to-site VPNs.

## 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

  Starting with Junos OS Release 15.1X49-D140, on all SRX Series Firewalls and vSRX Virtual Firewall instances, 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.
- Point-to-multipoint interfaces
- Dynamic routing protocols configured on st0 interfaces

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.

From Junos OS Release 21.1R1 onwards, 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.

## Understanding 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. There is no configuration of the metric for the static ARI route.

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.

## Understanding 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 | 508

## Example: Configuring Traffic Selectors in a Route-Based VPN

**IN THIS SECTION**


- Requirements | 543
- Overview | 544
- Configuration | 545
- Verification | 559


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 536.

- Install the IKE package.

```
user@host> request system software add optional://junos-ike.tgz
```

## Overview

IN THIS SECTION

- Topology | 544

This example configures traffic selectors to allow traffic to flow between subnetworks on SRX_A and subnetworks on SRX_B.

Table 61 on page 544 shows the traffic selectors for this example. Traffic selectors are configured under Phase 2 options.

**Table 61: Traffic Selector Configurations**

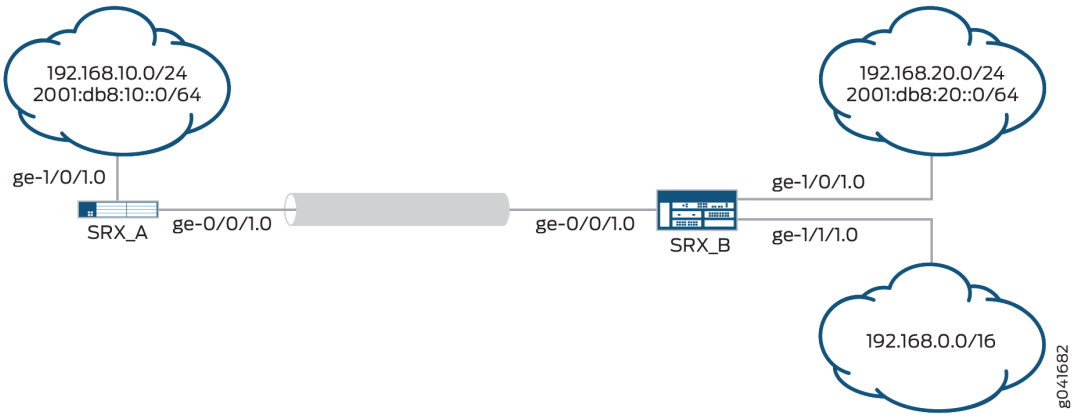
| SRX_A | | | SRX_B | | |
|---|---|---|---|---|---|
| Traffic Selector Name | Local IP | Remote IP | Traffic Selector Name | Local IP | Remote IP |
| TS1-ipv6 | 2001:db8:10::0/64 | 2001:db8:20::0/64 | TS1-ipv6 | 2001:db8:20::0/64 | 2001:db8:10::0/64 |
| TS2-ipv4 | 192.168.10.0/24 | 192.168.0.0/16 | TS2-ipv4 | 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 44 on page 545, 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 44: Traffic Selector Configuration Example**




## Configuration

**IN THIS SECTION**


- Configuring SRX_A | 545
- Configuring SRX_B | 552


### 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 | **560**
- Verifying Traffic Selectors | **562**
- Verifying Routes | **562**


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 | 508

**Release History Table**

| Release | Description |
|---|---|
| 15.1X49-D140 | Starting with Junos OS Release 15.1X49-D140, on all SRX Series Firewalls and vSRX Virtual Firewall instances, 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 | 406

8
CHAPTER

# Class-Of-Service Based VPN

CoS-Based IPsec VPNs | 565

# CoS-Based IPsec VPNs

IN THIS SECTION


- Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs | **565**
- Understanding Traffic Selectors and CoS-Based IPsec VPNs | **568**
- Example: Configuring CoS-Based IPsec VPNs | **571**
- Understanding CoS Support on st0 Interfaces | **600**


You can configure Junos class-of-service (CoS) features to 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.

## Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs

IN THIS SECTION


- Benefits of CoS-Based IPsec VPNs with Multiple IPsec SAs | **566**
- Overview | **566**
- Mapping FCs to IPsec SAs | **566**
- IPsec SA Negotiation | **566**
- Rekey | **567**
- Adding or Deleting FCs from a VPN | **567**
- Dead Peer Detection (DPD) | **567**
- Commands | **568**
- Supported VPN Features | **568**

Class of service (CoS) forwarding classes (FCs) configured on the SRX Series Firewall 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.

## Benefits of CoS-Based IPsec VPNs with Multiple IPsec SAs

- 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.

## Overview

This feature is proprietary to Juniper Networks and works with supported SRX platforms and Junos OS releases. The VPN peer device must be an SRX Series Firewall or vSRX Virtual Firewall instance that supports this feature or any other product that support the same functionality in the same way as SRX Series Firewall.

## Mapping FCs to IPsec SAs

Up to 8 forwarding classes (FC) can be configured for a VPN with the `multi-sa forwarding-classes` 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 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 SRX Series Firewalls or vSRX Virtual Firewall instances.

## IPsec SA Negotiation

When multiple FCs are configured 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 may 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 installed 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 may 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 may 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 SRX Series Firewall 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 SRX Series Firewall 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 Multi SAs with Differentiated Services Code Point (DSCP) traffic steering with traffic selectors, the following behavior occurs during rekey. When the traffic selectors performs 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 expire. In this case, traffic that use to match the specific SA is sent through the default traffic selector instead.

### Adding or Deleting 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. Policy-based VPNs are not supported.
- AutoVPN.
- Traffic selectors.
- Auto Discovery VPNs (ADVPNs).
- IKEv2. IKEv1 is not supported.
- Dead peer detection (DPD). VPN monitoring is not supported.

**SEE ALSO**

Understanding Traffic Selectors and CoS-Based IPsec VPNs

Example: Configuring CoS-Based IPsec VPNs

*Forwarding Classes Overview*

## Understanding Traffic Selectors and CoS-Based IPsec VPNs

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 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 SRX Series Firewall:

```
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:

```
ipsec {
    vpn vpn1 {
       ts1 {
           local-ip 3.3.3.0/24;
           remote-ip 4.4.4.0/24;
              }

    multi-sa {
        forwarding-class web-data;
           forwarding-class voip-data
           forwarding-class control-data;
         }
       }
   }
```

In the configuration above, 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.

In the second scenario, VPN vpn1 is configured with two traffic selectors ts1 and ts2 and the three FCs:

```
ipsec {
    vpn vpn1 {
       ts1 {
           local-ip 3.3.3.0/24;
           remote-ip 4.4.4.0/24;
              }
  ts2 {
           local-ip 6.6.6.0/24;
           remote-ip 7.7.7.0/24;
         }
```

```
    multi-sa {
        forwarding-class web-data;
           forwarding-class voip-data
           forwarding-class control-data;
         }
       }
   }
```

In the configuration above, 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.

In the third scenario, traffic selectors ts1 and ts2 support different sets of FCs. The traffic selectors need to be configured for different VPNs:

```
ipsec {
    vpn vpn1 {
        bind-interface st0.0;
        ts1 {
            local-ip 3.3.3.0/24;
            remote-ip 4.4.4.0/24;
            }

        multi-sa {
         forwarding-class web-data;
             forwarding-class voip-data;
           forwarding-class control-data;
                }
     vpn vpn2 {
         bind-interface st0.0;
         ts2 {
             local-ip 6.6.6.0/24;
             remote-ip 7.7.7.0/24;
             }
       multi-sa {
         forwarding-class web-data;
             forwarding-class voip-data;
             }
         }
```

In the configuration above, 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.

SEE ALSO

Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs

Example: Configuring CoS-Based IPsec VPNs

# Example: Configuring CoS-Based IPsec VPNs

IN THIS SECTION


- Requirements | 571
- Overview | 571
- Configuration | 575
- Verification | 596


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 SRX platforms and Junos OS releases. The VPN peer device must be an SRX Series Firewall or vSRX Virtual Firewall instance that supports this feature.

## Requirements

This example uses the following hardware:

- Any SRX Series Firewall

Before you begin:

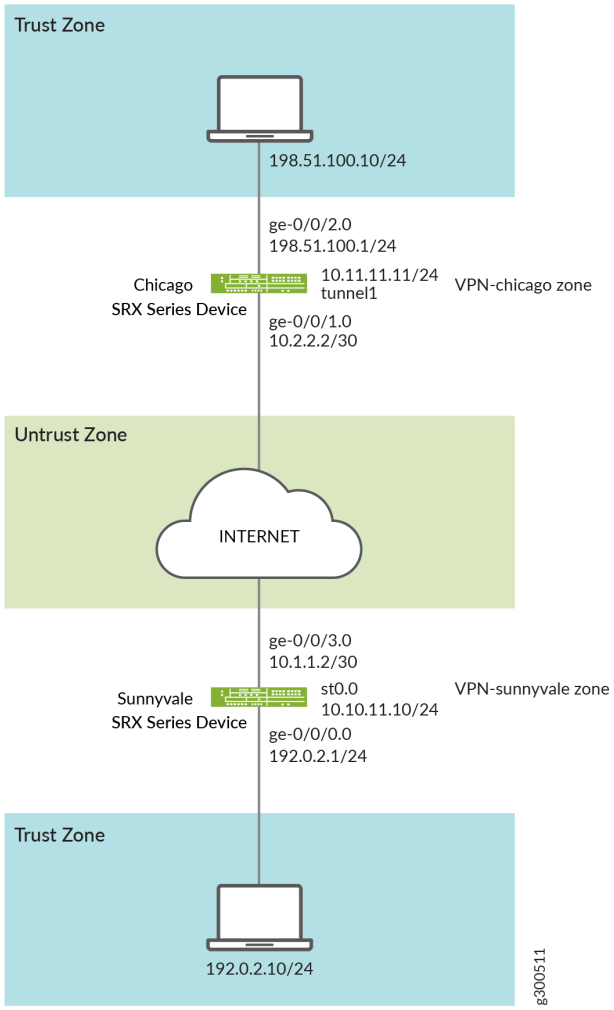
- Understand how Class of service (CoS) forwarding classes (FCs) configured on the SRX Series Firewall can be mapped to IPsec security associations (SAs). See Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs.
- Understand Traffic Selectors and CoS-Based IPsec VPNs. See Understanding Traffic 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 45 on page 572 shows an example of an IPsec route-based VPN topology. In this topology, one SRX Series Firewall is located in Sunnyvale, and one SRX Series Firewall is located in Chicago.

**Figure 45: 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 62 on page 573 through Table 65 on page 575.

**Table 62: 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.<br>• The ge-0/0/0.0 interface is bound to this zone. |
| | untrust | • All system services are allowed.<br>• The ge-0/0/3.0 interface is bound to this zone. |
| | vpn | The st0.0 interface is bound to this zone. |

**Table 63: IKE Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ike-proposal | • Authentication method: rsa-signatures<br>• Diffie-Hellman group: group14<br>• Authentication algorithm: sha-256<br>• Encryption algorithm: aes-256-cbc |

**Table 63: IKE Configuration Parameters *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Policy | ike-policy | • Mode: main<br>• Proposal reference: ike-proposal<br>• IKE policy authentication method: rsa-signatures |
| Gateway | gw-sunnyvale | • IKE policy reference: ike-policy<br>• External interface: ge-0/0/3.0<br>• Gateway address: 10.2.2.2 |

**Table 64: IPsec Configuration Parameters**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipsec_prop | • Protocol: esp<br>• Authentication algorithm: hmac-sha-256<br>• Encryption algorithm: aes-256-cbc |
| Policy | ipsec_pol | • Proposal reference: ipsec_prop |
| VPN | ipsec_vpn1 | • IKE gateway reference: gw-chicago<br>• IPsec policy reference: ipsec_pol |

**Table 65: Security Policy Configuration Parameters**

| Purpose | Name | Configuration Parameters |
|---|---|---|
| The security policy permits traffic from the trust zone to the vpn zone. | vpn | • Match criteria:<br>  • source-address sunnyvale<br>  • destination-address chicago<br>  • application any<br>• Action: permit |
| The security policy permits traffic from the vpn zone to the trust zone. | vpn | • Match criteria:<br>  • source-address chicago<br>  • destination-address sunnyvale<br>  • application any<br>• Action: permit |

## Configuration

**IN THIS SECTION**


- Configuring Basic Network and Security Zone Information | **576**
- Configuring CoS | **580**
- Configuring IKE | **587**
- Configuring IPsec | **590**
- Configuring Security Policies | **594**

## 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 *Using the CLI Editor in Configuration Mode* in the 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 *Using the CLI Editor in Configuration Mode* in the 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.

## 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 *Using the CLI Editor in Configuration Mode* in the 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 *Using the CLI Editor in Configuration Mode* in the 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 *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 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 | 596

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**

Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs

Understanding Traffic Selectors and CoS-Based IPsec VPNs

IPsec Overview | **20**

Example: Configuring a Policy-Based VPN | **280**

Introduction to IKE in Junos OS | **136**

## Understanding CoS Support on st0 Interfaces

**IN THIS SECTION**

- Limitations of CoS support on VPN st0 interfaces | **601**

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.

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 st0 interface on all available SRX Series Firewalls and vSRX2.0:

- Classifiers
- Policers

- Queuing, scheduling, and shaping
- Rewrite markers
- Virtual channels

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. Starting with Junos OS Release 17.4R1, support for listed CoS features is added for the st0 interface for SRX4600 devices.

## 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 66 on page 601 describes the st0 CoS feature support for different types of VPNs.

**Table 66: CoS Feature Support for VPN**

| Classifier Features | Site-to-Site VPN (P2P) | AutoVPN (P2P) | Site-to-Site/Auto VPN /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 |

- On SRX300, SRX320, SRX340, SRX345, and SRX550HM devices, one st0 logical interface can bind to multiple VPN tunnels. The eight queues for the st0 logical interface cannot reroute the traffic to different tunnels, so pre-tunneling is not supported.

  The virtual channel feature can be used as a workaround on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

- When defining a CoS shaping rate on an 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.

- On SRX550M, SRX5400, SRX5600, and SRX5800 devices, bandwidth limit and burst size limit values in a policer configuration are a per-SPU, not per-system limitation. This is the same policer behavior as on the physical interface.

**SEE ALSO**

Class of Service User Guide (Security Devices)

**Release History Table**

| Release | Description |
|---|---|
| 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. |
| 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)

9

CHAPTER

# NAT-T

Route-Based and Policy-Based VPNs with NAT-T | 605

# Route-Based and Policy-Based VPNs with NAT-T

IN THIS SECTION


- Understanding NAT-T | 605
- Example: Configuring a Route-Based VPN with the Responder behind a NAT Device | 606
- Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device | 643
- Example: Configuring NAT-T with Dynamic Endpoint VPN | 690


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.

## 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.

On SRX5400, SRX5600, and SRX5800 devices, the IPsec NAT-T tunnel scaling and sustaining issues are as follows:

- For a given private IP address, the NAT device should 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.

Starting from Junos OS Release 19.2R1, PowerMode IPSec (PMI) for NAT-T is supported only on SRX5400, SRX5600, and SRX5800 devices equipped with SRX5K-SPC3 Services Processing Card (SPC), or with vSRX Virtual Firewall.

**SEE ALSO**

IPsec Overview | 20

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

**IN THIS SECTION**

Requirements | 607

- Overview | **607**
- Configuration | **613**
- Verification | **632**


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

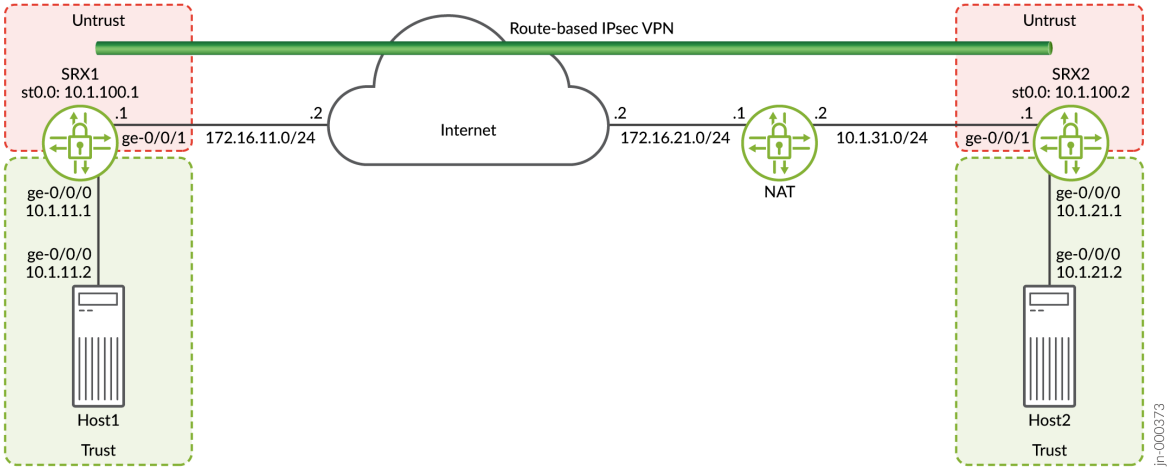
Before you begin, read "IPsec Overview" on page 20.

## Overview

In this example, you configure a route-based VPN. Host1 will use the VPN to connect to their corporate headquarters on SRX2.

Figure 46 on page 607 shows an example of a topology for route-based VPN with only the responder behind a NAT device.

**Figure 46: 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 67 on page 608 through Table 69 on page 610 for specific configuration parameters used for the initiator in the examples.

**Table 67: 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.<br>• The ge-0/0/1.0 and the st0.0 interfaces are bound to this zone. |
| | trust | • Allow all system services.<br>• Allow all protocols.<br>• 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. |

**Table 67: Interface, Routing Options, and Security Parameters for SRX1 *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| | 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 68: IKE Phase 1 Configuration Parameters for SRX1**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ike_prop | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: sha1<br>• Encryption algorithm: 3des-cbc |
| Policy | ike_pol | • Mode: main<br>• Proposal reference: ike_prop<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |
| Gateway | gw1 | • IKE policy reference: ike_pol<br>• External interface: ge-0/0/1.0<br>• Gateway address: 172.16.21.1<br>• Local peer (initiator): branch_natt1@example.net<br>• Remote peer (responder): responder_natt1@example.net |

**Table 69: IPsec Phase 2 Configuration Parameters for SRX1**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipsec_prop | • Protocol: esp<br>• Authentication algorithm: hmac-sha1-96<br>• Encryption algorithm: 3des-cbc |
| Policy | ipsec_pol | • Proposal reference: ipsec_prop<br>• Perfect forward secrecy (PFS) keys: group2 |
| VPN | vpn1 | • IKE gateway reference: gw1<br>• IPsec policy reference: ipsec_pol<br>• Bind to interface: st0.0<br>• Establish tunnels immediately |

See Table 70 on page 610 through Table 72 on page 612 for specific configuration parameters used for the responder in the examples.

**Table 70: 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. |

**Table 70: Interface, Routing Options, and Security Parameters for SRX2 *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Security zones | untrust | • Allow IKE and ping system services.<br>• The ge-0/0/1.0 and the st0.0 interfaces are bound to this zone. |
| | trust | • Allow all system services.<br>Allow all protocols.<br>• 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 71: IKE Phase 1 Configuration Parameters for SRX2**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ike_prop | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: sha1<br>• Encryption algorithm: 3des-cbc |

**Table 71: IKE Phase 1 Configuration Parameters for SRX2 *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Policy | ike_pol | • Mode: main<br>• Proposal reference: ike_prop<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |
| Gateway | gw1 | • IKE policy reference: ike_pol<br>• External interface: ge-0/0/1.0<br>• Gateway address: 172.16.11.1<br>• Local peer (responder): responder_natt1@example.net<br>• Remote peer (initiator): branch_natt1@example.net |

**Table 72: IPsec Phase 2 Configuration Parameters for SRX2**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipsec_prop | • Protocol: esp<br>• Authentication algorithm: hmac-sha1-96<br>• Encryption algorithm: 3des-cbc |
| Policy | ipsec_pol | • Proposal reference: ipsec_prop<br>• PFS keys: group2 |
| VPN | vpn1 | • IKE gateway reference: gw1<br>• IPsec policy reference: ipsec_pol<br>• Bind to interface: st0.0<br>• Establish tunnels immediately |

## Configuration

**IN THIS SECTION**


- Configuring Interface, Routing Options, and Security Parameters for SRX1 | **613**
- Configuring IKE for SRX1 | **618**
- Configuring IPsec for SRX1 | **620**
- Configuring Interfaces, Routing Options, and Security Parameters for SRX2 | **622**
- Configuring IKE for SRX2 | **628**
- Configuring IPsec for SRX2 | **630**
- Configuration for the NAT Device | **632**


### 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 | **633**
- Verifying IPsec Security Associations on SRX1 | **635**

- Verifying the IKE Phase 1 Status on SRX2 | **637**
- Verifying IPsec Security Associations on SRX2 | **639**
- Verifying Host-to-Host Reachability | **642**


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 | 20

Example: Configuring a Policy-Based VPN

# Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device

IN THIS SECTION


- Requirements | 643
- Overview | 643
- Configuration | 650
- Verification | 680


This example shows how to configure a policy-based VPN with both an initiator and a responder behind a NAT device to allow data to be securely transferred between a branch office and the corporate office.

## Requirements

Before you begin, read "IPsec Overview" on page 20.

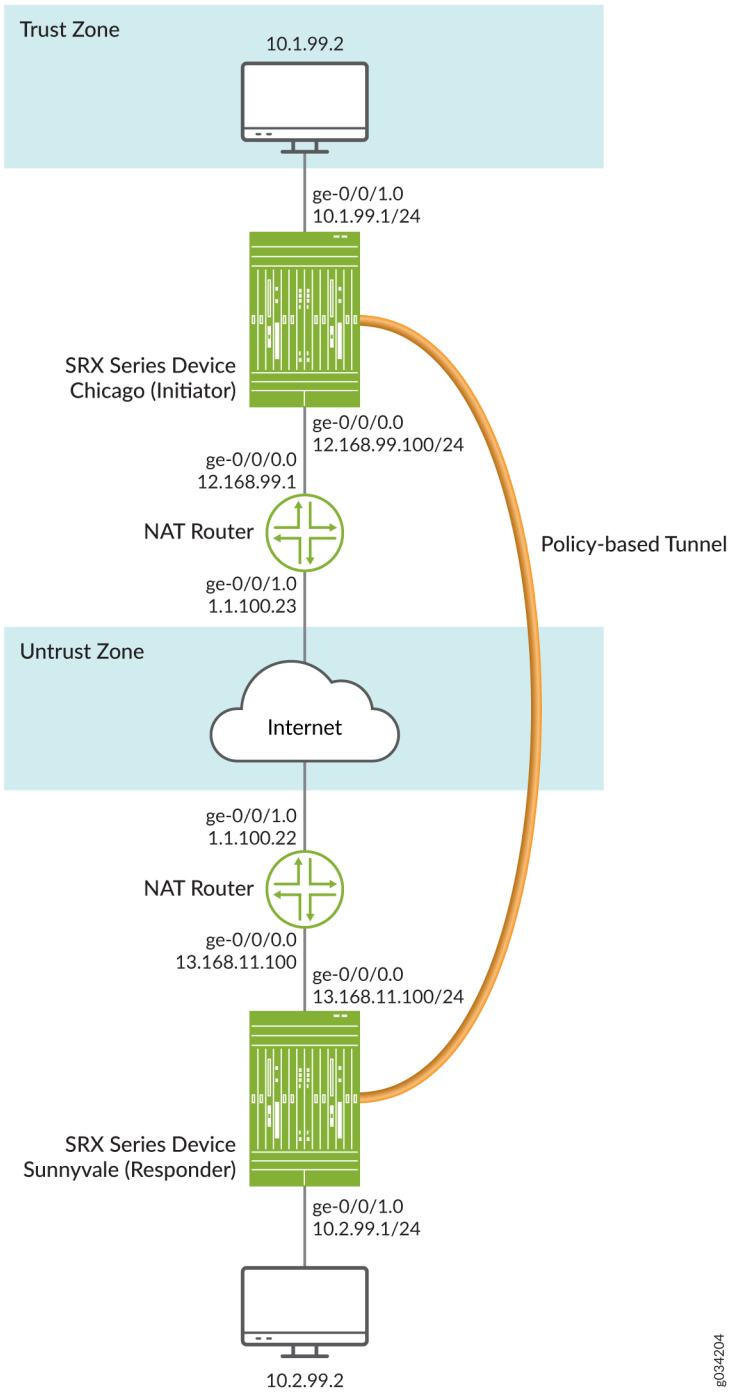
## Overview

In this example, you configure a policy-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 branch office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

In this example, you configure interfaces, routing options, security zones, security policies for both an initiator and a responder.

Figure 47 on page 644 shows an example of a topology for a VPN with both an initiator and a responder behind a static NAT device.

**Figure 47: Policy-Based VPN Topology with Both an Initiator and a Responder Behind a NAT Device**




In this example, you configure interfaces, an IPv4 default route, and security zones. Then you configure IKE Phase 1, including local and remote peers, IPsec Phase 2, and the security policy. Note in the

example above, the responder's private IP address 13.168.11.1 is hidden by the static NAT device and mapped to public IP address 1.1.100.1.

See Table 73 on page 645 through Table 76 on page 647 for specific configuration parameters used for the initiator in the examples.

**Table 73: Interface, Routing Options, and Security Zones for the Initiator**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Interfaces | ge-0/0/0 | 12.168.99.100/24 |
| | ge-0/0/1 | 10.1.99.1/24 |
| Static routes | 10.2.99.0/24 (default route) | The next hop is 12.168.99.100. |
| | 1.1.100.0/24 | 12.168.99.100 |
| Security zones | trust | • All system services are allowed.<br>• All protocols are allowed.<br>• The ge-0/0/1.0 interface is bound to this zone. |
| | untrust | • The ge-0/0/0.0 interface is bound to this zone. |

**Table 74: IKE Phase 1 Configuration Parameters for the Initiator**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ike_prop | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: md5<br>• Encryption algorithm: 3des-cbc |

**Table 74: IKE Phase 1 Configuration Parameters for the Initiator *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Policy | ike_pol | • Mode: main<br>• Proposal reference: ike_prop<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |
| Gateway | gate | • IKE policy reference: ike_pol<br>• External interface: ge-0/0/1.0<br>• Gateway address: 1.1.100.23<br>• Local peer is hostname chicago<br>• Remote peer is hostname sunnyvale |

**Table 75: IPsec Phase 2 Configuration Parameters for the Initiator**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipsec_prop | • Protocol: esp<br>• Authentication algorithm: hmac-md5-96<br>• Encryption algorithm: 3des-cbc |
| Policy | ipsec_pol | • Proposal reference: ipsec_prop<br>• Perfect forward secrecy (PFS): group1 |
| VPN | first_vpn | • IKE gateway reference: gate<br>• IPsec policy reference: ipsec_pol |

**Table 76: Security Policy Configuration Parameters for the Initiator**

| Purpose | Name | Configuration Parameters |
|---|---|---|
| The security policy permits tunnel traffic from the trust zone to the untrust zone. | pol1 | • Match criteria:<br>  • source-address any<br>  • destination-address any<br>  • application any<br>• Action: permit tunnel ipsec-vpn first_vpn |
| The security policy permits tunnel traffic from the untrust zone to the trust zone. | pol1 | • Match criteria:<br>  • application any<br>• Action: permit tunnel ipsec-vpn first_vpn |

See Table 77 on page 647 through Table 80 on page 650 for specific configuration parameters used for the responder in the examples.

**Table 77: Interface, Routing Options, and Security Zones for the Responder**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Interfaces | ge-0/0/0 | 13.168.11.100/24 |
|  | ge-0/0/1 | 10.2.99.1/24 |
| Static routes | 10.1.99.0/24 (default route) | The next hop is 13.168.11.100 |
|  | 1.1.100.0/24 | 13.168.11.100 |

**Table 77: Interface, Routing Options, and Security Zones for the Responder *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Security zones | trust | • All system services are allowed.<br>• All protocols are allowed.<br>• The ge-0/0/1.0 interface is bound to this zone. |
| | untrust | • The ge-0/0/0.0 interface is bound to this zone. |

**Table 78: IKE Phase 1 Configuration Parameters for the Responder**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ike_prop | • Authentication method: pre-shared-keys<br>• Diffie-Hellman group: group2<br>• Authentication algorithm: md5<br>• Encryption algorithm: 3des-cbc |
| Policy | ike_pol | • Mode: main<br>• Proposal reference: ike_prop<br>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text |

**Table 78: IKE Phase 1 Configuration Parameters for the Responder *(Continued)***

| Feature | Name | Configuration Parameters |
|---|---|---|
| Gateway | gate | • IKE policy reference: ike_pol<br>• External interface: ge-0/0/1.0<br>• Gateway address: 1.1.100.22<br>• Always send dead-peer detection<br>• Local peer is hostname sunnyvale<br>• Remote peer is hostname chicago |

**Table 79: IPsec Phase 2 Configuration Parameters for the Responder**

| Feature | Name | Configuration Parameters |
|---|---|---|
| Proposal | ipsec_prop | • Protocol: esp<br>• Authentication algorithm: hmac-md5-96<br>• Encryption algorithm: 3des-cbc |
| Policy | ipsec_pol | • Proposal reference: ipsec_prop<br>• Perfect forward secrecy (PFS): group1 |
| VPN | first_vpn | • IKE gateway reference: gate<br>• IPsec policy reference: ipsec_pol |

**Table 80: Security Policy Configuration Parameters for the Responder**

| Purpose | Name | Configuration Parameters |
|---|---|---|
| The security policy permits tunnel traffic from the trust zone to the untrust zone. | pol1 | • Match criteria:<br>  • source-address any<br>  • destination-address any<br>  • application any<br>• Action: permit tunnel ipsec-vpn first_vpn |
| The security policy permits tunnel traffic from the untrust zone to the trust zone. | pol1 | • Match criteria:<br>  • application any<br>• Action: permit tunnel ipsec-vpn first_vpn |

## Configuration

**IN THIS SECTION**


- Configuring Interface, Routing Options, and Security Zones for the Initiator | **651**
- Configuring IKE for the Initiator | **654**
- Configuring IPsec for the Initiator | **657**
- Configuring Security Policies for the Initiator | **659**
- Configuring NAT for the Initiator | **661**
- Configuring Interface, Routing Options, and Security Zones for the Responder | **666**
- Configuring IKE for the Responder | **669**
- Configuring IPsec for the Responder | **672**
- Configuring Security Policies for the Responder | **674**
- Configuring NAT for the Responder | **676**

### Configuring Interface, Routing Options, and Security Zones for the 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.

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 12.168.99.100/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.99.1/24
set routing-options static route 10.2.99.0/24 next-hop 12.168.99.1
set routing-options static route 1.1.100.0/24 next-hop 12.168.99.1
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-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/0.0
```

