Junos® OS

VPN Feature Guide for Security Devices

Modified: 2017-05-01

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

Junos® OS VPN Feature Guide for Security Devices

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

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- vSRX
- SRX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see CLI Explorer.

Documentation Conventions

Table 1 on page xxix defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xxx defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		static {
		route default {
		nexthop *address*;
		retain;
		}
		}
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Introduction to IPsec VPNs

IPsec VPN Overview

Supported Platforms SRX Series, vSRX

A virtual private network (VPN) provides a means for securely communicating among remote computers 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.

NOTE: The term *tunnel* does not denote tunnel mode (see "Packet Processing in Tunnel Mode" on page 11). Instead, it refers to the IPsec connection.

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.

This topic includes the following sections:

IPsec VPN Topologies

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*.

Comparison of Policy-Based VPNs and Route-Based VPNs

NOTE: **Policy-based VPNs are only supported on SRX5400, SRX5600, and SRX5800 devices. Platform support depends on the Junos OS release in your installation.**

Table 3 on page 4 summarizes the differences between policy-based VPNs and route-based VPNs.

Table 3: 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.

Table 3: Comparison Between Policy-Based VPNs and Route-Based VPNs *(continued)*

Policy-Based VPNs	Route-Based VPNs
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.
With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy.	When the security device does a route lookup to find the interface through which it must send traffic to reach an address, it finds a route through a secure tunnel (st0) interface. With a route-based VPN tunnel, you can consider a tunnel as a means for delivering traffic, and can consider the policy as a method for either permitting or denying the delivery of that traffic.

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 11.)
- Key-management method, either manual key or AutoKey IKE. (See "IPsec Key Management" on page 6.)
- 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. (See "IPsec Security Protocols" on page 8.)
- Security parameter index (SPI) value.

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

IPsec Key Management

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 "IPsec Tunnel Negotiation" on page 9.

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.

  NOTE: **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 follows:

- DH Group 1—768-bit modulus
- DH Group 2—1024-bit modulus
- DH Group 5—1536-bit modulus
- DH Group 14—2048-bit modulus
- DH Group 19—256-bit modulus elliptic curve
- DH Group 20—384-bit modulus elliptic curve
- DH Group 24—2048-bit modulus with 256-bit prime order subgroup

NOTE: **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

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 "IPsec Tunnel Negotiation" on page 9.

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:

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-1 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-1)—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.

NOTE: **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.**

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 11.)

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 the MD5 or the SHA-1 algorithm.

NOTE: **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

To establish an AutoKey IKE IPsec tunnel, two phases of negotiation are required:

- In Phase 1, the participants establish a secure channel in which to negotiate the IPsec security associations (SAs).
- In Phase 2, the participants negotiate the IPsec SAs for encrypting and authenticating the ensuing exchanges of user data.

For a manual key IPsec tunnel, because all the SA parameters have been previously defined, there is no need to negotiate which SAs to use. In essence, the tunnel has already been established. When traffic matches a policy using that manual key tunnel or when a route involves the tunnel, the Juniper Networks device simply encrypts and authenticates the data, as you determined, and forwards it to the destination gateway.

The remote IKE gateway address can be in any virtual routing (VR) instance. VR is determined during IKE Phase 1 and Phase 2 negotiation. VR does not have to be configured in the IKE proposals. If the IKE gateway interface is moved from one VR to another, the existing IKE Phase 1 and Phase 2 negotiations for the IKE gateway are cleared, and new Phase 1 and Phase 2 negotiations are performed.

NOTE:

- On SRX Series devices, when you enable VPN, 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 virtual routers.
  - An st0 interface address cannot overlap in route-based VPN in point-to-multipoint tunnel such as NHTB.
  - An st0 interface address can overlap in route-based VPN in point-to-point tunnel.
- The combinations of local IP addresses and remote gateway IP addresses of IPsec VPN tunnels configured across VRs have to be unique.
- When the loopback interface is used as the IKE gateway external interface, the physical interface for IKE negotiation should be in the same VR.

Related Documentation

- Example: Configuring a Policy-Based VPN on page 206
- Example: Configuring a Route-Based VPN on page 50
- Understanding IKE and IPsec Packet Processing on page 10
- Understanding Phase 1 of IKE Tunnel Negotiation on page 18
- Understanding Phase 2 of IKE Tunnel Negotiation on page 20
- Understanding Hub-and-Spoke VPNs on page 69

Understanding IKE and IPsec Packet Processing

Supported Platforms SRX Series, vSRX

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 VPN Overview” on page 3.) 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.