#### 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 zones:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 12.168.99.100/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.1.99.1/24
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 10.2.99.0/24 next-hop 12.168.99.1
user@host# set routing-options static route 1.1.100.0/24 next-hop 12.168.99.1
```

3. Configure the trust security zone.

```
[edit ]
user@host# set security zones security-zone trust host-inbound-traffic protocols all
```

4. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/1.0
```

5. Specify system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

6. Assign an interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/0.0
```

### 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 instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 12.168.99.100/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
```

```
            family inet {
                address 10.1.99.1/24;
            }
        }
    }
```

```
[edit]
user@host# show routing-options
    static {
        route 10.2.99.0/24 next-hop 12.168.99.1;
        route 1.1.100.0/24 next-hop 12.168.99.1;
    }
```

```
[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 {
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
```

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

### Configuring IKE for the 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 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 md5
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode aggressive
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 gate ike-policy ike_pol
set security ike gateway gate address 13.168.11.100
set security ike gateway gate external-interface ge-0/0/0.0
set security ike gateway gate local-identity hostname chicago
```

## 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# edit proposal ike_prop
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike_prop]
user@host# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike_prop]
user@host# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike_prop]
user@host# set authentication-algorithm md5
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike_prop]
user@host# set encryption-algorithm 3des-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike policy ]
user@host# edit policy ike_pol
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike_pol]
user@host# set mode aggressive
```

8. Specify a reference to the IKE proposal.

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

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike_pol pre-shared-key]
user@host# set ascii-text "$ABC123”
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike ]
user@host# set gateway gate external-interface ge-0/0/0.0
```

11. Create an IKE Phase 1 gateway address.

```
[edit security ike gateway gate]
set address 13.168.11.100
```

12. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gate]
set ike-policy ike_pol
```

13. Set `local-identity` for the local peer.

```
[edit security ike gateway gate]
user@host# set local-identity hostname chicago
```

### 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@host# show security ike
    proposal ike_prop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm 3des-cbc;
    }
    policy ike_pol {
        mode aggressive;
        proposals ike_prop;
```

```
        pre-shared-key ascii-text "$ABC123”
    }
    gateway gate {
        ike-policy ike_pol;
        address 13.168.11.100;
        local-identity hostname chicago;
        external-interface ge-0/0/0.0;
    }
```

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

#### Configuring IPsec for the 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 security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group1
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn first_vpn ike gateway gate
set security ipsec vpn first_vpn ike ipsec-policy ipsec_pol
set security ipsec vpn first_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 CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@host# edit security ipsec proposal ipsec_prop
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec_prop]
user@host# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set authentication-algorithm hmac-md5-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set encryption-algorithm 3des-cbc
```

5. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec_pol]
user@host# set proposals ipsec_prop
```

6. Specify IPsec Phase 2 to use perfect forward secrecy (PFS) group1.

```
[edit security ipsec policy ipsec_pol ]
user@host# set perfect-forward-secrecy keys group1
```

7. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn first_vpn ike gateway gate
```

8. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn first_vpn ike ipsec-policy ipsec_pol
```

### 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@host# show security ipsec
    proposal ipsec_prop {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm 3des-cbc;
    }
    policy ipsec_pol {
        perfect-forward-secrecy {
            keys group1;
        }
        proposals ipsec_prop;
    }
    vpn first_vpn {
        ike {
            gateway gate;
            ipsec-policy ipsec_pol;
        }
        establish-tunnels immediately;
    }
```

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

#### Configuring Security Policies for the 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 security policies from-zone trust to-zone untrust policy pol1 match source-address any
set security policies from-zone trust to-zone untrust policy pol1 match destination-address any
set security policies from-zone trust to-zone untrust policy pol1 match application any
set security policies from-zone trust to-zone untrust policy pol1 then permit tunnel ipsec-vpn
```

```
first_vpn
set security policies from-zone untrust to-zone trust policy pol1 match application any
set security policies from-zone untrust to-zone trust policy pol1 then permit tunnel ipsec-vpn
first_vpn
```

### 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 untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy pol1 match source-address any
user@host# set policy pol1 match destination-address any
user@host# set policy pol1 match application any
user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
```

2. Create the security policy to permit traffic from the untrust zone to the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy pol1 match application any
user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
```

### 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 instructions in this example to correct the configuration.

```
[edit]
user@host# show security policies
    from-zone trust to-zone untrust {
        policy pol1 {
            match {
                source-address any;
                destination-address any;
```

```
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
    from-zone untrust to-zone trust {
        policy pol1 {
            match {
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
```

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

#### Configuring NAT for the 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 security nat source rule-set ipsec from zone trust
set security nat source rule-set ipsec to zone untrust
set security nat source rule-set ipsec rule 1 match source-address 0.0.0.0/0
set security nat source rule-set ipsec rule 1 then source-nat interface
set security policies from-zone trust to-zone untrust policy allow-all match source-address any
set security policies from-zone trust to-zone untrust policy allow-all match destination-address
```

```
any
set security policies from-zone trust to-zone untrust policy allow-all match application any
set security policies from-zone trust to-zone untrust policy allow-all then permit
set security policies from-zone untrust to-zone trust policy allow-all match application any
set security policies from-zone untrust to-zone trust policy allow-all then permit
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 security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet address 12.168.99.1/24
set interfaces ge-0/0/1 unit 0 family inet address 1.1.100.23/24
set routing-options static route 0.0.0.0/0 next-hop 1.1.100.22
```

### 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 initiator providing NAT:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 12.168.99.1/24
user@host# set ge-0/0/1 unit 0 family inet address 1.1.100.23/24
```

2. 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-0/0/0.0
```

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/1.0
```

3. Configure NAT.

```
[edit security nat source rule-set ipsec]
user@host# set from zone trust
user@host# set to zone untrust
user@host# set rule 1 match source-address 0.0.0.0/0
user@host# set rule 1 then source-nat interface
```

4. Configure the default security policy.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy allow-all match source-address any
user@host# set from-zone trust to-zone untrust policy allow-all match destination-address any
user@host# set from-zone trust to-zone untrust policy allow-all match application any
user@host# set from-zone trust to-zone untrust policy allow-all then permit
user@host# set from-zone untrust to-zone trust policy allow-all match application any
user@host# set from-zone untrust to-zone trust policy allow-all then permit
```

5. Configure the routing option.

```
[edit routing-options
user@host# set static route 0.0.0.0/0 next-hop 1.1.100.22
```

### Results

From configuration mode, confirm your configuration by entering the `show security nat` 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 nat
        source {
            rule-set ipsec {
                from zone trust;
                to zone untrust;
                rule 1 {
                    match {
                        source-address 0.0.0.0/0;
                    }
```

```
                then {
                    source-nat {
                        interface;
                    }
                }
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow-all {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone untrust to-zone trust {
            policy allow-all {
                match {
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
```

```
            interfaces {
                ge-0/0/0.0;
            }
        }
        security-zone untrust {
            host-inbound-traffic {
            }
            interfaces {
                ge-0/0/1.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 12.168.99.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 1.1.100.23/24;
            }
        }
    }
}
routing-options {
    static {
        route 0.0.0.0/0 next-hop 1.1.100.22;
    }
```

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

**Configuring Interface, Routing Options, and Security Zones for the 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 interfaces ge-0/0/0 unit 0 family inet address 13.168.11.100/24
set interfaces ge-0/0/1 unit 0 family inet address 10.2.99.1/24
set routing-options static route 10.1.99.0/24 next-hop 13.168.11.1
set routing-options static route 1.1.100.0/24 next-hop 13.168.11.1
set security zones security-zone untrust interfaces ge-0/0/0.0
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-0/0/1.0
```

### 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 interfaces, static routes, security zones, and security policies:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 13.168.11.100/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.2.99.1/24
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 10.1.99.0/24 next-hop 13.168.11.1
user@host# set routing-options static route 1.1.100.0/24 next-hop 13.168.11.1
```

3. Assign an interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/0.0
```

4. Configure the trust security zone.

```
[edit]
user@host# set security zones security-zone trust host-inbound-traffic protocols all
```

5. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/1.0
```

6. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
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 instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 13.168.11.100/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
```

```
            family inet {
                address 10.2.99.1/24;
            }
        }
    }
```

```
[edit]
user@host# show routing-options
    static {
        route 10.1.99.0/24 next-hop 13.168.11.1;
        route 1.1.100.0/24 next-hop 13.168.11.1;
    }
```

```
[edit]
user@host# show security zones
    security-zone untrust {
        host-inbound-traffic {
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
```

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

### Configuring IKE for the 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 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 md5
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode aggressive
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 gate ike-policy ike_pol
set security ike gateway gate dynamic hostname chicago
set security ike gateway gate external-interface ge-0/0/0.0
```

#### 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. Define the IKE proposal authentication method.

```
[edit security ike proposal ike_prop]
user@host# set authentication-method pre-shared-key
```

2. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike_prop]
user@host# set dh-group group2
```

3. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike_prop]
user@host# set authentication-algorithm md5
```

4. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike_prop]
user@host# set encryption-algorithm 3des-cbc
```

5. Create an IKE Phase 1 policy.

```
[edit security ike]
user@host# edit policy ike_pol
```

6. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike_pol]
user@host# set mode aggressive
```

7. Specify a reference to the IKE proposal.

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

8. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike_pol]
user@host# set pre-shared-key ascii-text "$ABC123"
```

9. Create an IKE Phase 1 gateway and define its dynamic host name.

```
[edit security ike gateway gate]
user@host# set dynamic hostname chicago
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike gateway gate]
user@host# set external-interface ge-0/0/0.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gate]
user@host# set ike-policy ike_pol
```

### 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@host# show security ike
    proposal ike_prop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm 3des-cbc;
    }
    policy ike_pol {
        mode aggressive;
        proposals ike_prop;
        pre-shared-key ascii-text "$ABC123";
    }
    gateway gate {
        ike-policy ike_pol;
        dynamic hostname chicago;
        external-interface ge-0/0/0.0;
    }
```

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

### Configuring IPsec for the 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 security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group1
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn first_vpn ike gateway gate
set security ipsec vpn first_vpn ike ipsec-policy ipsec_pol
```

## 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# edit security ipsec proposal ipsec_prop
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security security ipsec proposal ipsec_prop]
user@host# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set authentication-algorithm hmac-md5-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set encryption-algorithm 3des-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# edit policy ipsec_pol
```

6. Set IPsec Phase 2 to use perfect forward secrecy (PFS) group1.

```
[edit security ipsec policy ipsec_pol]
user@host# set perfect-forward-secrecy keys group1
```

7. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec_pol]
user@host# set proposals ipsec_prop
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn first_vpn ike gateway gate
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn first_vpn ike ipsec-policy ipsec_pol
```

### 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@host# show security ipsec
    proposal ipsec_prop {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm 3des-cbc;
    }
    policy ipsec_pol {
        perfect-forward-secrecy {
            keys group1;
        }
        proposals ipsec_prop;
    }
    vpn first_vpn {
        ike {
            gateway gate;
            ipsec-policy ipsec_pol;
        }
    }
```

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

#### Configuring Security Policies for the 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 security policies from-zone trust to-zone untrust policy pol1 match source-address any
set security policies from-zone trust to-zone untrust policy pol1 match destination-address any
set security policies from-zone trust to-zone untrust policy pol1 match application any
set security policies from-zone trust to-zone untrust policy pol1 then permit tunnel ipsec-vpn
first_vpn
```

```
set security policies from-zone untrust to-zone trust policy pol1 match application any
set security policies from-zone untrust to-zone trust policy pol1 then permit tunnel ipsec-vpn
first_vpn
```

### 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 untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy pol1 match source-address any
user@host# set policy pol1 match destination-address any
user@host# set policy pol1 match application any
user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
```

2. Create the security policy to permit traffic from the untrust zone to the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy pol1 match application any
user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
```

### 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 instructions in this example to correct the configuration.

```
[edit]
user@host# show security policies
    from-zone trust to-zone untrust {
        policy pol1 {
            match {
                source-address any;
                destination-address any;
                application any;
```

```
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
    from-zone untrust to-zone trust {
        policy pol1 {
            match {
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
```

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

#### Configuring NAT for the 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 security nat source rule-set ipsec from zone trust
set security nat source rule-set ipsec to zone untrust
set security nat source rule-set ipsec rule 1 match source-address 0.0.0.0/0
set security nat source rule-set ipsec rule 1 then source-nat interface
set security policies from-zone trust to-zone untrust policy allow-all match source-address any
set security policies from-zone trust to-zone untrust policy allow-all match destination-address
any
```

```
set security policies from-zone trust to-zone untrust policy allow-all match application any
set security policies from-zone trust to-zone untrust policy allow-all then permit
set security policies from-zone untrust to-zone trust policy allow-all match application any
set security policies from-zone untrust to-zone trust policy allow-all then permit
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 security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet address 13.168.11.1/24
set interfaces ge-0/0/1 unit 0 family inet address 1.1.100.22/24
set routing-options static route 0.0.0.0/0 next-hop 1.1.100.23
```

### 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 responder providing NAT:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 13.168.11.1/24
user@host# set ge-0/0/1 unit 0 family inet address 1.1.100.22/24
```

2. 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-0/0/0.0
```

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/1.0
```

3. Configure NAT.

```
[edit security nat source rule-set ipsec]
user@host# set from zone trust
```

```
user@host# set to zone untrust
user@host# set rule 1 match source-address 0.0.0.0/0
user@host# set rule 1 then source-nat interface
```

4. Configure the default security policy.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy allow-all match source-address any
user@host# set from-zone trust to-zone untrust policy allow-all match destination-address any
user@host# set from-zone trust to-zone untrust policy allow-all match application any
user@host# set from-zone trust to-zone untrust policy allow-all then permit
user@host# set from-zone untrust to-zone trust policy allow-all match application any
user@host# set from-zone untrust to-zone trust policy allow-all then permit
```

5. Configure the routing option.

```
[edit routing-options
user@host# set static route 0.0.0.0/0 next-hop 1.1.100.23
```

### Results

From configuration mode, confirm your configuration by entering the `show security nat` 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 nat
    nat {
        source {
            rule-set ipsec {
                from zone trust;
                to zone untrust;
                rule 1 {
                    match {
                        source-address 0.0.0.0/0;
                    }
                    then {
                        source-nat {
                            interface;
```

```
                    }
                }
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow-all {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone untrust to-zone trust {
            policy allow-all {
                match {
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                ge-0/0/0.0;
            }
```

```
        }
        security-zone untrust {
            host-inbound-traffic {
            }
            interfaces {
                ge-0/0/1.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 13.168.11.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 1.1.100.22/24;
            }
        }
    }
}
routing-options {
    static {
        route 0.0.0.0/0 next-hop 1.1.100.23;
    }
```

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

## Verification

**IN THIS SECTION**


- Verifying the IKE Phase 1 Status for the Initiator | **681**
- Verifying IPsec Security Associations for the Initiator | **683**
- Verifying the IKE Phase 1 Status for the Responder | **686**

- Verifying IPsec Security Associations for the Responder | 688

To confirm that the configuration is working properly, perform these tasks:

### Verifying the IKE Phase 1 Status for the Initiator

#### Purpose

Verify the IKE Phase 1 status.

#### Action

Before starting the verification process, you must send traffic from a host in the 10.1.99.0 network to a host in the 10.2.99.0 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 operation from 10.1.99.2 to 10.2.99.2.

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   State  Initiator cookie  Responder cookie  Mode           Remote Address
5649304 UP     c3193077d38e426f  011f0ef28d928f4c  Aggressive     13.168.11.
```

```
user@host> show security ike security-associations index 5649304 detail
IKE peer 13.168.11.100, Index 5649304, Gateway Name: gate
  Role: Initiator, State: UP
  Initiator cookie: c3193077d38e426f, Responder cookie: 011f0ef28d928f4c
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 12.168.99.100:4500, Remote: 13.168.11.100:4500
  Lifetime: Expires in 26359 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: 13.168.11.100
```

```
AAA assigned IP: 0.0.0.0
Algorithms:
 Authentication        : hmac-md5-96
 Encryption            : 3des-cbc
 Pseudo random function: hmac-md5
 Diffie-Hellman group  : DH-group-2
Traffic statistics:
 Input  bytes  :                 1140
 Output bytes  :                 1203
 Input  packets:                    6
 Output packets:                    6
 Input  fragmentated packets:       0
 Output fragmentated packets:       0
IPSec security associations: 2 created, 3 deleted
Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Initiator, Message ID: 0
  Local: 12.168.99.100:4500, Remote: 13.168.11.100:4500
  Local identity: chicago
  Remote identity: 13.168.11.100
  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.
- Role initiator state
  - Up—The Phase 1 SA has been established.
  - Down—There was a problem establishing the Phase 1 SA.
  - Both peers in the IPsec SA pair are using port 4500, which indicates that NAT-T is implemented. (NAT-T uses port 4500 or another random high-numbered port.)

    - Peer IKE ID—Verify the remote (responder) ID is correct. In this example, the hostname is sunnyvale.
    - 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 for the Initiator**

### 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@host> show security ipsec security-associations

Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <2      ESP:3des/md5    aff3ac30 1103/ unlim  -   root 4500  13.168.11.100
  >2      ESP:3des/md5    40539d12 1103/ unlim  -   root 4500  13.168.11.100
```

```
user@host> show security ipsec security-associations detail

ID: 2 Virtual-system: root, VPN Name: first_vpn
  Local Gateway: 12.168.99.100, Remote Gateway: 13.168.11.100
  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                              , Policy-name: pol1
  Port: 4500, Nego#: 7, Fail#: 0, Def-Del#: 0 Flag: 0x600829
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Wed Apr 08 2020 19:13:53: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Wed Apr 08 2020 19:13:09: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020 19:13:09: User cleared IPSec SA from CLI (1 times)
    Wed Apr 08 2020 19:13:09: IKE SA negotiation successfully completed (5 times)
    Wed Apr 08 2020 19:12:18: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020 19:12:18: User cleared IPSec SA from CLI (1 times)
    Wed Apr 08 2020 19:12:12: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020 19:12:12: User cleared IPSec SA from CLI (1 times)
    Wed Apr 08 2020 19:06:52: Peer's IKE-ID validation failed during negotiation (2 times)
    Wed Apr 08 2020
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (2 times)
    Wed Apr 08 2020 19:05:26: Peer's IKE-ID validation failed during negotiation (1 times)
    Wed Apr 08 2020
```

```
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (1 times)
    Wed Apr 08 2020 19:04:26: Peer's IKE-ID validation failed during negotiation (1 times)
    Wed Apr 08 2020
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (1 times)
    Wed Apr 08 2020 19:03:26: Peer's IKE-ID validation failed during negotiation (1 times)
  Direction: inbound, SPI: aff3ac30, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1093 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 453 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-md5-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 40539d12, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1093 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 453 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-md5-96, 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 a NAT address of 13.168.11.100.
- Both peers in the IPsec SA pair are using port 4500, which indicates that NAT-T is implemented. (NAT-T uses port 4500 or another random high-numbered port.).
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3390/ unlimited value indicates that the Phase 2 lifetime expires in 3390 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 for the Responder

#### Purpose

Verify the IKE Phase 1 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@host> show security ike security-associations

Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
2914355 UP     c3193077d38e426f  011f0ef28d928f4c  Aggressive     1.1.100.23
```

```
user@host> show security ike security-associations index 2914355 detail

  IKE peer 1.1.100.23, Index 2914355, Gateway Name: gate
  Role: Responder, State: UP
  Initiator cookie: c3193077d38e426f, Responder cookie: 011f0ef28d928f4c
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 13.168.11.100:4500, Remote: 1.1.100.23:23434
  Lifetime: Expires in 26137 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: chicago
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-md5-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-md5
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                1203
   Output bytes  :                1140
   Input  packets:                   6
   Output packets:                   6
```

```
 Input  fragmentated packets:       0
 Output fragmentated packets:       0
IPSec security associations: 2 created, 0 deleted
Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Responder, Message ID: 0
  Local: 13.168.11.100:4500, Remote: 1.1.100.23:23434
  Local identity: 13.168.11.100
  Remote identity: chicago
  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.
- 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 local ID for the peer is correct. In this example, the hostname is chicago.
  - 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 for the Responder**

### 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@host> show security ipsec security-associations

Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108878 ESP:3des/md5  40539d12 939/ unlim   -   root 23434 1.1.100.23
  >67108878 ESP:3des/md5  aff3ac30 939/ unlim   -   root 23434 1.1.100.23
```

```
user@host> show security ipsec security-associations detail

  ID: 67108878 Virtual-system: root, VPN Name: first_vpn
  Local Gateway: 13.168.11.100, Remote Gateway: 1.1.100.23
  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                              , Policy-name: pol1
Port: 23434, Nego#: 8, Fail#: 0, Def-Del#: 0 Flag: 0x608829
Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
Tunnel events:
  Wed Apr 08 2020 19:14:22: IPSec SA negotiation successfully completed (1 times)
  Wed Apr 08 2020 19:14:15: User cleared IPSec SA from CLI (1 times)
  Wed Apr 08 2020 19:13:39: IPSec SA negotiation successfully completed (3 times)
  Wed Apr 08 2020 19:13:39: IKE SA negotiation successfully completed (4 times)
  Wed Apr 08 2020
  : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
  Wed Apr 08 2020 19:10:39: IPSec SA negotiation successfully completed (1 times)
  Wed Apr 08 2020 19:10:20: User cleared IPSec SA from CLI (1 times)
  Wed Apr 08 2020 19:10:08: IPSec SA negotiation successfully completed (1 times)
  Wed Apr 08 2020
  : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
Direction: inbound, SPI: 40539d12, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 930 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 335 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-md5-96, Encryption: 3des-cbc
  Anti-replay service: counter-based enabled, Replay window size: 64
Direction: outbound, SPI: aff3ac30, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 930 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 335 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-md5-96, 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 a NAT address of 1.1.100.23.
- Both peers in the IPsec SA pair are using port 4500, which indicates that NAT-T is implemented. (NAT-T uses port 4500 or another random high-numbered port.)

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3571/ unlim value indicates that the Phase 2 lifetime expires in 3571 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.

**SEE ALSO**

IPsec Overview | 20

Understanding Policy-Based IPsec VPNs | 279

## Example: Configuring NAT-T with Dynamic Endpoint VPN

**IN THIS SECTION**

- Requirements | 690
- Overview | 691
- Configuration | 693
- Verification | 709

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:

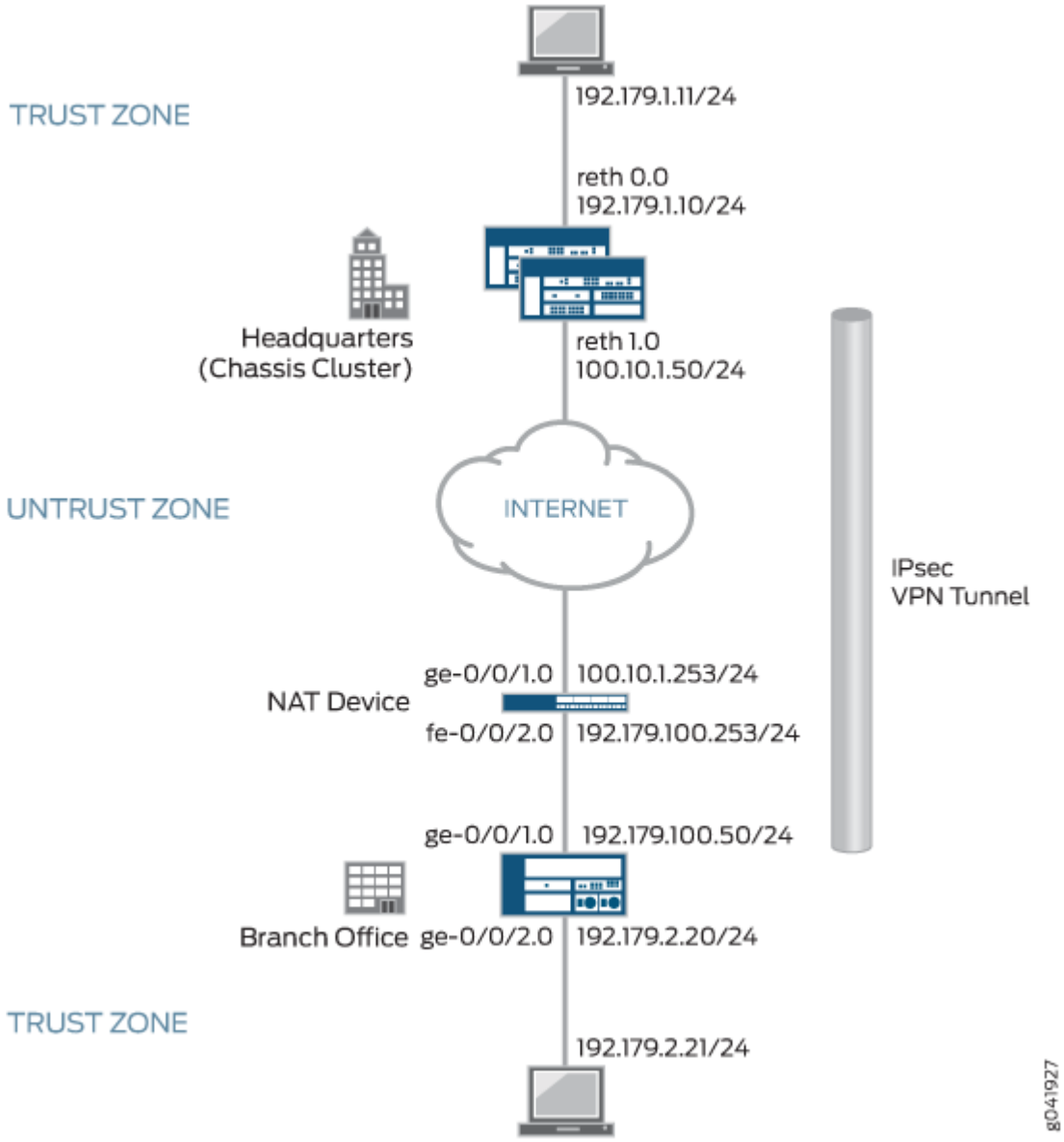
- Two SRX Series Firewalls configured in a chassis cluster
- One SRX Series Firewall providing NAT
- One SRX Series Firewall providing branch office network access
- Junos OS Release 12.1X46-D10 or later for IKEv2 NAT-T support

## 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 48 on page 692 shows an example of a topology with NAT-Traversal (NAT-T) and dynamic endpoint VPN.

**Figure 48: 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*.

Starting with Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, the default value for the `nat-keepalive` option configured at the `[edit security ike gateway gateway-name]` hierarchy level has been changed from 5 seconds to 20 seconds.

In SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, IKE negotiations involving NAT traversal do not work if the IKE peer is behind a NAT device that will change the source IP address of the IKE packets during the negotiation. For example, if the NAT device is configured with DIP, it changes the source IP because the IKE protocol switches the UDP port from 500 to 4500. (Platform support depends on the Junos OS release in your installation.)

## Configuration

**IN THIS SECTION**


- Configuring the Branch Office Device (IKEv2 Initiator) | **693**
- Configuring the NAT Device | **699**
- Configuring the Headquarters Device (IKEv2 Responder) | **702**


### 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 | 709
- Verifying IPsec Security Associations for the Responder | 711


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 | 20

*Security Policies Overview*

**Release History Table**

| Release | Description |
|---|---|
| 12.1X46-D10 | Starting with Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, the default value for the `nat-keepalive` option configured at the `[edit security ike gateway gateway-name]` hierarchy level has been changed from 5 seconds to 20 seconds. |

## RELATED DOCUMENTATION

Traffic Selectors in Route-Based VPNs | 536

10
CHAPTER

# Group VPN

Group VPNv1 | 715

Group VPNv2 | 769

Group VPNv2 Server Clusters | 830

# Group VPNv1

IN THIS SECTION


- Group VPNv1 Overview | 715
- Group VPNv1 Configuration Overview | 725
- Understanding IKE Phase 1 Configuration for Group VPNv1 | 726
- Understanding IPsec SA Configuration for Group VPNv1 | 727
- Understanding Dynamic Policies for Group VPNv1 | 727
- Understanding Antireplay for Group VPNv1 | 728
- Example: Configuring Group VPNv1 Server and Members | 729
- Example: Configuring Group VPNv1 Server-Member Communication for Unicast Rekey Messages | 750
- Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages | 752
- Example: Configuring Group VPNv1 with Server-Member Colocation | 756


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.

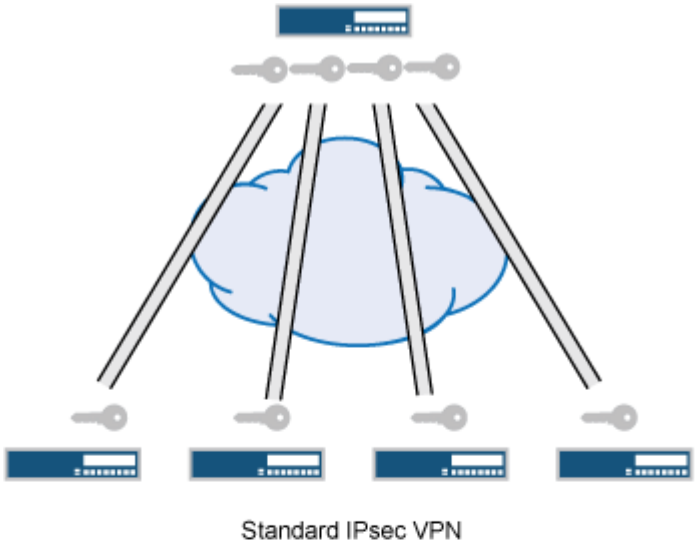
## Group VPNv1 Overview

IN THIS SECTION


- Understanding the GDOI Protocol for Group VPNv1 | 718
- Understanding Group VPNv1 Limitations | 718
- Understanding Group VPNv1 Servers and Members | 720
- Understanding Group VPNv1 Server-Member Communication | 720
- Understanding Group VPNv1 Group Key Operations | 721
- Understanding Group VPNv1 Heartbeat Messages | 724
- Understanding Group VPNv1 Server-Member Colocation Mode | 724

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 49 on page 717).

**Figure 49: Standard IPsec VPN and Group VPNv1**




Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices. 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.

Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members can interoperate with Group VPNv2 servers.

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. Group VPNv2 is supported on vSRX Virtual Firewall instances starting with Junos OS Release 15.1X49-D30, SRX Series Firewalls starting with Junos OS Release 15.1X49-D40, and MX Series devices starting with Junos OS Release 15.1r2.

## 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

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. 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

Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices. 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 | 20

Understanding IKE and IPsec Packet Processing | 181

Group VPNv1 Configuration Overview | 725

## 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 777.
2. Phase 2 IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv1" on page 727.
3. VPN group. See "Group VPNv1 Configuration Overview" on page 725.

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 726.
2. Phase 2 IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv1" on page 727.
3. Scope policy that determines which group policies are installed on the member. See "Understanding Dynamic Policies for Group VPNv1" on page 727.

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 727.
- Server-member communication—Optional configuration that allows the server to send rekey messages to members. See "Group VPNv1 Overview" on page 715.
- Antireplay—Optional configuration that detects packet interception and replay. See "Understanding Antireplay for Group VPNv1" on page 728.

## 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 on SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, and SRX650 devices. 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 | 20

Understanding IKE and IPsec Packet Processing | 181

## Example: Configuring Group VPNv1 Server and Members

**IN THIS SECTION**

- Requirements | 729
- Overview | 729
- Configuration | 730
- Verification | 747

This example shows how to configure group VPNv1 to extend IPsec architecture to support SAs that are shared by a group of security devices. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

### Requirements

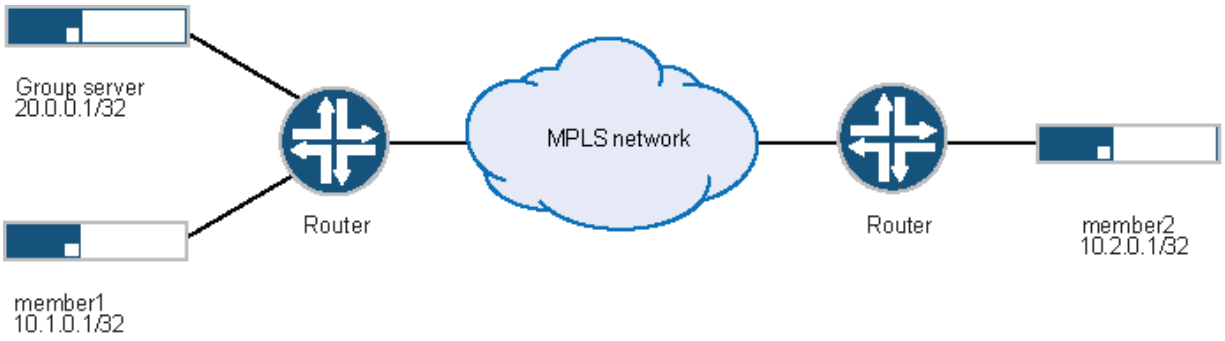
Before you begin:

- 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 50 on page 730, 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 50: 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 | **731**
- Configuring Member1 | **735**
- Configuring Member2 | **740**

### 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 | **748**
- Verifying Dynamic Policies for Member2 | **749**


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 | **751**
- Overview | **751**
- Configuration | **751**
- Verification | **752**

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. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

## 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 | 751

### 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 | 181

## Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages

IN THIS SECTION


- Requirements | 753
- Overview | 753
- Configuration | 753
- Verification | 755

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. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

## 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 729 or "Example: Configuring Group VPNv1 with Server-Member Colocation" on page 756.
- 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 | 754

**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 | **756**


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

Example: Configuring a Group IKE ID for Multiple Users

Understanding IKE and IPsec Packet Processing | 181

## Example: Configuring Group VPNv1 with Server-Member Colocation

**IN THIS SECTION**


- Requirements | 756
- Overview | 757
- Configuration | 757
- Verification | 767


This example shows how to configure a device for colocation mode, which allows server and member functions to coexist on the same physical device. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

### Requirements

Before you begin:

- Configure the Juniper Networks security devices for network communication.

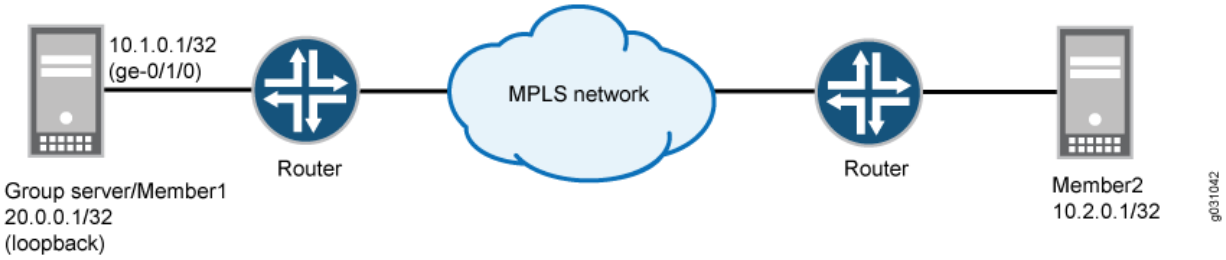
- Configure network interfaces on server and member devices. See Interfaces User Guide for Security Devices.

## Overview

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 51 on page 757, 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 51: 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 729.

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 | 758

**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 | **767**
- Verifying Group VPN Server Security Associations for IKE | **768**
- Verifying Group VPN Server Security Associations for IPsec | **768**
- Verifying Group VPN Member Security Associations for IKE | **768**
- Verifying Group VPN Member Security Associations for IPsec | **768**


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.

**Release History Table**

| 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. |

RELATED DOCUMENTATION

Monitoring VPN Traffic

# Group VPNv2

**IN THIS SECTION**


- Group VPNv2 Overview | **770**
- Group VPNv2 Configuration Overview | **775**
- Understanding IKE Phase 1 Configuration for Group VPNv2 | **777**
- Understanding IPsec SA Configuration for Group VPNv2 | **777**
- Understanding Group VPNv2 Traffic Steering | **778**
- Understanding the Group VPNv2 Recovery Probe Process | **780**
- Understanding Group VPNv2 Antireplay | **781**
- Example: Configuring a Group VPNv2 Server and Members | **781**
- Example: Configuring Group VPNv2 Server-Member Communication for Unicast Rekey Messages | **828**

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.