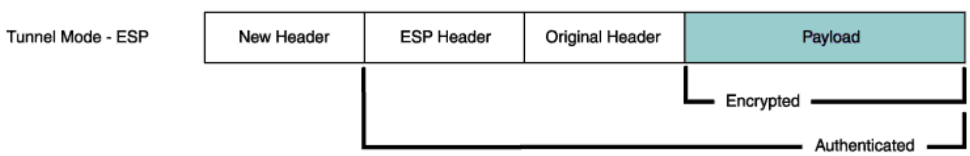
This topic includes the following sections:

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.

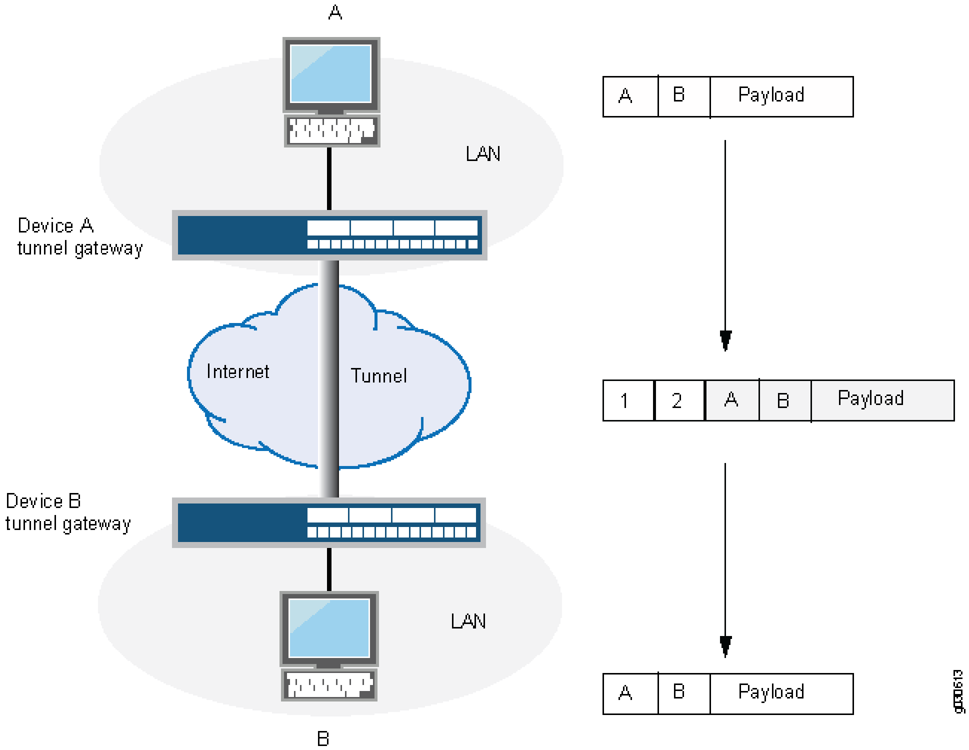
In tunnel mode, the entire original IP packet—payload and header—is encapsulated within another IP payload, and a new header is appended to it, as shown in Figure 1 on page 11. The entire original packet can be encrypted, authenticated, or both. With the Authentication Header (AH) protocol, the AH and new headers are also authenticated. With the Encapsulating Security Payload (ESP) protocol, the ESP header can also be authenticated.

Figure 1: 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 2 on page 12.

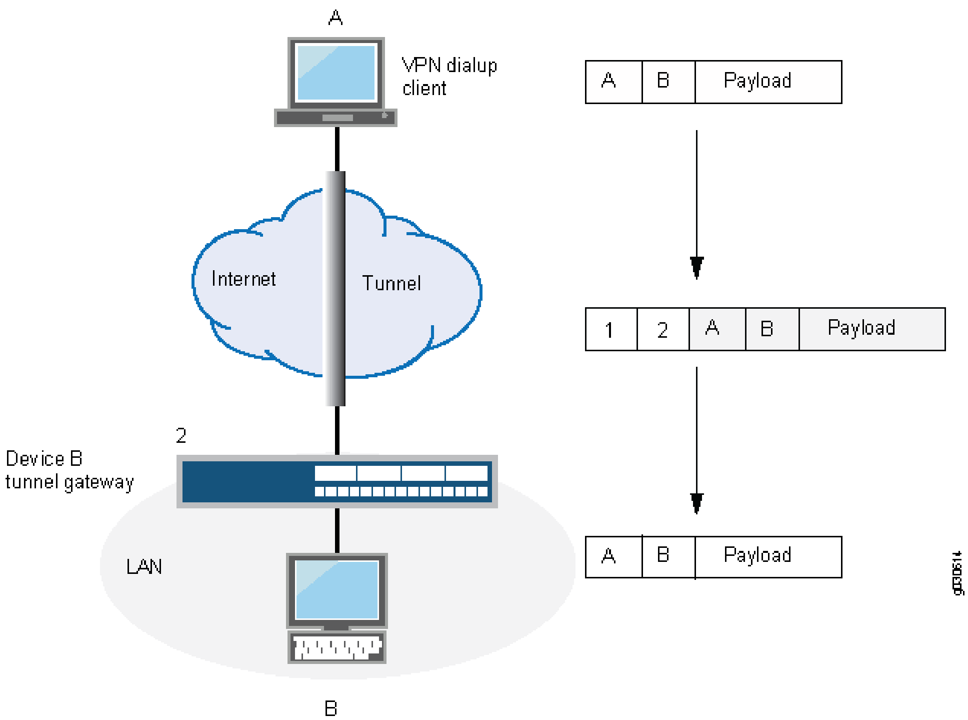
Figure 2: Site-to-Site VPN in Tunnel Mode