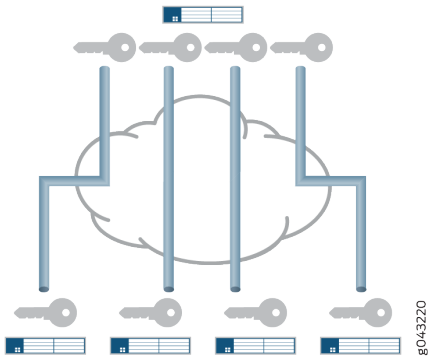
## Group VPNv2 Overview

IN THIS SECTION


- Understanding the GDOI Protocol for Group VPNv2 | **771**
- Understanding Group VPNv2 Servers and Members | **772**
- Understanding Group VPNv2 Limitations | **773**
- Understanding Group VPNv2 Server-Member Communication | **773**
- Understanding Group VPNv2 Key Operations | **774**


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 52 on page 770).

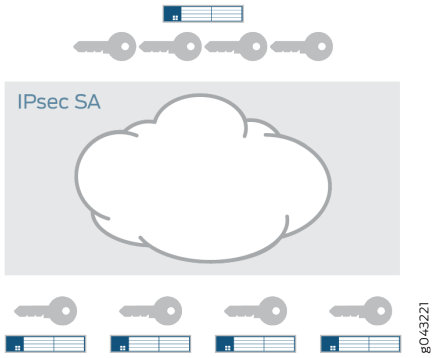
**Figure 52: Point-to-Point SAs**




Group VPNv2 extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 53 on page 771). With Group VPNv2, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header. Group VPNv2 is supported on SRX300,

SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

**Figure 53: 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. Group VPNv2 is supported on vSRX Virtual Firewall instances and all SRX Series Firewalls except for SRX5400, SRX5600, and SRX5800 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

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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)*.

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. The following are not supported in this release for 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

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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 830.

On the group server(s), configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 777.
2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 777.
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 775 and "Understanding Group VPNv2 Traffic Steering" on page 778.

On the group member, configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 777.
2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 777.
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 778.
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 778.
- 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 830.

- 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 830.
- Antireplay—Optional configuration that detects packet interception and replay. See "Understanding Group VPNv2 Antireplay" on page 781.

## 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. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

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

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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 | 830

## Understanding Group VPNv2 Traffic Steering

**IN THIS SECTION**


- Group Policies Configured on Group Servers | 778
- IPsec Policies Configured on Group Members | 779
- Fail-Close | 779
- Exclude and Fail-Open Rules | 779
- Priorities of IPsec Policies and Rules | 780


Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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 | 838

## Understanding the Group VPNv2 Recovery Probe Process

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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

Group VPNv2 antireplay is supported on vSRX Virtual Firewall instances and all SRX Series Firewalls except for SRX5400, SRX5600, and SRX5800 devices. 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 | 728

## Example: Configuring a Group VPNv2 Server and Members

**IN THIS SECTION**


- Requirements | 782
- Overview | 782
- Configuration | 783
- Verification | 819


This example shows how to configure a Group VPNv2 server to provide group controller/key server (GCKS) support to Group VPNv2 group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

## 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 | **783**

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.

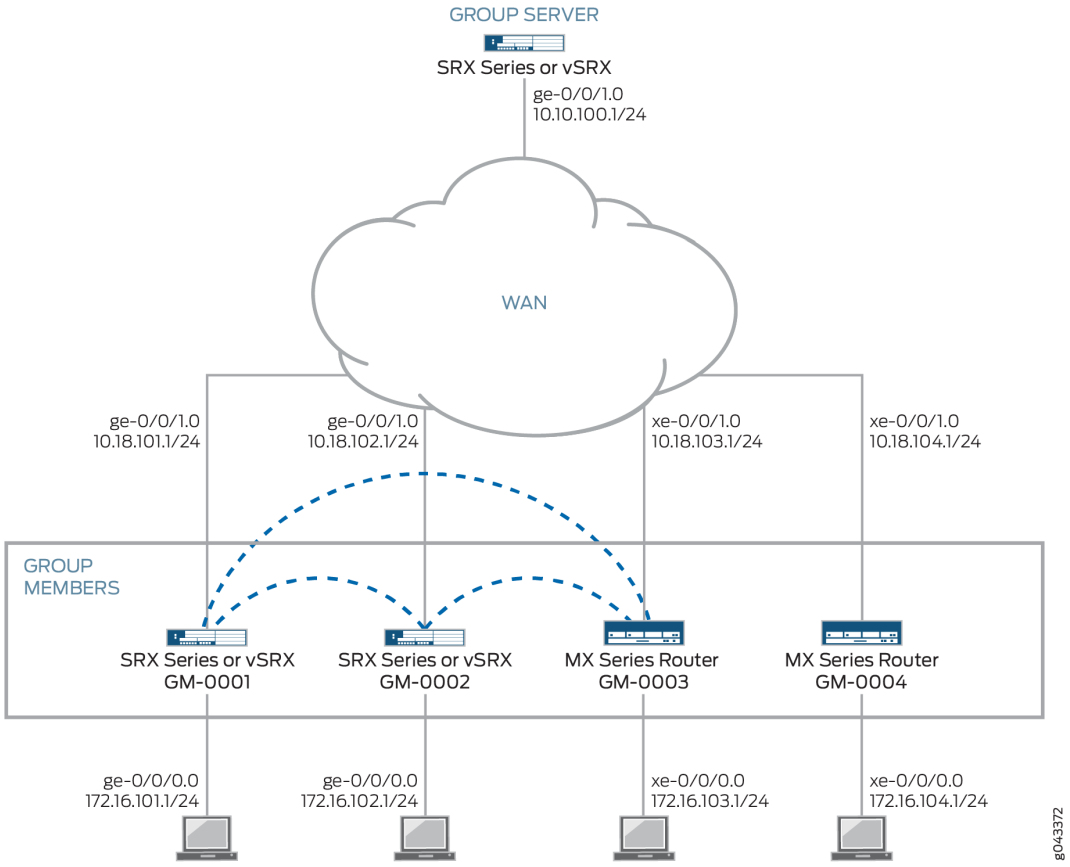
The same group identifier must be configured on both the group server and the group members. In this example, the group name is GROUP_ID-0001 and the group identifier is 1. The group policy configured on the server specifies that the SA and key are applied to traffic between subnetworks in the 172.16.0.0/12 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 54 on page 783 shows the Juniper Networks devices to be configured for this example.

**Figure 54: Group VPNv2 Server with SRX Series Firewall or vSRX Virtual Firewall and MX Series Members**




## Configuration

**IN THIS SECTION**


- Configuring the Group Server | **784**
- Configuring Group Member GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **791**
- Configuring Group Member GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **799**
- Configuring Group Member GM-0003 (MX Series Device) | **806**

- Configuring Group Member GM-0004 (MX Series Device) | **812**


### 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 | **819**
- Verifying That Group Keys Are Distributed | **820**
- Verifying Group VPN SAs on the Group Server | **821**
- Verifying Group VPN SAs on Group Members | **822**
- Verifying IPsec SAs on the Group Server | **823**
- Verifying IPsec SAs on the Group Members | **824**
- Verifying Group Policies (SRX Series Firewall or vSRX Virtual Firewall Group Members Only) | **827**


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 | 828
- Overview | 828
- Configuration | 829
- Verification | 829


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. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

## 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 | 829

### 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 | 729

Example: Configuring Group VPNv1 with Server-Member Colocation | 756

RELATED DOCUMENTATION

Monitoring VPN Traffic

VPN Session Affinity | 1342

# Group VPNv2 Server Clusters

IN THIS SECTION

- Understanding Group VPNv2 Server Clusters | 830
- Understanding Group VPNv2 Server Cluster Limitations | 835
- Understanding Group VPNv2 Server Cluster Messages | 836
- Understanding Configuration Changes with Group VPNv2 Server Clusters | 838
- Migrating a Standalone Group VPNv2 Server to a Group VPNv2 Server Cluster | 843
- Example: Configuring a Group VPNv2 Server Cluster and Members | 844

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 | 831

Group Member Registration with Server Clusters | 833
Dead Peer Detection | 834
Load Balancing | 834

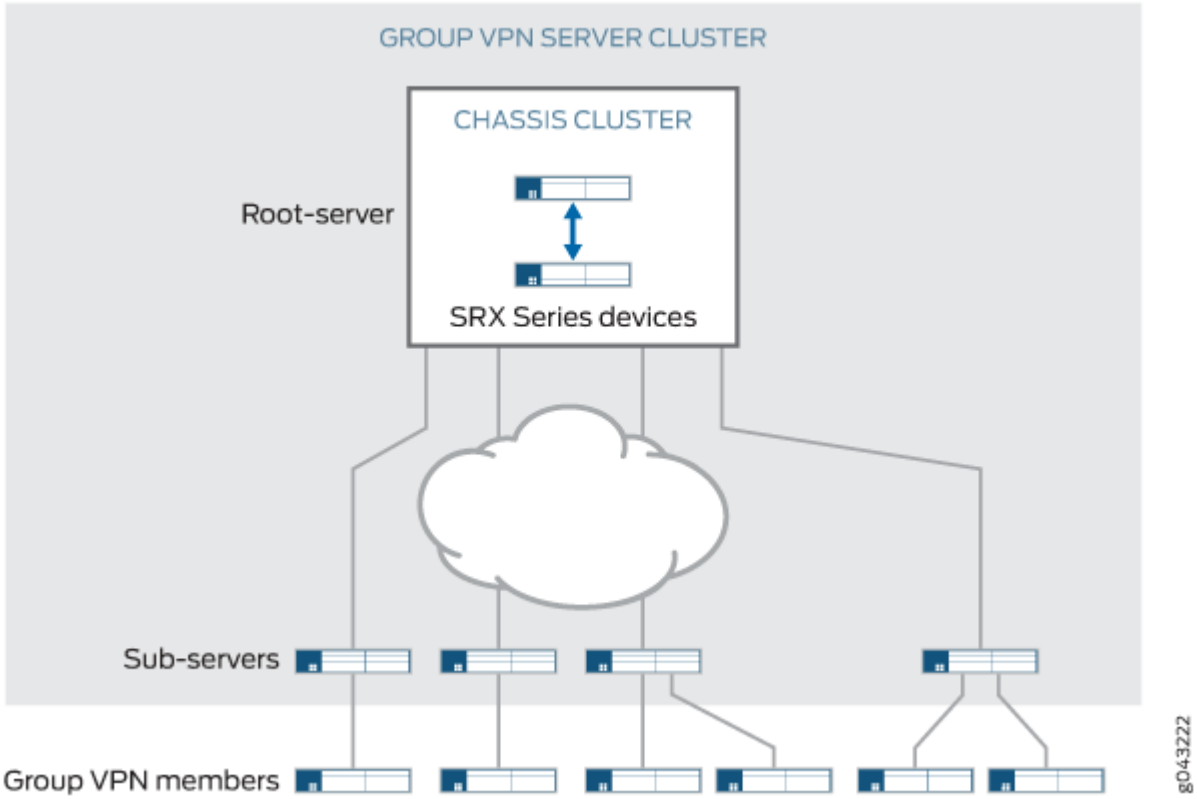
In the Group Domain of Interpretation (GDOI) protocol, the group controller/key server (GCKS) manages Group VPN security associations (SAs), and generates encryption keys and distributes them to group members. Group members encrypt traffic based on the group SAs and keys provided by the GCKS. If the 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.

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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 55 on page 832.

**Figure 55: 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 834.

### 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<br>Sub-server 2<br>Sub-server 3<br>Sub-server 4 | Sub-server 4<br>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 834 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 | 770

## Understanding Group VPNv2 Server Cluster Limitations

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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 | **836**
- Cluster-Init Exchanges | **837**
- Cluster-Update Messages | **838**

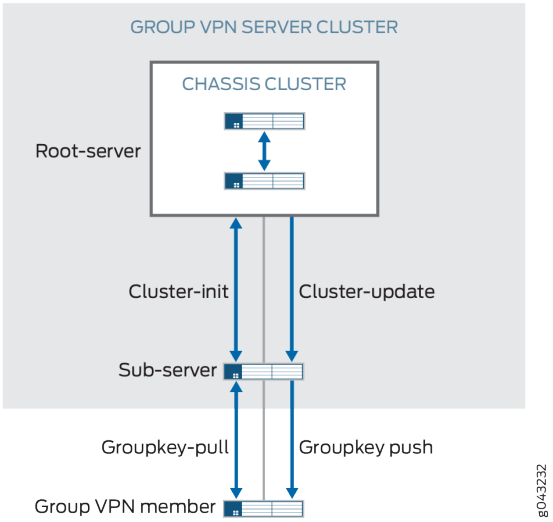

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. All messages between servers in a Group VPNv2 server cluster are encrypted and authenticated by an IKE security association (SA). Each sub-server initiates 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 56 on page 837 shows the basic messages exchanged between the Group VPNv2 server cluster and Group VPNv2 members.

**Figure 56: 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 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. 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 81 on page 839 describes the effects of various configuration changes on Group VPNv2 servers.

**Table 81: 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. | | |

**Table 81: 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 anti-replay time window | New value takes effect after the TEK rekey. | | |
| 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. |

**Table 81: 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 |
| Change | New value takes effect after TEK rekey.<br><br>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. |
| Delete | Delete TEK immediately. Send delete to delete this TEK SA. | Send delete to sub-servers. Delete TEK immediately. | Delete TEK immediately. |

Table 82 on page 841 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 82: 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: | | |

**Table 82: Effects of Group VPNv2 Server Cluster Configuration Changes *(Continued)***

| Server Cluster Configuration Change | Group VPNv2 Server Cluster | |
|---|---|---|
| | Root-server | Sub-server |
| 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<br><br>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. |
| 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

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances. 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 | 770

# Example: Configuring a Group VPNv2 Server Cluster and Members

IN THIS SECTION


- Requirements | 844
- Overview | 845
- Configuration | 847
- Verification | 916


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. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

## 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 57 on page 847. 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 | **846**

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.

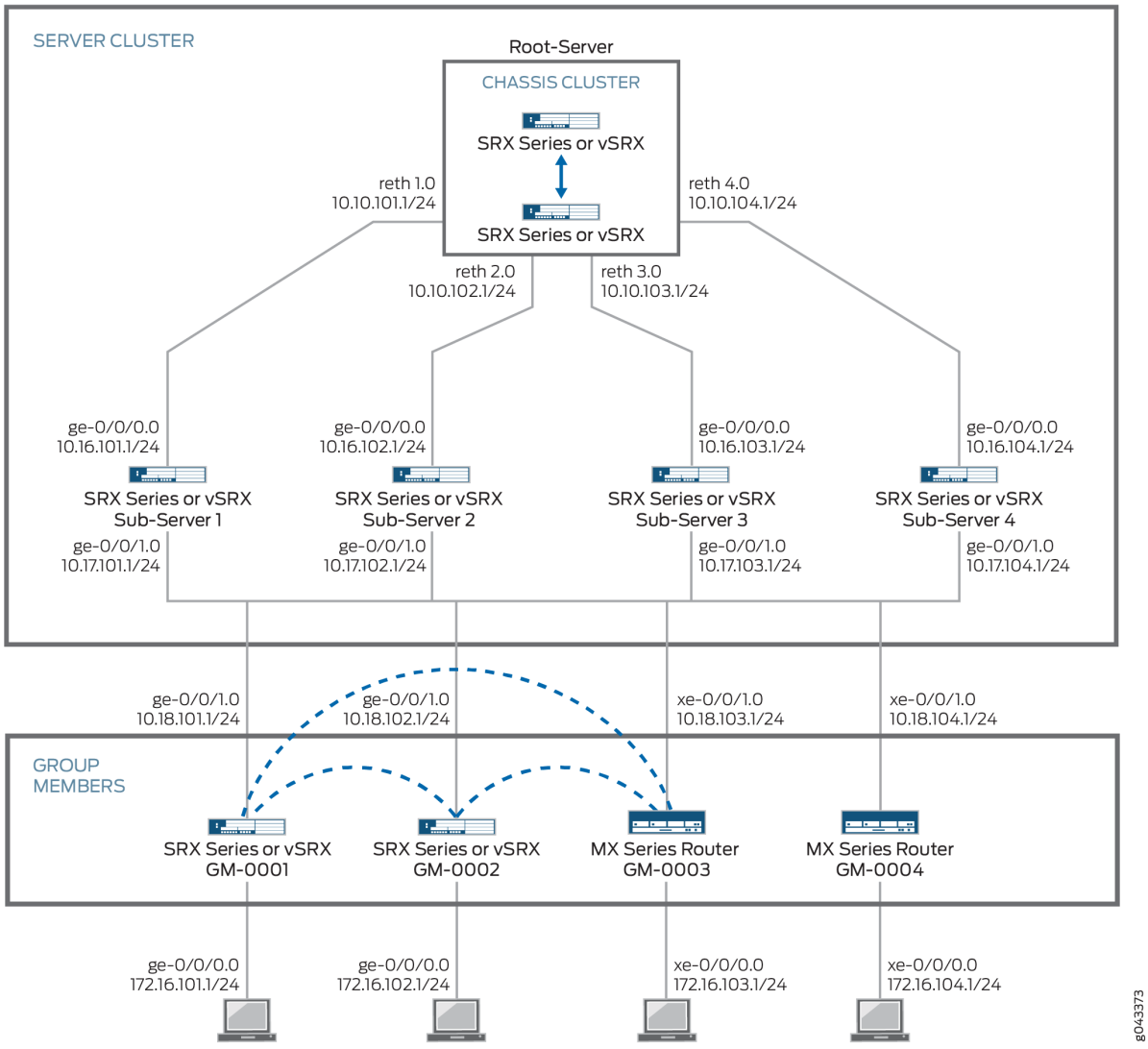
On SRX Series Firewalls or vSRX Virtual Firewall group members, an IPsec policy is configured for the group with the LAN zone as the from-zone (incoming traffic) and the WAN zone as the to-zone (outgoing traffic). A security policy is also needed to allow traffic between the LAN and WAN zones.

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 57 on page 847 shows the Juniper Networks devices to be configured for this example.

**Figure 57: Group VPNv2 Server Cluster with SRX Series or vSRX Virtual Firewall and MX Series Members**




## Configuration

**IN THIS SECTION**


- Configuring the Root-Server | **848**
- Configuring Sub-Server 1 | **857**
- Configuring Sub-Server 2 | **865**

- Configuring Sub-Server 3 | **873**
- Configuring Sub-Server 4 | **882**
- Configuring GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **890**
- Configuring GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance) | **897**
- Configuring GM-0003 (MX Series Device) | **904**
- Configuring GM-0004 (MX Series Device) | **910**


### 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 | **916**
- Verifying That SAs Are Distributed to Members | **919**
- Verifying IKE SAs on the Servers | **922**
- Verifying IPsec SAs on the Servers and Group Members | **925**
- Verifying IPsec Policies on Group Members | **928**


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 | 775

Configuring Group VPNs in Group VPNv2 on Routing Device

### RELATED DOCUMENTATION

Group VPNv1 | 715

11

CHAPTER

# ADVPN

Auto Discovery VPNs | 931

# Auto Discovery VPNs

IN THIS SECTION


- Understanding Auto Discovery VPN | **931**
- Understanding Traffic Routing with Shortcut Tunnels | **937**
- Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels | **940**
- Example: Configuring ADVPN with OSPFv3 for IPv6 Traffic | **993**
- Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established | **1029**


Auto Discovery VPN (ADVPN) dynamically establishes VPN tunnels between spokes to avoid routing traffic through the hub.

## Understanding Auto Discovery VPN

IN THIS SECTION


- ADVPN Protocol | **932**
- Establishing a Shortcut | **932**
- Shortcut Initiator and Responder Roles | **933**
- Shortcut Attributes | **934**
- Shortcut Termination | **935**
- ADVPN Configuration Limitations | **936**


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.

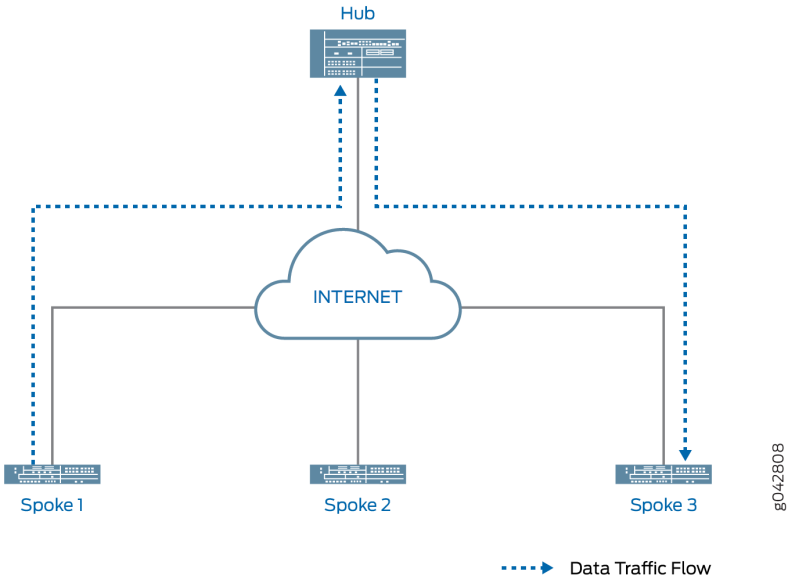
## ADVPN Protocol

ADVPN uses an extension of IKEv2 protocol to exchange messages between two peers, which 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 58 on page 932 shows traffic from Spoke 1 to Spoke 3 passing through the Hub.

**Figure 58: Spoke-to-Spoke Traffic Passing Through Hub**



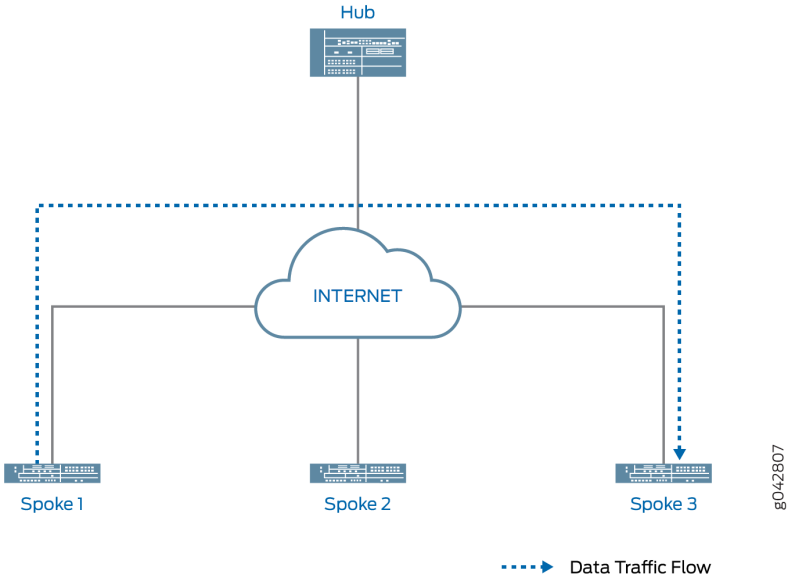

When ADVPN is configured on the devices, ADVPN shortcut capability information is exchanged between the hub and the spokes. As long as Spokes 1 and 3 have previously advertised ADVPN shortcut partner capability to the Hub, the Hub can suggest that Spokes 1 and 3 establish a shortcut between each other.