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 3 on page 13). 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.

NOTE: Some VPN clients, such as the dynamic VPN client and Netscreen-Remote, use a virtual inner IP address (also called a "sticky address"). Netscreen-Remote enables you to define the virtual IP address. The dynamic VPN client uses the virtual IP address assigned during the XAuth configuration exchange. 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 3: Dial-Up VPN in Tunnel Mode

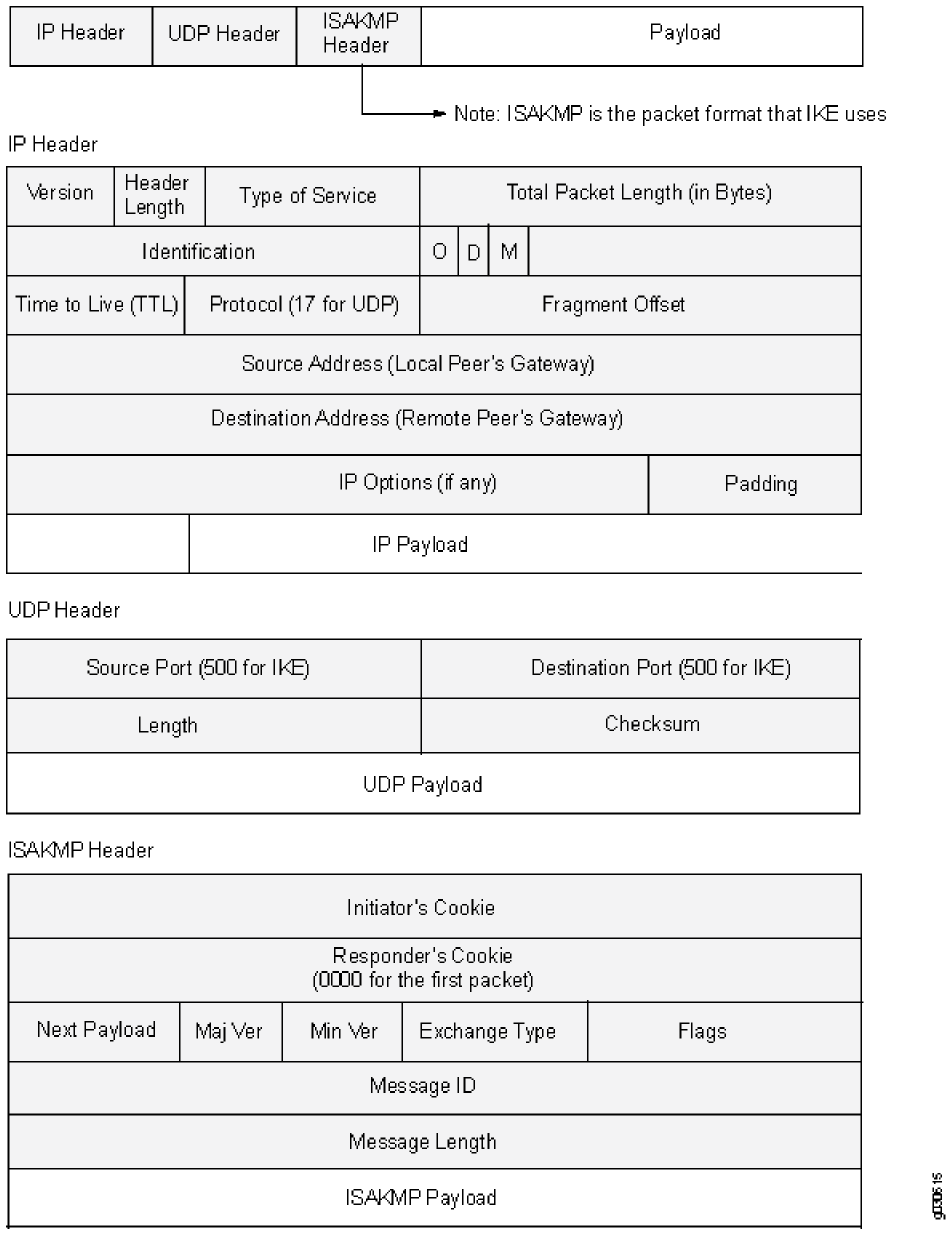


IKE Packet Processing

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 4 on page 14.