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 59 on page 933 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 59: 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, then 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 83 on page 934 shows the parameters of the shortcut.

**Table 83: 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 |

**Table 83: Shortcut Parameters *(Continued)***

| Attributes | Received/Inherited From |
|---|---|
| Install interval | Configuration |
| Local address | Configuration |
| 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 is terminated if traffic falls below 5 packets per second for 300 seconds; the idle time and idle threshold values are configurable for partners. The shortcut can be manually deleted 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.

## 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.
- As mentioned previously, you cannot create a shortcut between partners that are both behind NAT devices. The suggester can initiate a shortcut exchange if only one of the partners is behind a NAT device or if no partners are behind NAT devices.
- Multicast traffic is not supported.

  1. Starting in Junos OS Release 19.2R1, on SRX300, SRX320, SRX340, SRX345, SRX550, SRX1500, vSRX Virtual Firewall 2.0 (with 2 vCPUs), and vSRX Virtual Firewall 3.0 (with 2 vCPUs) Series devices, Protocol Independent Multicast (PIM) using point-to-multipoint (P2MP) mode supports Auto Discovery VPN in which a new `p2mp` interface type is introduced for PIM. The `p2mp` interface tracks all PIM joins per neighbor to ensure multicast forwarding or replication only happens to those neighbors that are in joined state.

- Starting with Junos OS Release 18.1R1, ADVPN supports IPv6 with your firewall running IPsec VPN service using kmd process.
- 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.

The following configurations are not supported with ADVPN with both kmd and iked processes:

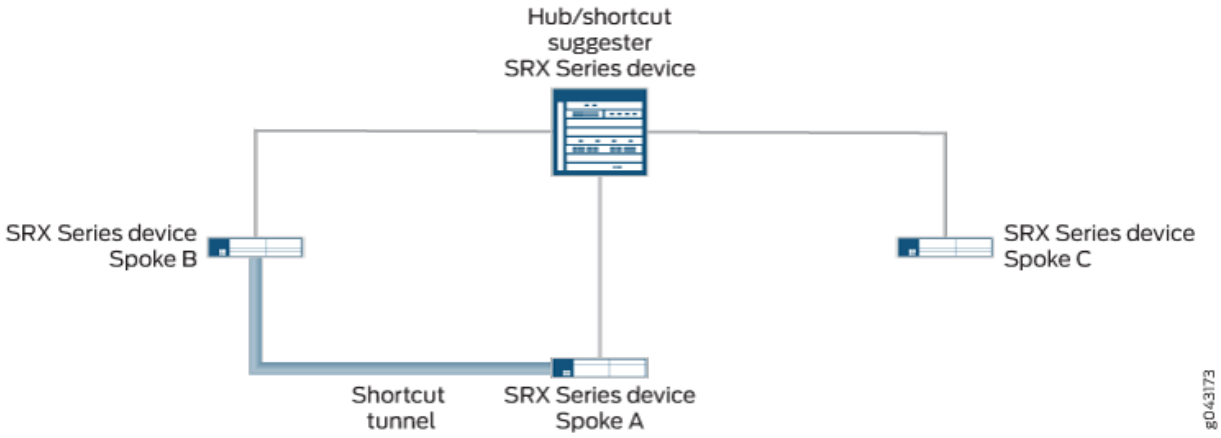
- IKEv1
- Policy-based VPN
- IKEv2 configuration payload
- Traffic selectors
- Point-to-point secure tunnel interfaces
- Seeded preshared key
- Shared preshared key—No support with kmd process
- IPv6—No support with iked process

## 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.

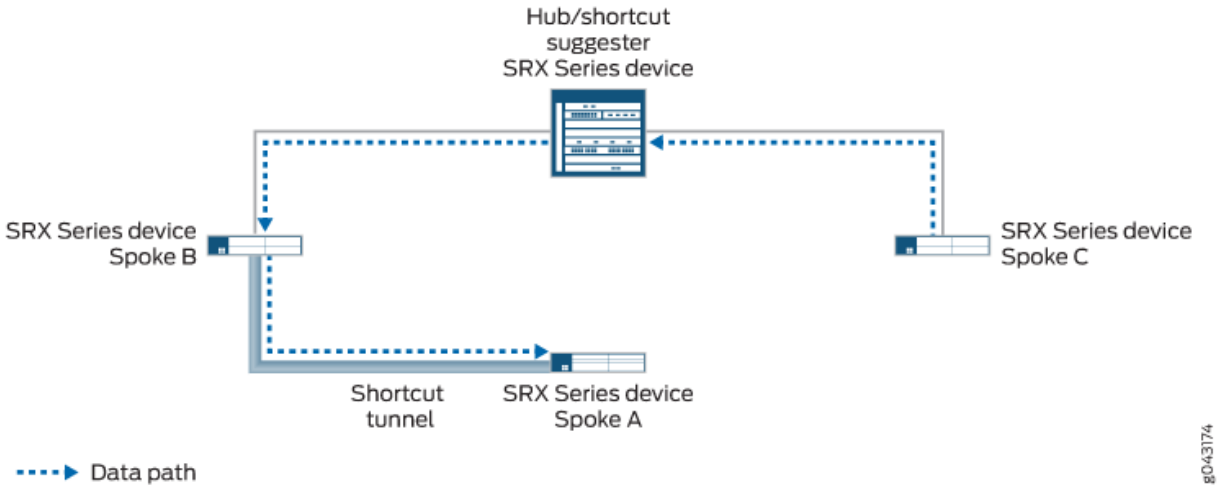
In Figure 60 on page 938, static tunnels exist between the hub and each of the spokes. OSPF adjacencies are established between the hub and spokes. Spoke A also has a shortcut tunnel with Spoke B and OSPF adjacencies are established between the spokes. The hub (the shortcut suggester) recognizes that 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 60: Static Tunnels and Shortcut Tunnel Established in Hub-and-Spoke Network**




In Figure 61 on page 938, 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.

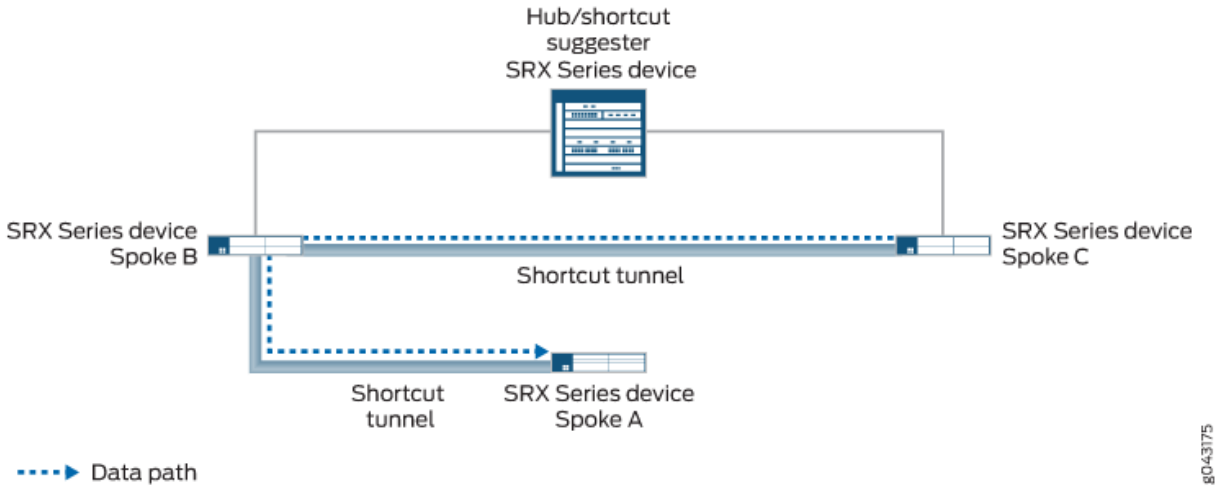
**Figure 61: Traffic Path from Spoke C to Spoke A**




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

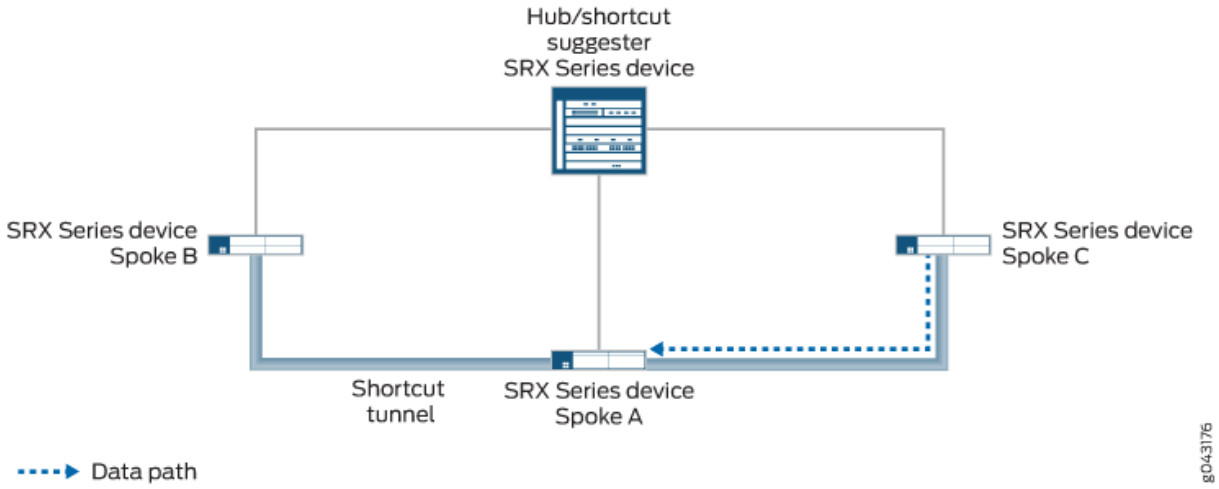
shortcut tunnel between Spoke C and Spoke B, and then through the shortcut tunnel between Spoke B and Spoke A (see Figure 62 on page 939).

**Figure 62: 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 63 on page 940). 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 63: 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.

**SEE ALSO**

Understanding Hub-and-Spoke VPNs | **198**

## Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels

**IN THIS SECTION**


- Requirements | **941**
- Overview | **941**
- Configuration | **945**
- Verification | **970**

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 SRX Series Firewalls as AutoVPN hub and spokes.
- Junos OS Release 12.3X48-D10 or later releases 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. See "Understanding Local Certificate Requests" on page 53.
2. Enroll the digital certificates in each device. See "Example: Loading CA and Local Certificates Manually" on page 64.

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 | **943**

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 84 on page 942 shows the values used in this example.

**Table 84: 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 |
| Encryption algorithm | aes-256-cbc |
| *IPsec policy:* | |

**Table 84: Phase 1 and Phase 2 Options for AutoVPN Hub and Spokes for ADVPN Example *(Continued)***

| Option | Value |
|---|---|
| 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 85 on page 943 shows the IKE gateway configuration for the hub and spokes in this example.

**Table 85: IKE Gateway Configuration for ADVPN Example**

| Option | Hub | Spokes |
|---|---|---|
| Remote IP address | Dynamic | Spoke 1: 11.1.1.1<br>Spoke 2: 11.1.1.1 |
| Local IP address | 11.1.1.1 | Spoke 1: 21.1.1.2<br>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 |

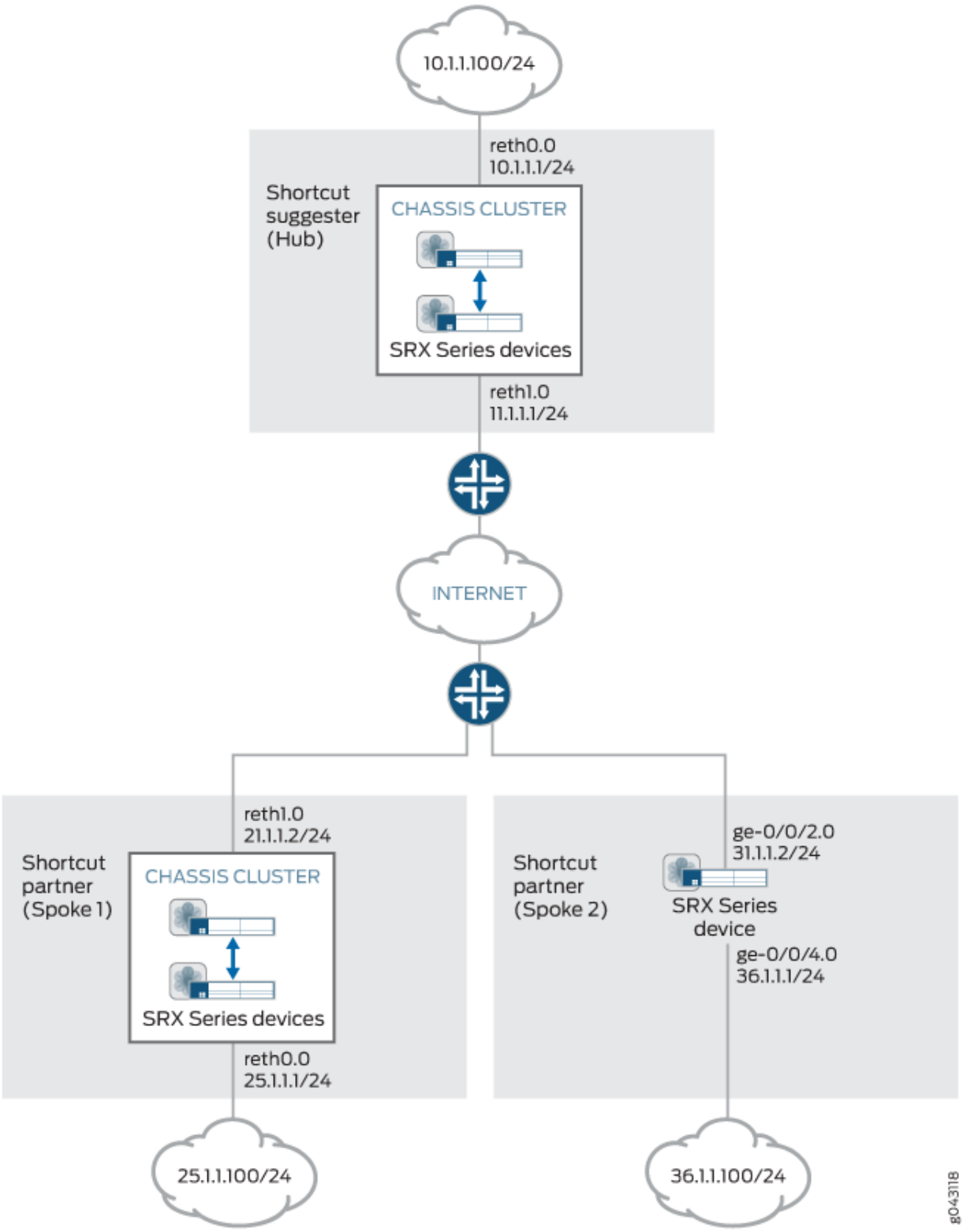
The hub authenticates the spokes’ IKE ID if the subject fields of the spokes’ certificates contain the string “XYZ” in the O field and “Sales” in the OU field.

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. See *Security Policies Overview*.

### Topology

Figure 64 on page 944 shows the SRX Series Firewalls to be configured for this example.

**Figure 64: AutoVPN Deployment with ADVPN**



## Configuration

IN THIS SECTION


- Configuring the Suggester (Hub) | 945
- Configuring the Partner (Spoke 1) | 954
- Configuring the Partner (Spoke 2) | 962


### 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 *Using the CLI Editor in Configuration Mode* in the 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 *Using the CLI Editor in Configuration Mode* in the 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 *Using the CLI Editor in Configuration Mode* in the 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 | 970
- Verifying the Shortcut Tunnel Between Partners | 980


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.

### SEE ALSO

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | 211

## Example: Configuring ADVPN with OSPFv3 for IPv6 Traffic

**IN THIS SECTION**


- Requirements | 993
- Overview | 994
- Configuration | 997
- Verification | 1026


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 SRX Series Firewalls as ADVPN hub and spokes
- Junos OS Release 18.1R1, and later releases.

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 | 996


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 86 on page 994 shows the options used in this example.

**Table 86: 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:* | |

**Table 86: Phase 1 and Phase 2 Options for ADPN Hub and Spoke Basic OSPFv3 Configurations *(Continued)***

| Option | Value |
|---|---|
| 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 87 on page 995 shows the options configured on the hub and on all spokes.

**Table 87: 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 |

**Table 87: ADVPN OSPFv3 Configuration for Hub and All Spokes *(Continued)***

| Option | Hub | All Spokes |
|---|---|---|
| External interface | reth1 | Spoke 1: ge-0/0/0.0<br>Spoke 2: ge-0/0/0.0 |
| *VPN:* | | |
| Bind interface | st0.1 | st0.1 |
| Establish tunnels | (not configured) | establish-tunnels immediately |

Table 88 on page 996 shows the configuration options that are different on each spoke.

**Table 88: 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 |

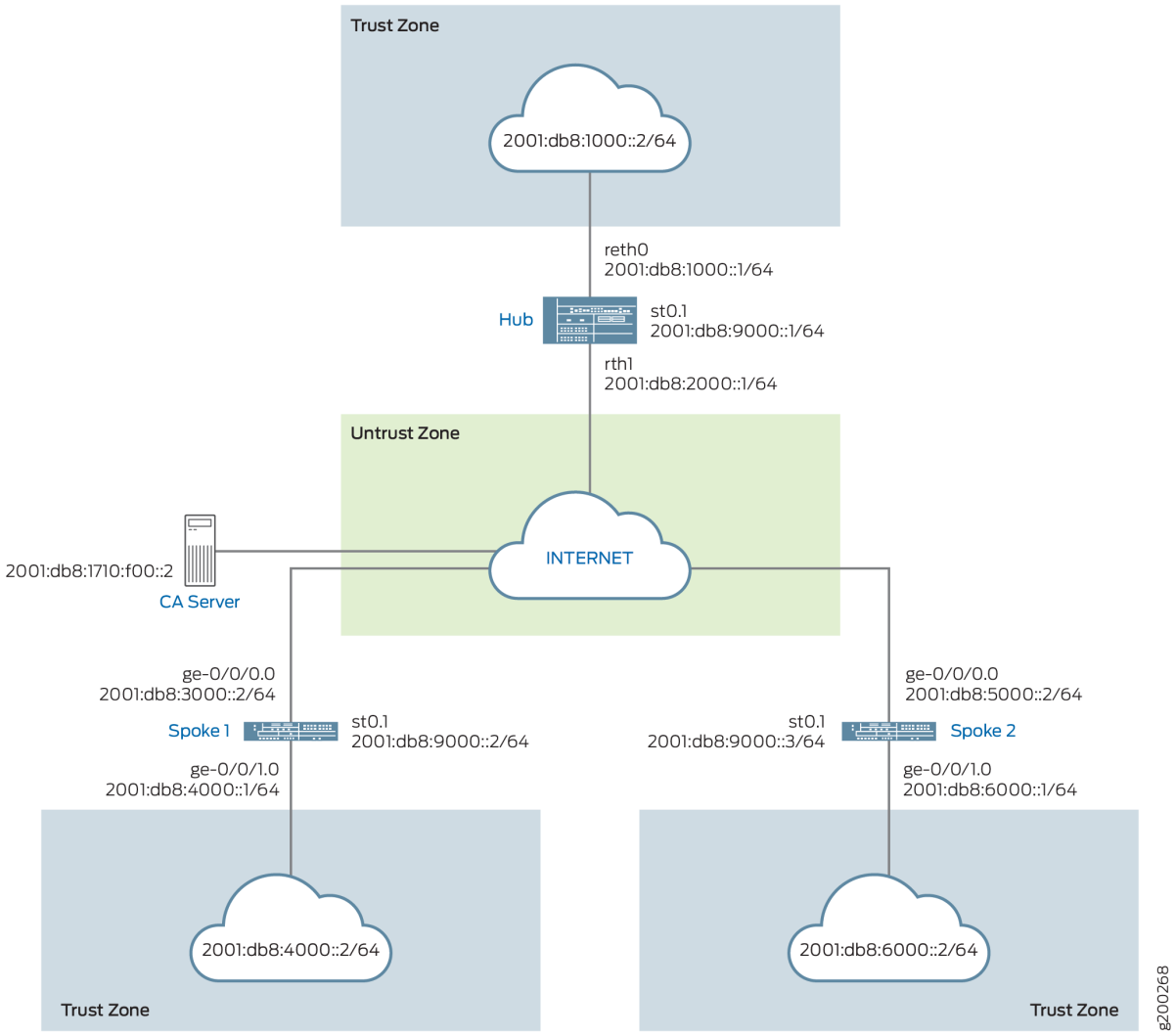
Routing information for all devices is exchanged through the VPN tunnels.

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. See *Security Policies Overview*.

### Topology

Figure 65 on page 997 shows the SRX Series Firewalls to be configured for ADVPN in this example.