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 4: IKE Packet for Phases 1 and 2



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

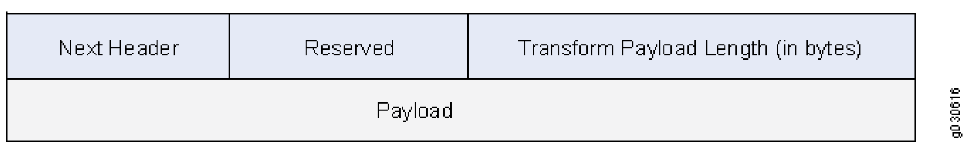
- 0002—SA Negotiation Payload contains a definition for a Phase 1 or Phase 2 SA.
- 0004—Proposal Payload can be a Phase 1 or Phase 2 proposal.
- 0008—Transform Payload gets encapsulated in a proposal payload that gets encapsulated in an SA payload.
- 0010—Key Exchange (KE) Payload contains information 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.

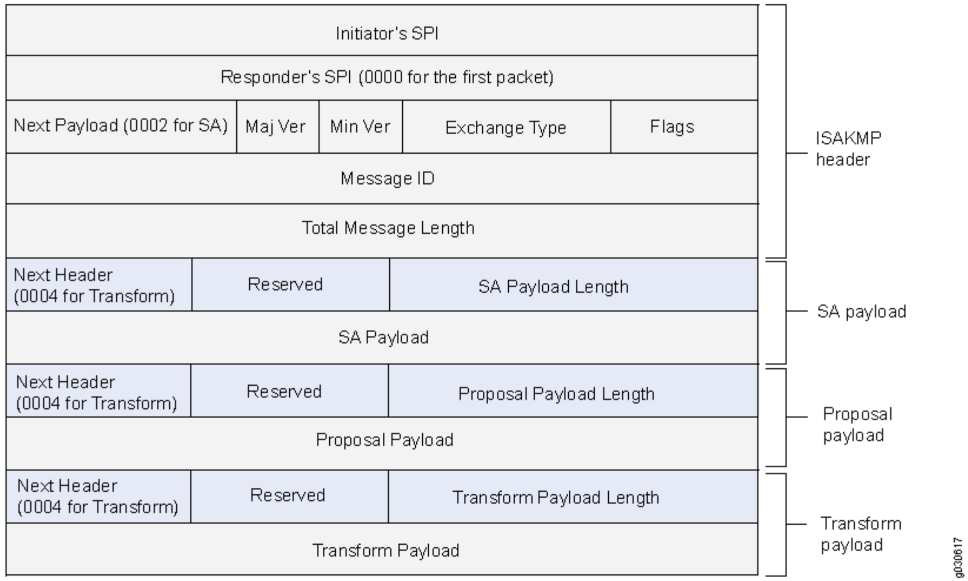
Each ISAKMP payload begins with the same generic header, as shown in Figure 5 on page 15.

Figure 5: 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 6 on page 16 for an example.

Figure 6: ISAKMP Header with Generic ISAKMP Payloads



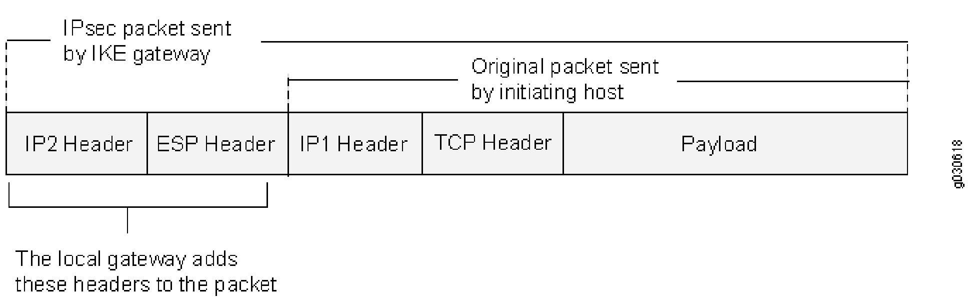
IPsec Packet Processing

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 7 on page 16. The device adds two additional headers to the original packet that the initiating host sends.

NOTE: For information about ESP, see "ESP Protocol" on page 9. For information about tunnel mode, see "Packet Processing in Tunnel Mode" on page 11.