**Figure 65: ADVPN Deployment with OSPFv3**




## Configuration

**IN THIS SECTION**


- Enroll Device Certificates with SCEP | **998**
- Configuring the Hub | **1003**
- Configuring Spoke 1 | **1012**
- Configuring Spoke 2 | **1019**

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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_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-0/0/0.0
[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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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_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 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 | **1027**
- Verifying IPsec Status | **1027**
- Verifying IPsec Next-Hop Tunnels | **1028**
- Verifying OSPFv3 | **1029**

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
```

#### SEE ALSO

Example: Configuring a Route-Based VPN | 407

## Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established

**IN THIS SECTION**


- Problem | 1030
- Solution | 1030

## 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
```

### SEE ALSO

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | 211

**Release History Table**

| Release | Description |
|---|---|
| 23.4R1 | Support for ADVPN with the iked process is added in Junos OS Release 23.4R1. |
| 19.2R1 | Starting in Junos OS Release 19.2R1, on SRX300, SRX320, SRX340, SRX345, SRX550, SRX1500, vSRX Virtual Firewall 2.0 (with 2 vCPUs), and vSRX Virtual Firewall 3.0 (with 2 vCPUs) Series devices, Protocol Independent Multicast (PIM) using point-to-multipoint (P2MP) mode supports Auto Discovery VPN in which a new `p2mp` interface type is introduced for PIM. |

| | |
|---|---|
| 18.1R1 | Starting with Junos OS Release 18.1R1, ADVPN supports IPv6 with the kmd process. |

## RELATED DOCUMENTATION

IPv6 IPsec VPNs | **361**

12
CHAPTER

# AutoVPN

AutoVPN on Hub-and-Spoke Devices | 1033

# AutoVPN on Hub-and-Spoke Devices

IN THIS SECTION


- Understanding AutoVPN | 1033
- Understanding Spoke Authentication in AutoVPN Deployments | 1039
- AutoVPN Configuration Overview | 1042
- Example: Configuring Basic AutoVPN with iBGP | 1043
- Example: Configuring Basic AutoVPN with iBGP for IPv6 Traffic | 1080
- Example: Configuring AutoVPN with iBGP and ECMP | 1118
- Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels | 1151
- Example: Configuring Basic AutoVPN with OSPF | 1188
- Example: Configuring AutoVPN with OSPFv3 for IPv6 Traffic | 1221
- Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors | 1258
- Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors | 1280
- Example: Configuring AutoVPN with Pre-Shared Key | 1308


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.

## Understanding AutoVPN

IN THIS SECTION


- Secure Tunnel Modes | 1034
- Authentication | 1035
- Configuration and Management | 1037
- Understanding AutoVPN Limitations | 1037

- Understanding AutoVPN with Traffic Selectors | **1038**


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. Starting with Junos OS Release 17.4R1, IPv6 address is supported on AutoVPN.
- 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 89 on page 1034 compares AutoVPN point-to-point and point-to-multipoint secure tunnel interface modes.

**Table 89: 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 |

**Table 89: Comparison Between AutoVPN Point-to-Point and Point-to-Multipoint Secure Tunnel Modes *(Continued)***

| Point-to-Point Mode | Point-to-Multipoint Mode |
|---|---|
| Allows spoke devices to be SRX Series or third-party devices. | This mode is only supported with SRX Series 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. See "Understanding Spoke Authentication in AutoVPN Deployments" on page 1039.

Starting in Junos OS Release 21.2R1, SRX5000 line with SPC3 card and vSRX Virtual Firewall running iked process supports AutoVPN with seeded preshared key.

**NOTE**: The SRX5000 line with the SPC3 card and vSRX Virtual Firewall supports AutoVPN with PSK, only if you install the `junos-ike` package.

We support AutoVPN with the following two options:

- **Auto-VPN seeded PSK**: Multiple peers connecting to same gateway having different pre-shared key.
- **Auto-VPN 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.

  See *policy*.

- **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 *general-ikeid*.

The SRX5000 line with SPC3 card and vSRX Virtual Firewall running iked process (with the `junos-ike` package) supports the following IKE modes:

**Table 90: AutoVPN PSK Support**

| IKE Mode | SRX5000 line with SPC3 Card and vSRX Virtual Firewall running iked process | |
|---|---|---|
| | Shared PSK | Seeded-PSK |
| IKEv2 | Yes | Yes |
| IKEv2 with any-`remote-id` | Yes | Yes |
| IKEv1 Aggressive Mode | Yes | Yes |

**Table 90: AutoVPN PSK Support *(Continued)***

| IKE Mode | SRX5000 line with SPC3 Card and vSRX Virtual Firewall running iked process | |
|---|---|---|
| | Shared PSK | Seeded-PSK |
| 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 |

See "Example: Configuring AutoVPN with Pre-Shared Key" on page 1308.

## Configuration and Management

AutoVPN is configured and managed on SRX Series Firewalls using the CLI. Multiple AutoVPN hubs can be configured on a single SRX Series Firewall. The maximum number of spokes supported by a configured hub is specific to the model of the SRX Series Firewall.

## 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.
- 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 non-SRX Series Firewalls.
- 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.

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.

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 non-SRX Series Firewalls; however, note the following differences:
  - In IKEv2, a non-SRX Series spoke can propose multiple traffic selectors in a single SA negotiation. This is not supported on SRX Series Firewalls and the negotiation is rejected.
  - A non-SRX Series spoke can identify specific ports or protocols for traffic selector use. Ports and protocols are not supported with traffic selectors on SRX Series Firewalls and the negotiation is rejected.

### SEE ALSO

Understanding Spoke Authentication in AutoVPN Deployments | 1039

Understanding Traffic Selectors in Route-Based VPNs | 536

Example: Configuring Traffic Selectors in a Route-Based VPN | 543

# Understanding Spoke Authentication in AutoVPN Deployments

IN THIS SECTION


- Group IKE ID Configuration on the Hub | 1039
- Excluding a Spoke Connection | 1041


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. The wildcard match supports only one value per field (for example, `ou=eng` or `ou=sw` but not `ou=eng,ou=sw`). The order of the fields is inconsequential.

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 "Understanding Online Certificate Status Protocol and Certificate Revocation Lists" on page 67 and "Understanding Certificate Authority Profiles" on page 46.

SEE ALSO

IPsec VPN with Autokey IKE Configuration Overview | 202

## 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 an SRX Series 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 SRX Series 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. For more details, see Example: Configuring the Multipoint VPN Configuration with Next-Hop Tunnel Binding.

SEE ALSO

Understanding Traffic Selectors in Route-Based VPNs | 536

# Example: Configuring Basic AutoVPN with iBGP

IN THIS SECTION


- Requirements | 1043
- Overview | 1044
- Configuration | 1047
- Verification | 1076


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 1055 to configure the hub, " spoke1" on page 1063 to configure the spoke1, and the " spoke2" on page 1070 to configure the spoke2.

## Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes
- Junos OS Release 12.1X44-D10 and later that support 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 | **1046**

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 91 on page 1044 shows the options used in this example.

**Table 91: 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 | 2 |
| Authentication algorithm | SHA-1 |
| Encryption algorithm | AES 128 CBC |
| *IKE policy:* | |
| Mode | Main |

**Table 91: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke 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 92 on page 1045 shows the options configured on the hub and on all spokes.

**Table 92: 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 |

**Table 92: AutoVPN 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<br>Spoke 2: ge-0/0/1.0 |
| *VPN:* | | |
| Bind interface | st0.0 | st0.0 |
| Establish tunnels | (not configured) | Immediately on configuration commit |

Table 93 on page 1046 shows the configuration options that are different on each spoke.

**Table 93: 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 |

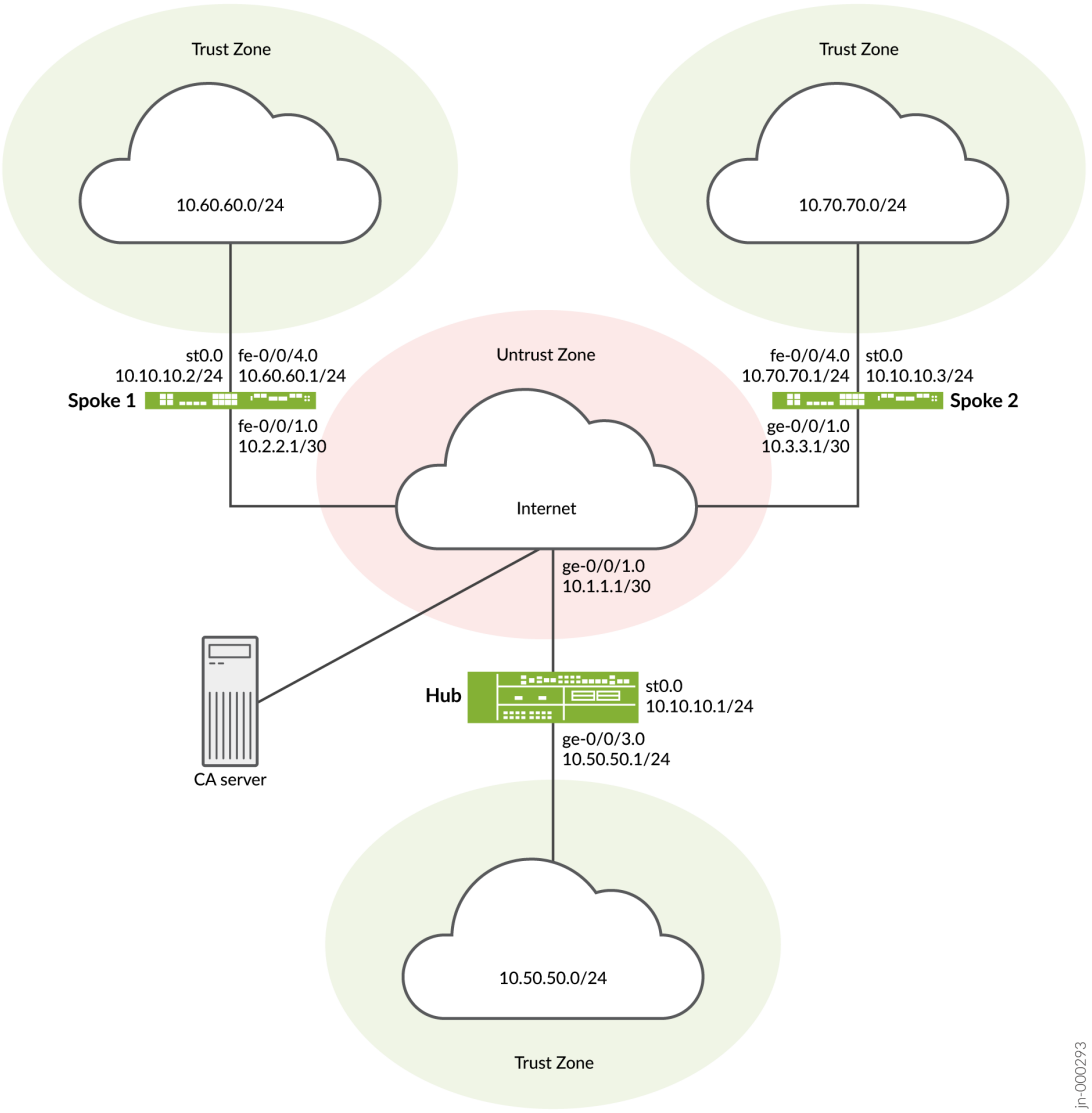
Routing information for all devices is exchanged through the VPN tunnels.

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. See *Security Policies Overview*.

### Topology

Figure 66 on page 1047 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

**Figure 66: Basic AutoVPN Deployment with iBGP**




## Configuration

**IN THIS SECTION**


- Enroll Device Certificates with SCEP | **1048**
- Configuring the Hub | **1053**
- Configuring Spoke 1 | **1061**

- Configuring Spoke 2 | **1069**


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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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 10
```

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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 10
```

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 | **1076**
- Verifying IPsec Phase 2 Status | **1077**
- Verifying IPsec Next-Hop Tunnels | **1077**
- Verifying BGP | **1078**
- Verifying Learned Routes | **1079**


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

Route-Based IPsec VPNs | 406

Routing Protocols Overview

# Example: Configuring Basic AutoVPN with iBGP for IPv6 Traffic

IN THIS SECTION


- Requirements | 1080
- Overview | 1080
- Configuration | 1084
- Verification | 1115


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 1043.

## Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes.
- Junos OS Release 18.1R1 and later releases.

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 | 1083

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 94 on page 1081 shows the options used in this example.

**Table 94: 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 |
| *IKE policy:* | |
| Mode | Main |
| *IPsec proposal:* | |
| Protocol | ESP |
| Lifetime Seconds | 3000 |
| Encryption algorithm | AES 256 GCM |

**Table 94: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations *(Continued)***

| Option | Value |
|---|---|
| *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 95 on page 1082 shows the options configured on the hub and on all spokes.

**Table 95: 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 |
| 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<br>Spoke 2: ge-0/0/0.0 |
| *VPN:* | | |
| Bind interface | st0.1 | st0.1 |
| Establish tunnels | (not configured) | establish-tunnels on-traffic |

Table 96 on page 1083 shows the configuration options that are different on each spoke.

**Table 96: 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 |

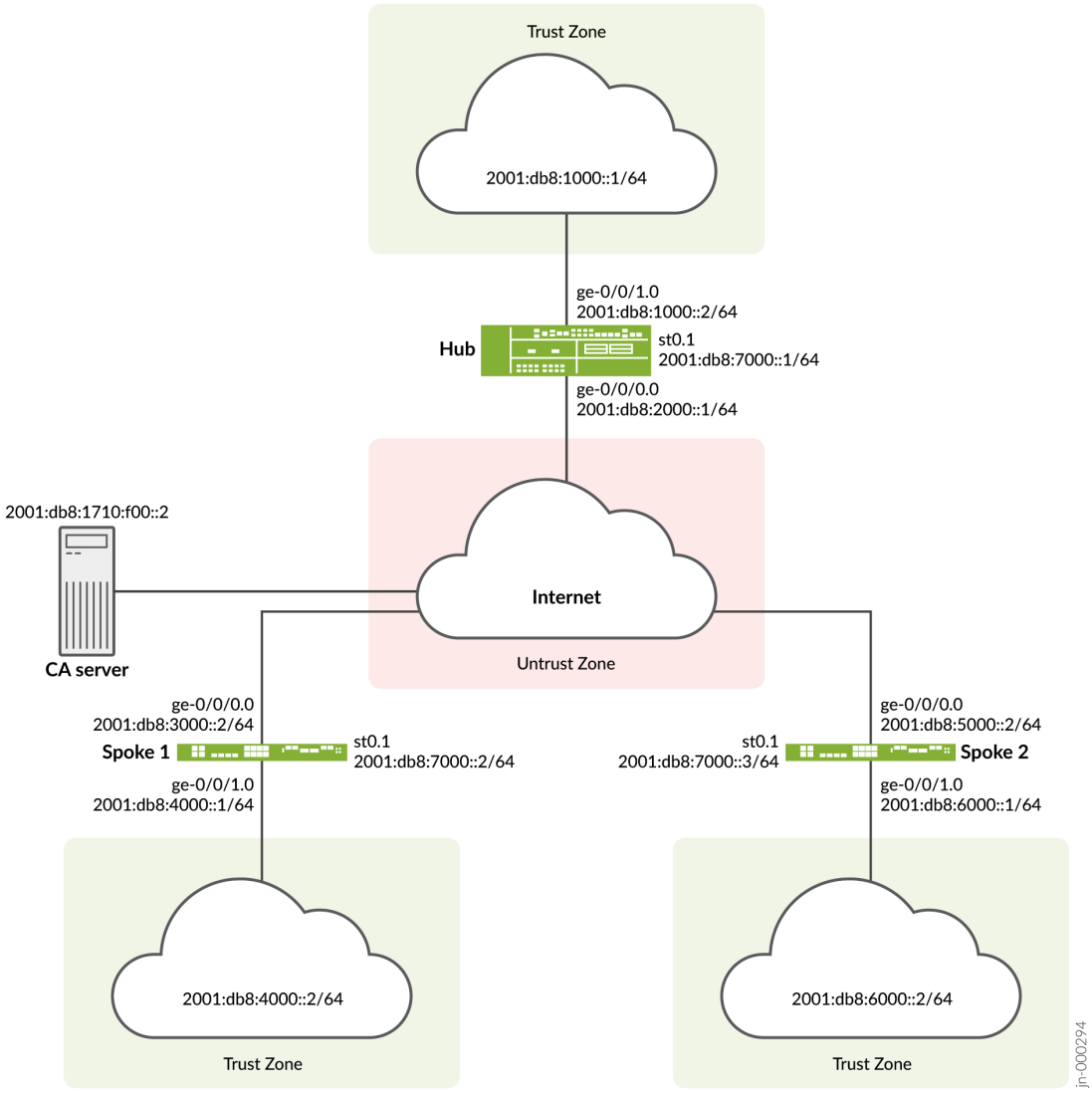
Routing information for all devices is exchanged through the VPN tunnels.

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. See *Security Policies Overview*.

### Topology

Figure 67 on page 1084 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

**Figure 67: Basic AutoVPN Deployment with iBGP**




## Configuration

**IN THIS SECTION**


- Enroll Device Certificates with SCEP | **1085**
- Configuring the Hub | **1090**
- Configuring Spoke 1 | **1099**

- Configuring Spoke 2 | **1107**


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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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 | **1115**
- Verifying IPsec Status | **1116**
- Verifying IPsec Next-Hop Tunnels | **1116**
- Verifying BGP | **1117**


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

Example: Configuring a Route-Based VPN | 407

Routing Protocols Overview

# Example: Configuring AutoVPN with iBGP and ECMP

IN THIS SECTION

- Requirements | 1118
- Overview | 1119
- Configuration | 1122
- Verification | 1147

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 1043.

## Requirements

This example uses the following hardware and software components:

- Two supported SRX Series Firewalls as AutoVPN hub and spoke
- Junos OS Release 12.1X44-D10 and later that support 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.

## Overview

IN THIS SECTION

- Topology | 1121

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 97 on page 1119 shows the options used in this example.

**Table 97: 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 97: 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 98 on page 1120 shows the options configured on the hub and on the spoke.

**Table 98: AutoVPN iBGP ECMP Configuration for Hub and Spoke 1**

| Option | Hub | Spoke 1 |
|---|---|---|
| *IKE gateway:* | | |
| Remote IP address | hub-to-spoke-gw-1: Dynamic<br>hub-to-spoke-gw-2: Dynamic | spoke-to-hub-gw-1: 10.1.1.1<br>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<br>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<br>spoke-to-hub-gw-2: DN on the hub's certificate |

**Table 98: 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<br>hub-to-spoke-gw-2: ge-0/0/2.0 | spoke-to-hub-gw-1: fe-0/0/1.0<br>spoke-to-hub-gw-2: fe-0/0/2.0 |
| *VPN:* | | |
| Bind interface | hub-to-spoke-vpn-1: st0.0<br>hub-to-spoke-vpn-2: st0.1 | spoke-to-hub-1: st0.0<br>spoke-to-hub-2: st0.1 |
| Establish tunnels | (not configured) | Immediately on configuration commit |

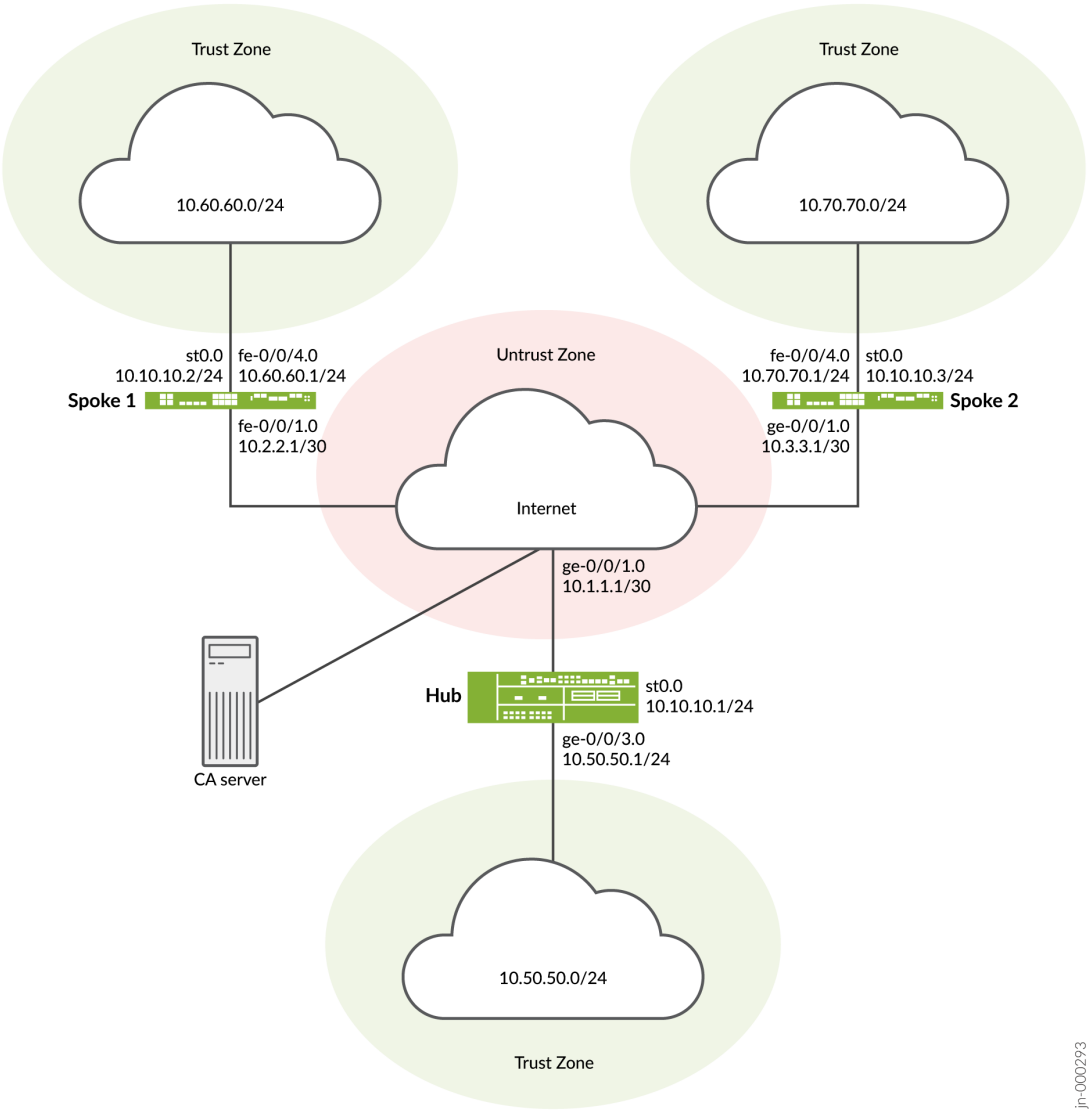
Routing information for all devices is exchanged through the VPN tunnels.

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. See *Security Policies Overview*.