As shown in Figure 7 on page 16, the packet that the initiating host constructs includes the payload, the TCP header, and the inner IP header (IP1).

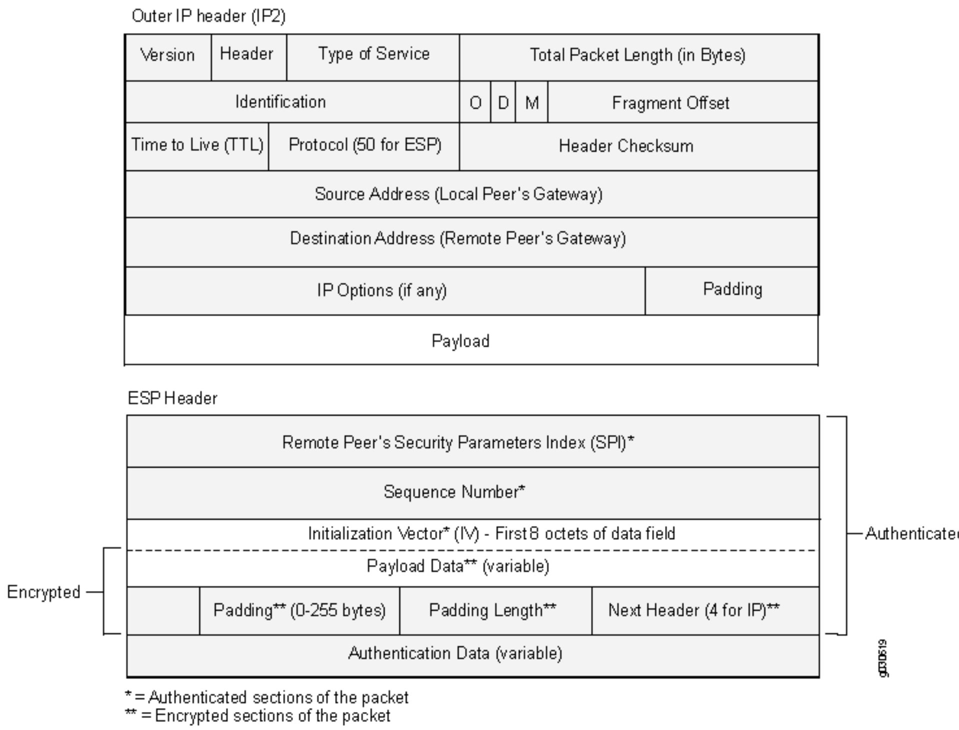
Figure 7: IPsec Packet—ESP in Tunnel Mode



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.

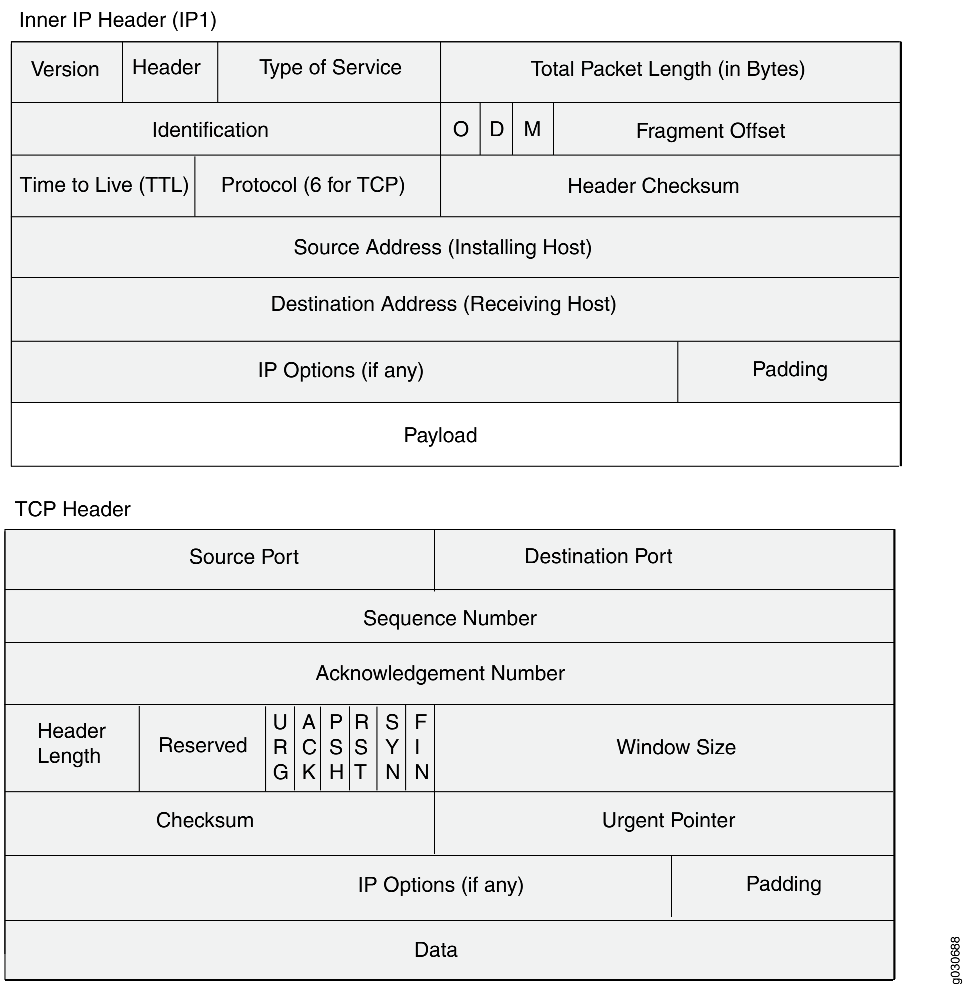
The ESP header contains information that allows the remote peer to properly process the packet when it receives it. This is shown in Figure 8 on page 17.

Figure 8: 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 9 on page 18.

Figure 9: Inner IP Header (IP1) and TCP Header



Related Documentation

- IPsec VPN Overview on page 3
- Understanding Phase 1 of IKE Tunnel Negotiation on page 18
- Understanding Phase 2 of IKE Tunnel Negotiation on page 20
- Understanding Hub-and-Spoke VPNs on page 69
- Example: Configuring a Policy-Based VPN on page 206
- Example: Configuring a Route-Based VPN on page 50

Understanding Phase 1 of IKE Tunnel Negotiation

Supported Platforms SRX Series, vSRX

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 Security Protocols” on page 8.)
- Authentication algorithms—Message Digest 5 (MD5) and Secure Hash Algorithm (SHA-1). (See “IPsec Security Protocols” on page 8.)
- Diffie-Hellman (DH) group. (See “Diffie-Hellman Exchange” on page 7.)
- Preshared key or RSA/DSA certificates. (See “IPsec Key Management” on page 6.)

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 the following predefined Phase 1 proposals:

- Standard—pre-g2-aes128-sha and pre-g2-3des-sha
- Compatible—pre-g2-3des-sha, pre-g2-3des-md5, pre-g2-des-sha, and pre-g2-des-md5
- Basic—pre-g1-des-sha and pre-g1-des-md5

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.

  NOTE: **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.

Related Documentation

- IPsec VPN Overview on page 3
- Understanding Phase 2 of IKE Tunnel Negotiation on page 20
- Example: Configuring a Policy-Based VPN on page 206
- Example: Configuring a Route-Based VPN on page 50

Understanding Phase 2 of IKE Tunnel Negotiation

Supported Platforms SRX Series, vSRX

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.

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 the following predefined Phase 2 proposals:

- Standard—g2-esp-3des-sha and g2-esp-aes128-sha
- Compatible—nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, and nopfs-esp-des-md5
- Basic—nopfs-esp-des-sha and nopfs-esp-des-md5

You can also define custom Phase 2 proposals.

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.

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Policy-Based VPN on page 206
- Example: Configuring a Route-Based VPN on page 50

IPsec VPN with Autokey IKE Configuration Overview

Supported Platforms SRX Series, vSRX

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 "Example: Configuring Global SPI and VPN Monitoring Features" on page 905.

Related Documentation

- Understanding Route-Based IPsec VPNs on page 47
- Understanding Policy-Based IPsec VPNs on page 205
- Configuring IPsec VPN Using the VPN Wizard on page 28

IPsec VPN with Manual Keys Configuration Overview

Supported Platforms SRX Series, vSRX

This overview describes the basic steps to configure a route-based or policy-based IPsec VPN using manual keys.

To configure a route-based or policy-based IPsec VPN using manual keys:

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

   (For route-based VPNs) Configure routing. Configure a secure tunnel st0.x interface.

2. Configure an IPsec VPN tunnel by specifying the following parameters:

   - Authentication algorithm and key
   - Encryption algorithm and key
   - Outgoing interface
   - IP address of the peer
   - IPsec protocol for the security association
   - Security parameter index

   (For route-based VPNs) Bind the secure tunnel interface st0.x to the IPsec VPN tunnel.