### Topology

Figure 68 on page 1122 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

**Figure 68: AutoVPN Deployment with iBGP and ECMP**




## Configuration

**IN THIS SECTION**


- Enroll Device Certificates with SCEP | **1123**
- Configuring the Hub | **1128**
- Configuring Spoke 1 | **1138**

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> rrequest 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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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 | **1147**
- Verifying IPsec Phase 2 Status | **1148**
- Verifying IPsec Next-Hop Tunnels | **1148**
- Verifying BGP | **1149**
- Verifying Learned Routes | **1149**
- Verifying Route Installation in Forwarding Table | **1151**


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
```

#### SEE ALSO

Route-Based IPsec VPNs | 406

## Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels

IN THIS SECTION

Requirements | 1152

- Overview | **1152**
- Configuration | **1156**
- Verification | **1181**


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 1043.

## Requirements

This example uses the following hardware and software components:

- Two supported SRX Series Firewalls as AutoVPN hub and spoke
- Junos OS Release 12.1X44-D10 and later that support 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.

## Overview

**IN THIS SECTION**


- Topology | **1155**


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 99 on page 1153 shows the options used in this example.

**Table 99: 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 100 on page 1154 shows the options configured on the hub and on the spoke.

**Table 100: 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<br>hub-to-spoke-gw-2: Dynamic | spoke-to-hub-gw-1: 10.1.1.1<br>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<br>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<br>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<br>hub-to-spoke-gw-2: ge-0/0/2.0 | spoke-to-hub-gw-1: fe-0/0/1.0<br>spoke-to-hub-gw-2: fe-0/0/2.0 |
| *VPN:* | | |
| Bind interface | hub-to-spoke-vpn-1: st0.0<br>hub-to-spoke-vpn-2: st0.1 | spoke-to-hub-1: st0.0<br>spoke-to-hub-2: st0.1 |
| VPN monitor | hub-to-spoke-vpn-1: ge-0/0/1.0 (source interface)<br>hub-to-spoke-vpn-2: ge-0/0/2.0 (source interface) | spoke-to-hub-1: 10.1.1.1 (destination IP)<br>spoke-to-hub-2: 10.1.2.1 (destination IP) |
| Establish tunnels | (not configured) | Immediately on configuration commit |

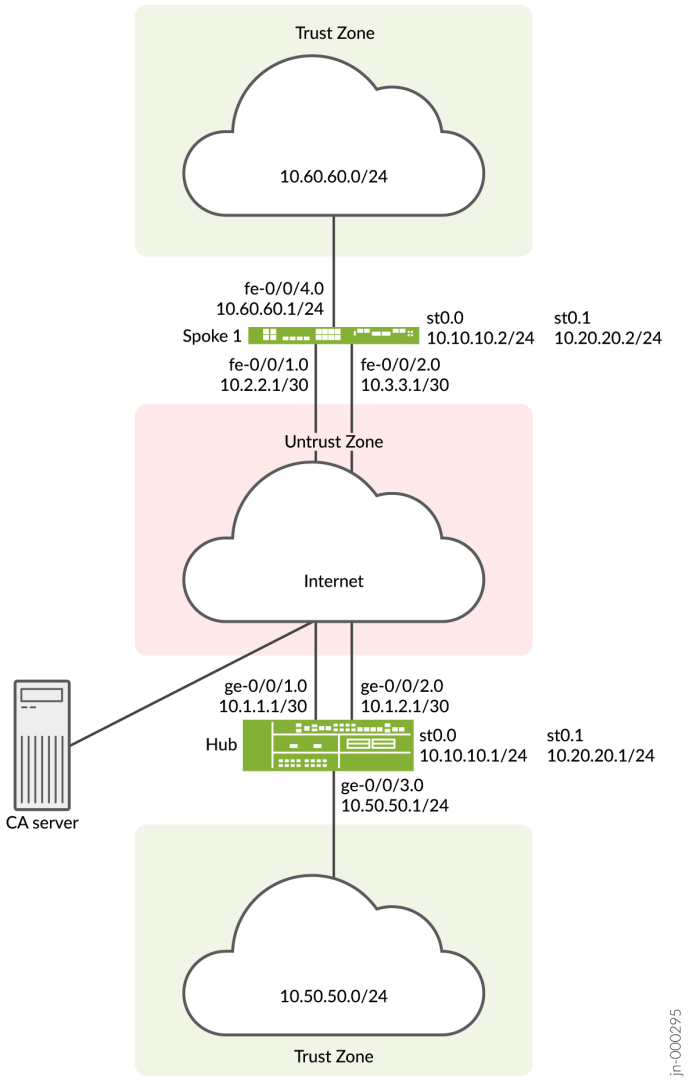
Routing information for all devices is exchanged through the VPN tunnels.

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. See *Security Policies Overview*.

### Topology

Figure 69 on page 1155 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

**Figure 69: 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 | **1156**
- Configuring the Hub | **1162**
- Configuring Spoke 1 | **1171**


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> rrequest 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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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) | **1181**
- Verifying IPsec Phase 2 Status (Both Tunnels Are Up) | **1182**
- Verifying IPsec Next-Hop Tunnels (Both Tunnels Are Up) | **1183**
- Verifying BGP (Both Tunnels Are Up) | **1183**
- Verifying Learned Routes (Both Tunnels Are Up) | **1184**
- Verifying IKE Phase 1 Status (Primary Tunnel Is Down) | **1185**
- Verifying IPsec Phase 2 Status (Primary Tunnel Is Down) | **1185**
- Verifying IPsec Next-Hop Tunnels (Primary Tunnel Is Down) | **1186**
- Verifying BGP (Primary Tunnel Is Down) | **1186**
- Verifying Learned Routes (Primary Tunnel Is Down) | **1187**


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
```

SEE ALSO

Route-Based IPsec VPNs | 406

# Example: Configuring Basic AutoVPN with OSPF

IN THIS SECTION


- Requirements | 1188
- Overview | 1189
- Configuration | 1192
- Verification | 1218


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 1043.

## Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes
- Junos OS Release 12.1X44-D10 and later that support 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.

## Overview

IN THIS SECTION

- Topology | 1191

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 101 on page 1189 shows the options used in this example.

**Table 101: 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 101: 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 102 on page 1190 shows the options configured on the hub and on all spokes.

**Table 102: 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 102: 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<br>Spoke 2: ge-0/0/1.0 |
| *VPN:* | | |
| Bind interface | st0.0 | st0.0 |
| Establish tunnels | (not configured) | Immediately on configuration commit |

Table 103 on page 1191 shows the configuration options that are different on each spoke.

**Table 103: 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 |

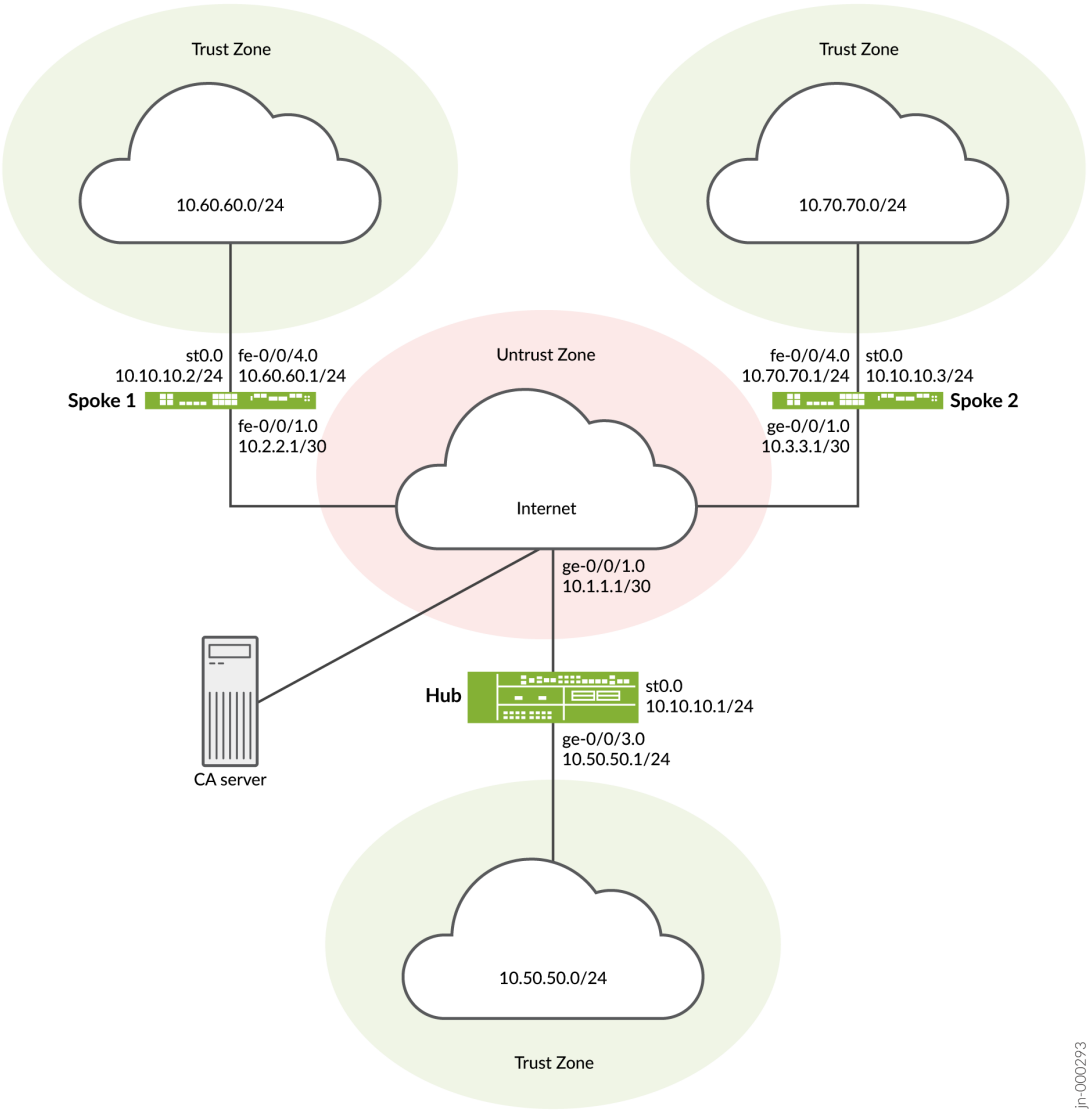
Routing information for all devices is exchanged through the VPN tunnels.

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. See *Security Policies Overview*.

### Topology

Figure 70 on page 1192 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

**Figure 70: Basic AutoVPN Deployment with OSPF**




## Configuration

**IN THIS SECTION**


- Enroll Device Certificates with SCEP | **1193**
- Configuring the Hub | **1198**
- Configuring Spoke 1 | **1205**

- Configuring Spoke 2 | **1211**


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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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 | **1218**
- Verifying IPsec Phase 2 Status | **1219**
- Verifying IPsec Next-Hop Tunnels | **1219**
- Verifying OSPF | **1220**
- Verifying Learned Routes | **1220**


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
```

### SEE ALSO

Route-Based IPsec VPNs | 406

## Example: Configuring AutoVPN with OSPFv3 for IPv6 Traffic

IN THIS SECTION


- Requirements | 1222
- Overview | 1222
- Configuration | 1226
- Verification | 1254

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 1043.

## Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes.
- Junos OS Release 18.1R1 and later releases.

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 | **1225**

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 104 on page 1223 shows the options used in this example.

**Table 104: Phase 1 and Phase 2 Options for AutoVPN 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 |
| *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 105 on page 1224 shows the options configured on the hub and on all spokes.

**Table 105: 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<br>Spoke 2: ge-0/0/0.0 |
| *VPN:* | | |
| Bind interface | st0.1 | st0.1 |
| Establish tunnels | (not configured) | Immediately on configuration commit |

Table 106 on page 1224 shows the configuration options that are different on each spoke.

**Table 106: 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 |

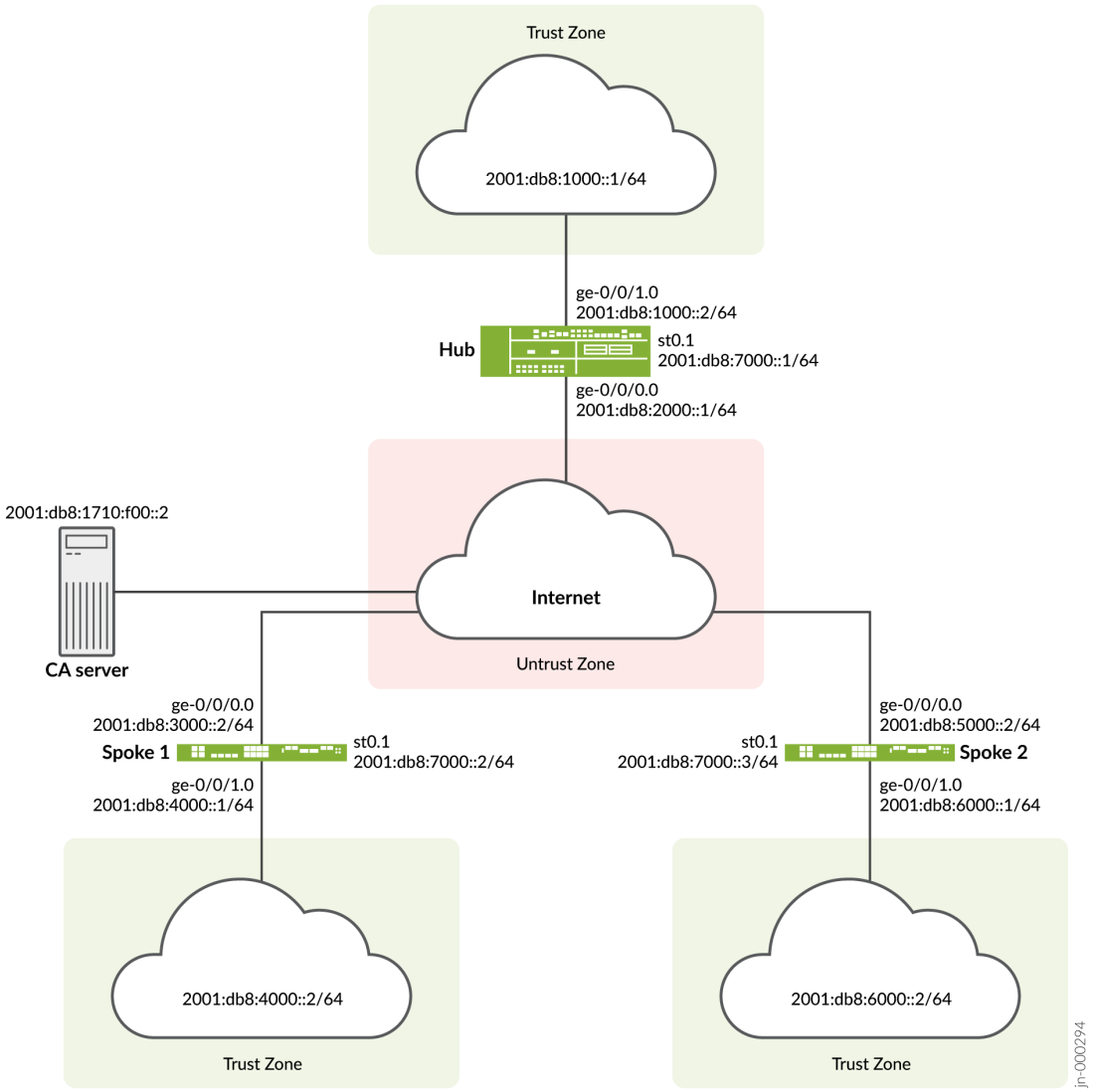
Routing information for all devices is exchanged through the VPN tunnels.

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. See *Security Policies Overview*.

### Topology

Figure 71 on page 1225 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

**Figure 71: Basic AutoVPN Deployment with OSPFv3**



## Configuration

IN THIS SECTION


- Enroll Device Certificates with SCEP | 1226
- Configuring the Hub | 1231
- Configuring Spoke 1 | 1239
- Configuring Spoke 2 | 1247


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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

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 | **1255**
- Verifying IPsec Status | **1255**
- Verifying IPsec Next-Hop Tunnels | **1256**
- Verifying OSPFv3 | **1257**


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
```

**SEE ALSO**

Example: Configuring a Route-Based VPN | 407

# Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors

**IN THIS SECTION**


- Requirements | 1258
- Overview | 1259
- Configuration | 1262
- Verification | 1276


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 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 1043.

## Requirements

This example uses the following hardware and software components:

- Two SRX Series Firewalls connected and configured in a chassis cluster. The chassis cluster is the AutoVPN hub.
- An SRX Series Firewall configured as an AutoVPN spoke.
- Junos OS Release 12.3X48-D10 or later.

- 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. See "Understanding Local Certificate Requests" on page 53.
- Enroll the digital certificates in each device. See "Example: Loading CA and Local Certificates Manually" on page 64.

## Overview

**IN THIS SECTION**

- Topology | **1261**

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 107 on page 1259 shows the values used in this example:

**Table 107: 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 |

**Table 107: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors *(Continued)***

| Option | Value |
|---|---|
| Authentication algorithm | sha-1 |
| Encryption algorithm | aes-256-cbc |
| *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 |

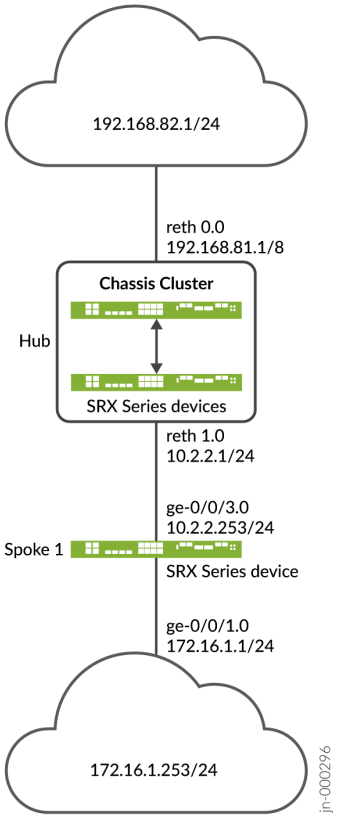
**Table 107: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors *(Continued)***

| Option | Value |
|---|---|
| Lifetime | 3600 seconds<br>150,000 kilobytes |
| *IPsec policy:* | |
| Perfect Forward Secrecy (PFS) group | group5 |

### Topology

Figure 72 on page 1262 shows the SRX Series Firewalls to be configured for this example.

**Figure 72: AutoVPN with Traffic Selectors**




## Configuration

**IN THIS SECTION**


- Configuring the Hub | **1263**
- Configuring the Spoke | **1270**

### 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
```

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. 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 *Using the CLI Editor in Configuration Mode* in the 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 *Using the CLI Editor in Configuration Mode* in the 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 | 1276
- Verifying Traffic Selectors | 1279


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 SRX Series hub).

SEE ALSO

Understanding Traffic Selectors in Route-Based VPNs | 536

# Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors

IN THIS SECTION


- Requirements | 1280
- Overview | 1281
- Configuration | 1284
- Verification | 1304


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 SRX Series 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 SRX Series 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 1043.

## Requirements

This example uses the following hardware and software components:

- Two SRX Series Firewalls connected and configured in a chassis cluster. The chassis cluster is AutoVPN hub A.
- An SRX Series Firewall configured as AutoVPN hub B.
- Junos OS Release 12.3X48-D10 or later.

- 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. See "Understanding Local Certificate Requests" on page 53.
- Enroll the digital certificates in each device. See "Example: Loading CA and Local Certificates Manually" on page 64.

This example uses the BGP dynamic routing protocol to advertise routes toward the eNodeB devices to the core network.

## Overview

**IN THIS SECTION**

- Topology | **1283**

In this example, two AutoVPN hubs are configured with multiple traffic selectors on SRX Series 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 108 on page 1281 shows the values used in this example:

**Table 108: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs**

| Option | Value |
|---|---|
| *IKE proposal:* | |
| Authentication method | rsa-signatures |

**Table 108: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs *(Continued)***

| Option | Value |
|---|---|
| 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 |
| Version | v2-only |
| *IPsec proposal:* | |
| Protocol | esp |
| Authentication algorithm | hmac-sha1-96 |
| Encryption algorithm | aes-256-cbc |

**Table 108: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs *(Continued)***

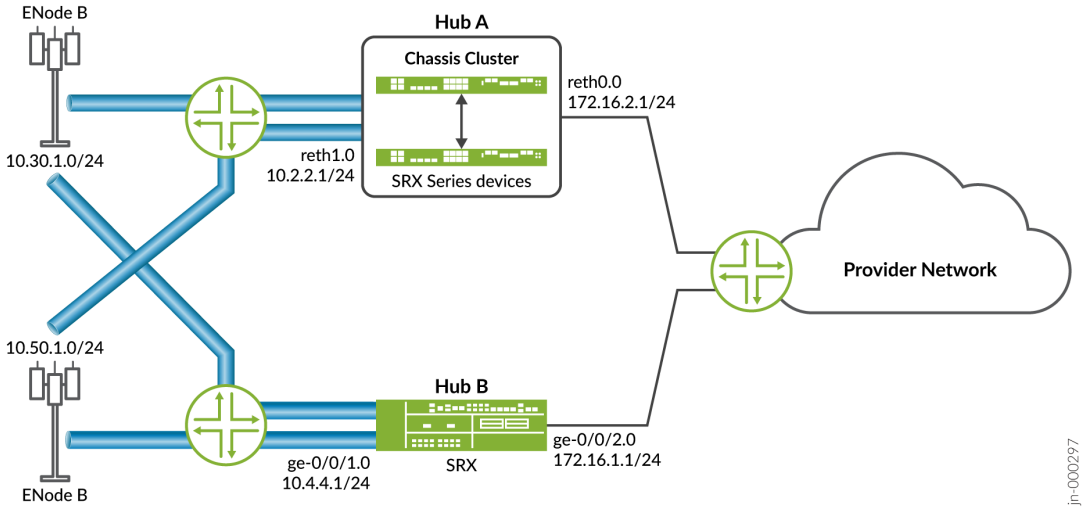
| Option | Value |
|---|---|
| *IPsec policy:* | |
| Perfect Forward Secrecy (PFS) group | group5 |

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. See *Security Policies Overview*. For simplicity, the configuration on the SRX Series Firewalls allows all types of inbound traffic; this configuration is not recommended for production deployments.

### Topology

Figure 73 on page 1283 shows the SRX Series Firewalls to be configured for this example.

**Figure 73: Georedundant IPsec VPN Gateways to eNodeB Devices**



## Configuration

IN THIS SECTION


- Configuring Hub A | 1284
- Configuring Hub B | 1294
- Configuring the eNodeB (Sample Configuration) | 1303


### 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 *Using the CLI Editor in Configuration Mode* in the 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 *Using the CLI Editor in Configuration Mode* in the 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
   - SRX Series IKE identity information and public IP address
   - Phase 1 and Phase 2 proposals that match the configurations on the SRX Series 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 | **1305**
- Verifying Traffic Selectors | **1306**

- Verifying ARI Routes | **1306**

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 SRX Series 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 SRX Series hub. These routes are reachable through the st0.1 interface.

**SEE ALSO**

Understanding Traffic Selectors in Route-Based VPNs | 536

## Example: Configuring AutoVPN with Pre-Shared Key

**IN THIS SECTION**


- Requirements | 1308
- Configure different IKE preshared key | 1309
- Configure same IKE preshared key | 1311


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.

### Requirements

This example uses the following hardware and software components:

- MX240, MX480, and MX960 with MX-SPC3 and Junos OS Release 21.1R1 that support AutoVPN
- or SRX5000 line with SPC3 and Junos OS Release 21.2R1 that support AutoVPN

- or vSRX Virtual Firewall running iked process (with the `junos-ike` package) and Junos OS Release 21.2R1 that support 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 1309) 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;
        }
    }
```

**Release History Table**

| Release | Description |
|---|---|
| 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

Monitoring VPN Traffic

13

CHAPTER

# Remote Access VPN

# Juniper Secure Connect

IN THIS SECTION


- Benefits of Juniper Secure Connect | 1317
- What's Next | 1317


Read this topic to get an overview about Juniper Secure Connect solution.