3. Configure 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.

Related Documentation

- Understanding Route-Based IPsec VPNs on page 47
- Understanding Policy-Based IPsec VPNs on page 205
- Example: Configuring an IPv6 IPsec Manual VPN on page 329

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

Supported Platforms SRX Series, vSRX

Table 4 on page 24 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 4: Recommended Configuration for Site-to-Site VPN with Static IP Addresses

Configuration Option	Comment
IKE configuration options:	
Autokey IKE with certificates	Manual key is not recommended.
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.
Anti-replay protection	Enabled by default. Disabling this feature might resolve compatibility issues with third-party peers.

Related Documentation

- IPsec VPN Overview on page 3

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

Supported Platforms SRX Series, vSRX

Table 5 on page 25 lists the configuration options for a generic site-to-site or dialup VPN, where the peer devices have dynamic IP addresses.

Table 5: Recommended Configuration for Site-to-Site or Dialup VPNs with Dynamic IP Addresses

Configuration Option	Comment
IKE configuration options:	
Autokey IKE with certificates	Manual key is not recommended.
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.
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.

Related Documentation

- IPsec VPN Overview on page 3

Understanding IPsec VPNs with Dynamic Endpoints

Supported Platforms SRX Series, vSRX

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 devices, IKEv1 or IKEv2 is supported with dynamic endpoint VPNs. Dynamic endpoint VPNs on SRX Series devices support IPv4 traffic on secure tunnels. Starting with Junos OS Release 15.1X49-D80, dynamic endpoint VPNs on SRX Series devices support IPv6 traffic on secure tunnels.

NOTE: **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 device 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. IKEv2 does not use aggressive mode, so you can configure either main or aggressive mode when using IKEv2 with dynamic endpoint VPNs.

IKE Policies and External Interfaces

Starting with Junos OS Releases 12.3X48-D40 and 15.1X49-D70, all dynamic endpoint gateways configured on SRX Series devices 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 device. 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 devices 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.

Release History Table

Release	Description
15.1X49-D80	Starting with Junos OS Release 15.1X49-D80, dynamic endpoint VPNs on SRX Series devices support IPv6 traffic on secure tunnels.
12.3X48-D40	Starting with Junos OS Releases 12.3X48-D40 and 15.1X49-D70, all dynamic endpoint gateways configured on SRX Series devices that use the same external interface can use different IKE policies, but the IKE policies must use the same IKE proposal.

Related Documentation

- Example: Configuring NAT-T with Dynamic Endpoint VPN on page 286

Configuring Remote IKE IDs for Site-to-Site VPNs

Supported Platforms SRX Series, vSRX

By default, SRX Series devices 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 device. This can lead to a Phase 1 validation failure.

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

- On the SRX Series device, 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.

  NOTE: **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 device. If the peer device is an SRX Series device, 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.

Related Documentation

- Understanding NAT-T on page 227
- Example: Configuring a Route-Based VPN with Only the Responder Behind a NAT Device on page 228
- Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device on page 257

Configuring IPsec VPN Using the VPN Wizard

Supported Platforms SRX Series

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>Tasks>Configure 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.

Related Documentation

- IPsec VPN Overview on page 3
- Understanding Phase 1 of IKE Tunnel Negotiation on page 18
- Understanding Phase 2 of IKE Tunnel Negotiation on page 20

Understanding Suite B and PRIME Cryptographic Suites

Supported Platforms SRX Series

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.

NOTE: Suite B and PRIME are not fully supported on SRX1500, SRX3400, and SRX3600 devices, and on SRX5400, SRX5600, and SRX5800 devices that do not have the SPC2 (SRX5K-SPC-4-14-320). (Platform support depends on the Junos OS release in your installation.) You can configure IKE with Suite B options on these devices, but AES-GCM options are not supported. If you configure IKE with Suite B options on these devices, VPN establishment is slower because the devices do not have the hardware processors that can accelerate Suite B algorithm processing.

NOTE: Suite B and PRIME are not supported with the Group VPNv2 feature.

CLI options support Suite B and PRIME compliance in IKE and IPsec proposal configuration:

- For IKE proposals configured at the [**edit security ike proposal** ***proposal-name***] hierarchy level:
  - **authentication-algorithm** options include **sha-256** and **sha-384**.
  - **authentication-method** options include **ecdsa-signatures-256** and **ecdsa-signatures-384**.
  - **dh-group** options include **group19** and **group20**.
  - **encryption-algorithm** options for PRIME include **aes-128-gcm** and **aes-256-gcm**.
- For IPsec proposals configured at the [**edit security ipsec proposal** ***proposal-name***] hierarchy level, **encryption-algorithm** options include **aes-128-gcm**, **aes-192-gcm**, and **aes-256-gcm**.
- For IPsec policies configured at the [**edit security ipsec policy** ***policy-name***] hierarchy level, the **perfect-forward-secrecy keys** options include **group19** and **group20**.
- For convenience, predefined proposals that provide compliance with Suite B (**suiteb-gcm-128** and **suiteb-gcm-256**) and PRIME (**prime-128** and **prime-256**) are available at the [**edit security ike policy** ***policy-name***] and [**edit security ipsec policy** ***policy-name***] hierarchy levels.