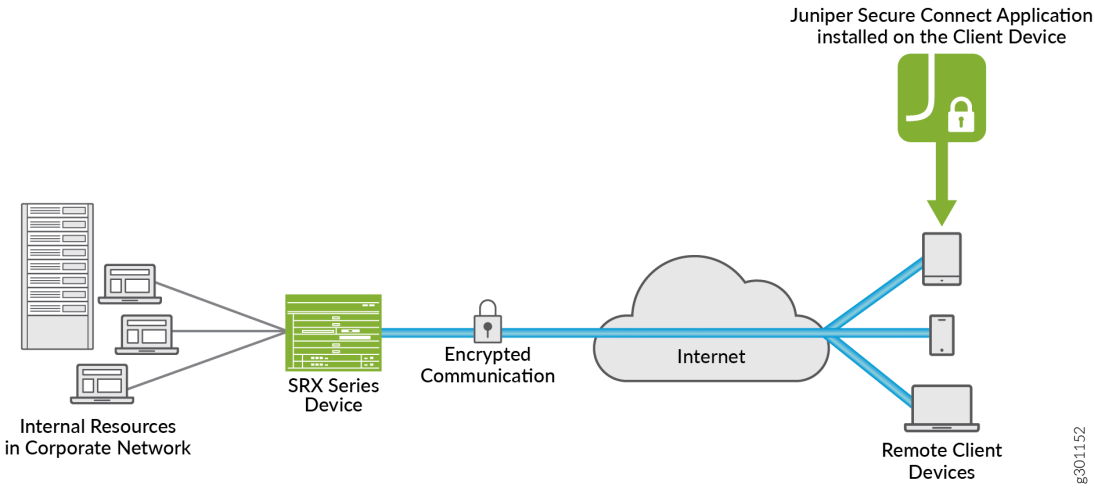
Juniper Secure Connect is a client-based SSL-VPN application that allows you to securely connect and access protected resources on your network. This application when combined with SRX Series Firewalls helps organizations quickly achieve dynamic, flexible, and adaptable connectivity from devices anywhere across the globe. Juniper Secure Connect extends visibility and enforcement from client to cloud using secure VPN connections.

Juniper Secure Connect application includes:

- SRX Series 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 SRX Series Firewall to gain access to the protected resources in the network.

Figure 74 on page 1316 illustrates the Juniper Secure Connect remote access solution for establishing secure VPN connectivity for remote users at different locations.

**Figure 74: Juniper Secure Connect Remote Access Solution**




To work with Juniper Secure Connect, you need SRX Series Firewall or vSRX Virtual Firewall instance running Junos OS Release 20.3R1 or later. See System Requirements.

**Table 109: 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

We recommend you to use J-Web wizard for Juniper Secure Connect configuration. For details on configuring Juniper Secure Connect, see Juniper Secure Connect Administrator Guide.

See Juniper Secure Connect User Guide for setting up client devices with Juniper Secure Connect application.

See these 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 SRX Series Firewalls, see platform specific Junos-FIPS configuration restrictions on the Juniper Tech Library. Search for the specific SRX Series 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

14

CHAPTER

# Monitoring VPN

VPN Monitoring Overview | 1319

VPN Monitoring Methods | 1320

VPN Alarms, Audits, and Events | 1330

# 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 | **1319**

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 SRX Series Firewalls, SRX1 and SRX2. 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 | 1320

VPN Alarms, Audits, and Events | 1330

# VPN Monitoring Methods

**SUMMARY**

Read this topic to understand multiple ways in which you can monitor the VPN tunnel in an SRX Series Firewall.

**IN THIS SECTION**

- IPsec Datapath Verification | 1321
- Dead Peer Detection | 1323
- VPN Tunnel Monitoring | 1326
- Configure Dead Peer Detection | 1327
- Configure VPN Tunnel Monitoring | 1328

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.

Additionally, you can use the following global VPN features for monitoring:

- VPN peers in a security association (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 Internet Control Message Protocol (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.

## IPsec Datapath Verification

**IN THIS SECTION**


- Why Do You Need IPsec Datapath Verification? | **1321**
- How Does IPsec Datapath Verification Work? | **1322**


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, 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.

- Starting in Junos OS Release 15.1X49-D120, 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 security association 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, then 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? | **1323**
- Configurable DPD Parameters | **1324**


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).

An SRX Series Firewall uses the DPD protocol to detect the liveness in an IPsec VPN connection.

**Figure 75: Message Exchange in DPD Protocol**

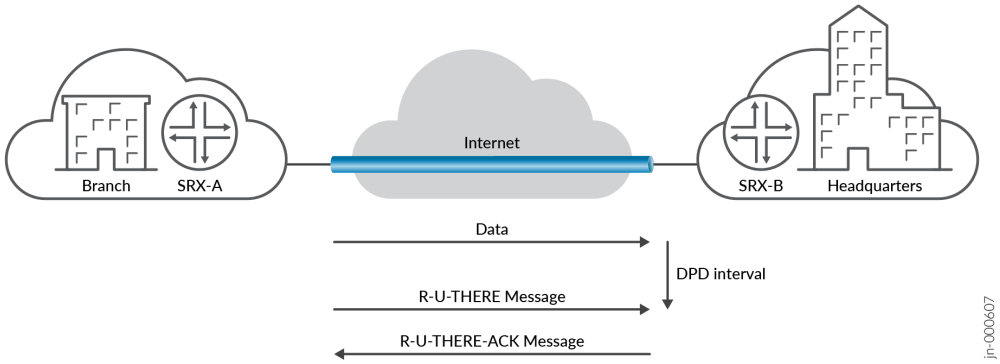



Figure 75 on page 1324 shows the exchange of DPD messages between the IKE peers in an IPsec VPN tunnel. The following events occur when the firewall device performs DPD:

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 security association and all Phase 2 security associations 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 security association.

- 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. Starting with Junos OS Release 15.1X49-D130, we've reduced the permissible interval parameter range at which R-U-THERE messages are sent to the peer device from 10 seconds through 60 seconds to 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 security associations.

- 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 security associations.

- 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 security associations.

- 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 services ipsec-vpn`] hierarchy level to tear down the st0 interface when no phase 1 and phase 2 security associations are available. See establish-tunnels.

- 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 security association. See dead-peer-detection.
- If you configure the IKE gateway with multiple peer IP addresses and DPD, but the connection fails with the current peer's IP address, IKE clears Phase 1 and Phase 2 security associations 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 security associations. If the gateway fails to receive DPD responses for the configured number of consecutive times, it clears the Phase 1 security association and the associated Phase 2 security association (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 1326 .

**SEE ALSO**

dead-peer-detection

gateway (Security IKE)

## VPN Tunnel Monitoring

**IN THIS SECTION**

How Does VPN Tunnel Monitoring Work? | 1327

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 Juniper Networks® SRX Series Firewalls. 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 1323 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 1323 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 1323 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 Juniper Networks® SRX Series Firewalls.

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
```

**NOTE**: The SRX5400, SRX5600, and SRX5800 Firewalls do not support VPN monitoring of an externally connected device (such as a PC). On these devices, the destination for VPN monitoring must be a local interface.

**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.

**Release History Table**

| Release | Description |
|---|---|
| 15.1X49-D130 | Starting with Junos OS Release 15.1X49-D130, the permissible interval parameter range at which R-U-THERE messages are sent to the peer device is reduced from 10 through 60 seconds to 2 seconds through 60 seconds. The minimum threshold parameter should be 3, when the DPD interval parameter is set less than 10 seconds. |
| 15.1X49-D120 | Starting in Junos OS Release 15.1X49-D120, you can configure the size of the packet that is used to verify an IPsec datapath before the `st0` interface is brought up. |

# 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 | **1331**
- VPN Tunnel Events | **1333**
- Configure VPN Alarms | **1334**
- Example: Configure an Audible Alert Notification | **1334**
- Example: Configure Security Alarms Generation | **1336**

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 may notice these failures before and after the tunnel creation.

## 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.
- Junos OS supports security event logging on the SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 Firewalls and on the vSRX Virtual Firewalls.
- 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 its 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 SRX Series 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 | 1335
- Overview | 1335
- Configuration | 1335
- Verification | 1336

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. This feature is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX Virtual Firewall instances.

## 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 | 1335


### 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 | 20

## Example: Configure Security Alarms Generation

**IN THIS SECTION**


- Requirements | 1336
- Overview | 1336
- Configuration | 1337
- Verification | 1340


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. This feature is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX Virtual Firewall instances.

### 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 | 1337


### 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 | 186

15

CHAPTER

# Performance Tuning

VPN Session Affinity | 1342

PowerMode IPsec | 1350

# VPN Session Affinity

IN THIS SECTION


- Understanding VPN Session Affinity | **1342**
- Enabling VPN Session Affinity | **1344**
- Accelerating the IPsec VPN Traffic Performance | **1346**
- IPsec Distribution Profile | **1348**
- Understanding the Loopback Interface for a High Availability VPN | **1349**


The performance of IPsec VPN traffic to minimize packet forwarding overhead can be optimized by enabling VPN session affinity and performance acceleration.

## 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. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices.

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 SRX Series 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.

Junos OS Release 15.1X49-D10 introduces the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) for SRX5400, SRX5600, and SRX5800 devices.

The SRX5K-MPC (IOC2) and the IOC3 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.

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 SRX5400, SRX5600, and SRX5800 devices, 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 SRX Series Firewalls. Enabling VPN session affinity can improve VPN throughput under certain conditions. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices. 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.

Junos OS Release 15.1X49-D10 introduces session affinity support on IOCs (SRX5K-MPC [IOC2], SRX5K-MPC3-100G10G [IOC3], and SRX5K-MPC3-40G10G [IOC3]) and Junos OS Release 12.3X48-D30 introduces session affinity support on IOC2. 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 SRX Series Firewalls. Enabling the VPN performance acceleration can improve the VPN throughput with VPN session affinity enabled. This feature is only supported on SRX5400, SRX5600, and SRX5800 devices.

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). Starting with Junos OS Release 17.4R1, 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 1342.

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

Starting with Junos OS Release 19.2R1, 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 SRX Series 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.

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.

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 SRX Series Firewalls as well as on SRX Series 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.

On SRX5400, SRX5600, and SRX5800 devices -

- For SPC2 based devices running kmd process, if the loopback interface is used as the IKE gateway external interface, configure the interface binding in a redundancy group other than RG0.
- For SPC3 or SPC3+SPC2 based devices running iked process, loopback interface binding to a redundancy group is not required.

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*

**Release History Table**

| Release | Description |
|---|---|
| 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 SRX5400, SRX5600, and SRX5800 devices, the tunnel overhead is calculated according to the negotiated encryption and authentication algorithms on the anchor Services Processing Unit (SPU). |
| 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. |
| Junos OS Release 20.2R | 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. |

RELATED DOCUMENTATION

VPN Support for Inserting Services Processing Cards | **186**

IPsec VPN Configuration Overview | **202**

# PowerMode IPsec

IN THIS SECTION

- Improving IPsec Performance with PowerMode IPsec | **1351**
- Example: Configuring Behavior Aggregate Classifier in PMI | **1358**
- Example: Configuring Behavior Aggregate Classifier in PMI for vSRX Virtual Firewall instances | **1364**
- Example: Configuring and Applying a Firewall Filter for a Multifield Classifier in PMI | **1370**
- Example: Configuring and Applying Rewrite Rules on a Security Device in PMI | **1378**
- Configure IPsec ESP Authentication-only Mode in PMI | **1383**

# Improving IPsec Performance with PowerMode IPsec

IN THIS SECTION


- PMI Processing | 1351
- PMI Statistics | 1351
- Advanced Encryption Standard New Instructions (AES-NI) and Inline Field-Programmable Gate Array (FPGA) | 1352
- Supported and Non-Supported Features for PMI | 1352
- Benefits of PMI | 1356
- Configuring Security Flow PMI | 1356
- Understanding Symmetric Fat IPsec Tunnel | 1357


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.

For SRX4100, SRX4200 devices running Junos OS Release 18.4R1, SRX4600 Series Firewalls running Junos OS Release 20.4R1, and vSRX Virtual Firewall running Junos OS Release 18.3R1 after you enable or disable the PMI, you must reboot the device for the configuration to take effect. However, for SRX5000 line and vSRX Virtual Firewall instances running Junos OS Release 19.2R1, reboot is not required.

## 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)

Starting in Junos OS Release 20.4R1, you can enhance PMI performance by using AES-NI. 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.

If a session is configured with any non-supported features listed in Table 110 on page 1352 and Table 111 on page 1354, 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 110 on page 1352 summarizes the supported and non-supported PMI features on SRX Series Firewalls.

**Table 110: Summary of Supported and Non-supported Features in PMI (SRX Series Firewalls)**

| Supported Features in PMI | Non-Supported Features in PMI |
|---|---|
| Internet Key Exchange (IKE) functionality | IPsec-in-IPsec tunnels |
| AutoVPN with traffic selectors | Layer 4 - 7 applications: application firewall and AppSecure |
| High availability | GPRS tunneling protocol (GTP) and Stream Control Transmission Protocol (SCTP) firewalls |

**Table 110: Summary of Supported and Non-supported Features in PMI (SRX Series Firewalls)** ***(Continued)***

| Supported Features in PMI | Non-Supported Features in PMI |
|---|---|
| IPv6 | Host traffic |
| Stateful firewall | Multicast |
| st0 interface | Nested tunnels |
| Traffic selectors | Screen options |
| NAT-T | DES-CBC encryption algorithm |
| GTP-U scenario with TEID distribution and asymmetric fat tunnel solution | 3DES-CBC encryption algorithm |
| Quality of Service (QoS) | Application Layer Gateway (ALG) |
| First path and fast path processing for fragment handling and unified encryption. | |
| NAT | |
| 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 | |
| AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA2 encryption algorithm | |
| NULL encryption algorithm | |

**Table 110: Summary of Supported and Non-supported Features in PMI (SRX Series Firewalls)** ***(Continued)***

| Supported Features in PMI | Non-Supported Features in PMI |
|---|---|
| | |

Table 111 on page 1354 summarizes the supported and non-supported PMI features on MX-SPC3 services card.

MX-SPC3 services card does not support np-cache and IPsec session-affinity.

**Table 111: Summary of Supported and Non-supported Features in PMI (MX-SPC3 Services Card)**

| Supported Features in PMI | Non-Supported Features in PMI |
|---|---|
| Internet Key Exchange (IKE) functionality | Layer 4 - 7 applications: application firewall, AppSecure, and ALGs |
| AutoVPN with traffic selectors, ADVPN | Multicast |
| High availability | Nested tunnels |
| IPv6 | Screen options |
| Stateful firewall | Application Layer Gateway (ALG) |
| st0 interface | |
| Traffic selectors | |
| Dead Peer Detection (DPD) | |
| Anti-Replay check | |
| NAT | |
| Post/Pre-Fragment | |

**Table 111: Summary of Supported and Non-supported Features in PMI (MX-SPC3 Services Card)** ***(Continued)***

| Supported Features in PMI | Non-Supported Features in PMI |
|---|---|
| incoming clear-text fragments and ESP fragment | |
| 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 | |
| AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA2 encryption algorithm | |
| NULL encryption 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)**
  - Starting in Junos OS Release 19.1R1, Class of Service(CoS) supports configuration of behavior aggregate (BA) classifier, multifield (MF) classifier, and rewrite-rule functions in PMI on SRX5K-SPC3 Services Processing Card (SPC) cards.
  - 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**
  - Junos OS Release 19.3R1 supports options aes-128-cbc, aes-192-cbc, and aes-256-cbc on SRX4100, SRX4200, and vSRX Virtual Firewall in PMI mode to improve IPsec performance, along with the existing support in normal mode.
- **GTP-U**
  - Starting in Junos OS Release 19.2R1, PMI supports GTP-U scenario with TEID distribution and asymmetric fat tunnel solution.

  - Starting in Junos OS Release 19.3R1, GTP-U scenario with TEID distribution and asymmetric fat tunnel solution and Software Receive Side Scaling feature on vSRX Virtual Firewall and vSRX Virtual Firewall.

- **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 SRX5400, SRX5600, SRX5800 line equipped with SRX5K-SPC3 Services Processing Card (SPC), or with vSRX Virtual Firewall.
- **PMI support (vSRX)**
  - Starting in Junos OS Release 19.4R1, 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 IPsec Tunnel

To improve the throughput of IPsec tunnel, you can use fat tunnel technology.

Starting in Junos OS Release 19.4R1, you can configure fat IPsec tunnel on SRX5400, SRX5600, and SRX5800 line with SRX5K-SPC3 service card, and vSRX Virtual Firewall instances.

Starting in Junos OS Release 21.1R1, you can configure fat IPsec tunnel on MX-SPC3 services card.

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 | **176**

flow (Security Flow)

*PMI Flow Based CoS functions for GTP-U*

show security flow pmi statistics

*inline-fpga-crypto*

*distribution-profile*

## Example: Configuring Behavior Aggregate Classifier in PMI

**IN THIS SECTION**


- Requirements | **1358**
- Overview | **1359**
- Configuration | **1359**
- Verification | **1363**


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 112: 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 | **1360**
- Procedure | **1360**

- Results | 1362

### 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 | 1363

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 | 1364
- Overview | 1364
- Configuration | 1365
- Verification | 1369


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 Packet loss priorities(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 113: 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 | 1365
- Procedure | 1366
- Results | 1368


### 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 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 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
```

3. Configure a best-effort 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 expedited forwarding 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 is applied to the Interfaces | 1370

To confirm that the configuration is working properly, perform these tasks:

### Verifying the Classifier is applied to the Interfaces

#### Purpose

Verify that the classifier is configured properly and confirm that the forwarding classes are configured correctly.

#### 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 | **1371**
- Overview | **1371**
- Configuration | **1372**

- Verification | 1377

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 1351.

## 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 | 1372
- Procedure | 1372
- Results | 1375


### 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 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 | 1377


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 | 1378
- Overview | 1378
- Configuration | 1379
- Verification | 1382


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 1351.

## 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 | 1379
- Procedure | 1380
- Results | 1381


### 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 | 1382

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. Starting in Junos OS Release 19.4R1, on SRX5000 line with SRX5K-SPC3 card, 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.

Starting in Junos OS release 22.1R3, we support the PMI express path processing for passthrough ESP traffic on the SRX Series Firewalls.

Before you begin:

- Make sure that the session is PMI capable. See "VPN Session Affinity " on page 1342.

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*

16
CHAPTER

# Troubleshooting

Troubleshoot a Flapping VPN Tunnel | 1386

Troubleshoot a VPN That Is Up But Not Passing Traffic | 1389

Troubleshoot a VPN Tunnel That is Down | 1394

How to Analyze IKE Phase 2 VPN Status Messages | 1396

# Troubleshoot a Flapping VPN Tunnel

IN THIS SECTION


- Problem | 1386
- Diagnosis | 1386


## Problem

### Description

Site-to-site VPN tunnel or remote IPsec VPN tunnel 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 SRX Series Firewall and switch interfaces. To check for errors on the SRX Series 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 SRX Series Firewall a non-Juniper device?
   - Yes: Verify the *proxy-id* value on the SRX Series 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 | 1389
- Solution | 1389


## 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 1390.
- **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 the loopback interface **lo0** as the external interface. Proceed to Step "3" on page 1390.
- **No**: VPN is not using the the loopback interface **lo0** as the external interface. Proceed to Step "4" on page 1390.

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 1390.
- **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 1391. Proceed to Step "8" on page 1392 if it is a policy-based VPN. 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 1391.
- **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 1391, 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 1392.
- **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**: Proceed to Step "8" on page 1392.
   - **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 there is a VPN tunnel security policy to allow traffic: **show security policies**

```
root@siteA# show security policies
...
from-zone trust to-zone untrust {
    policy vpn_egress {
        match {
            source-address local-net;
            destination-address remote-net;
            application any;
        }
        then {
            permit {
                tunnel {                      <----------
                    ipsec-vpn ike-vpn-siteC;  <----------
                }
            }
        }
```

```
    }
}

from-zone untrust to-zone trust {
    policy vpn_ingress {
        match {
            source-address remote-net;
            destination-address local-net;
            application any;
        }
        then {
            permit {
                tunnel {                      <----------
                    ipsec-vpn ike-vpn-siteC;  <----------
                }
            }
        }
    }
}
```

- **Yes**: Proceed to Step "9" on page 1393.
- **No**: Verify the policy-based VPN configuration. See Policy-Based site-to-site VPN .

9. Check whether the traffic is matching in the policies identified in step "7" on page 1392 or step "8" on page 1392: **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 "10" on page 1394.
- **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.

10. Collect logs and flow trace options and open a case with the Juniper Networks support team:

    - See the IPsec VPN policy-based or 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 or Client-to-LAN VPN:

      For branch SRX Series, see KB17220.

      For high-end SRX Series, proceed to Step 2.

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 1389.
- If SA oscillates between active and inactive states, see "Troubleshoot a Flapping VPN Tunnel" on page 1386.

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.

# How to Analyze IKE Phase 2 VPN Status Messages

IN THIS SECTION


- Problem | 1396
- Solution | 1396


## 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 1399.

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.

CHAPTER 17

# Configuration Statements and Operational Commands

Junos CLI Reference Overview | 1402

# Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and 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*
- *CLI Commands*