NOTE: VPN monitoring and cryptographic configuration options **ecdsa-signatures-384** (for IKE authentication) and DH group 20 consume considerable CPU resources. If VPN monitoring and the **ecdsa-signatures-384** and **group20** options are used on an SRX Series device with a large number of tunnels configured, the SRX Series device must have the SPC2 installed.

Related Documentation

- IPsec VPN Overview on page 3

CHAPTER 2

Understanding VPN Tunnel Management

Understanding Distributed VPNs in SRX Series Services Gateways

Supported Platforms SRX Series

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 6 on page 34 shows an example of load balancing on an SRX5000 line device with three SPUs running eight IPsec tunnels over four IKE gateways. Note that SPU 0, SPU 1, or SPU 2 could be selected for IKE gateway 4 because all three SPUs have an equal load of one gateway each.

Table 6: Load Balancing 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
SPU0, SPU1, or SPU2	IKE-4	IPsec-8

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**.

Understanding VPN Support for Inserting Services Processing Cards

Supported Platforms SRX Series

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.

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. For a new site-to-site tunnel, the SPU with the smallest load is chosen as the anchor SPU. (The load corresponds to the number of site-to-site gateways or manual VPN tunnels anchored on an SPU.) If multiple SPUs have the same smallest load, then any of the SPUs may be chosen as the anchor SPU. A newly configured site-to-site tunnel is guaranteed to be anchored on a new SPU only if the loads of the previously installed SPUs are all greater than 0.

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.

You can view the tunnel mapping on different SPUs using the **show security ike tunnel-map** command.

Related Documentation

- **show security ike tunnel-map on page 1226**
- Understanding Distributed VPNs in SRX Series Services Gateways on page 33

CHAPTER 3

Configuring IPsec SA for OSPF

Understanding OSPF and OSPFv3 Authentication on SRX Series Devices

Supported Platforms SRX Series, vSRX

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 devices:

- 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.

Related Documentation

- Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Device on page 39

Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Device

Supported Platforms SRX Series, vSRX

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 7 on page 39 lists the parameters and values configured for the manual SA in this example.

Table 7: Manual SA for IPsec OSPF Interface Authentication

Parameter	Value
SA name	sa1
Mode	transport
Direction	bidirectional

Table 7: Manual SA for IPsec OSPF Interface Authentication *(continued)*

Parameter	Value
Protocol	AH
SPI	256
Authentication algorithm Key	hmac-md5-96 (ASCII) 123456789012abc
Encryption algorithm Key	des (ASCII) cba210987654321

Configuration

- Configuring a Manual SA on page 40
- Enabling IPsec Authentication for an OSPF Interface on page 42

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.

NOTE: 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.

   NOTE: 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

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.

Related Documentation

- Understanding OSPF and OSPFv3 Authentication on SRX Series Devices on page 37

PART 2

Configuring Route-Based IPsec VPNs

CHAPTER 4

Configuring Route-Based VPNs

Understanding Route-Based IPsec VPNs

Supported Platforms SRX Series, vSRX

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.

NOTE: 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.

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.

NOTE: Configuring RIP demand circuits over 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.

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Hub-and-Spoke VPN on page 70
- Example: Configuring a Policy-Based VPN on page 206

Understanding CoS Support on st0 Interfaces

Supported Platforms SRX Series, vSRX

Starting with Junos OS 15.1X49-D60, 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 devices and vSRX2.0:

- Classifiers
- Policers
- Queuing, scheduling, and shaping
- Rewrite markers
- Virtual channels

NOTE: **Starting with Junos OS 15.1X49-D70, 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.**

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 8 on page 48 describes the st0 CoS feature support for different types of VPNs.

Table 8: CoS Feature Support for VPN

Classifier Features	Site-to-Site VPN (P2P)	AutoVPN (P2P)	Site-to-Site/Auto VPN /AD-VPN (P2MP)

Table 8: CoS Feature Support for VPN *(continued)*

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.

  NOTE: **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.

Release History Table

Release	Description
15.1X49-D70	Starting with Junos OS 15.1X49-D70, 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 15.1X49-D60, 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 Feature Guide for Security Devices*

Example: Configuring a Route-Based VPN

Supported Platforms SRX Series, vSRX

This example shows how to configure a route-based IPsec VPN to allow data to be securely transferred between a branch office and the corporate office.

Requirements

This example uses the following hardware:

- SRX240 device
- SSG140 device

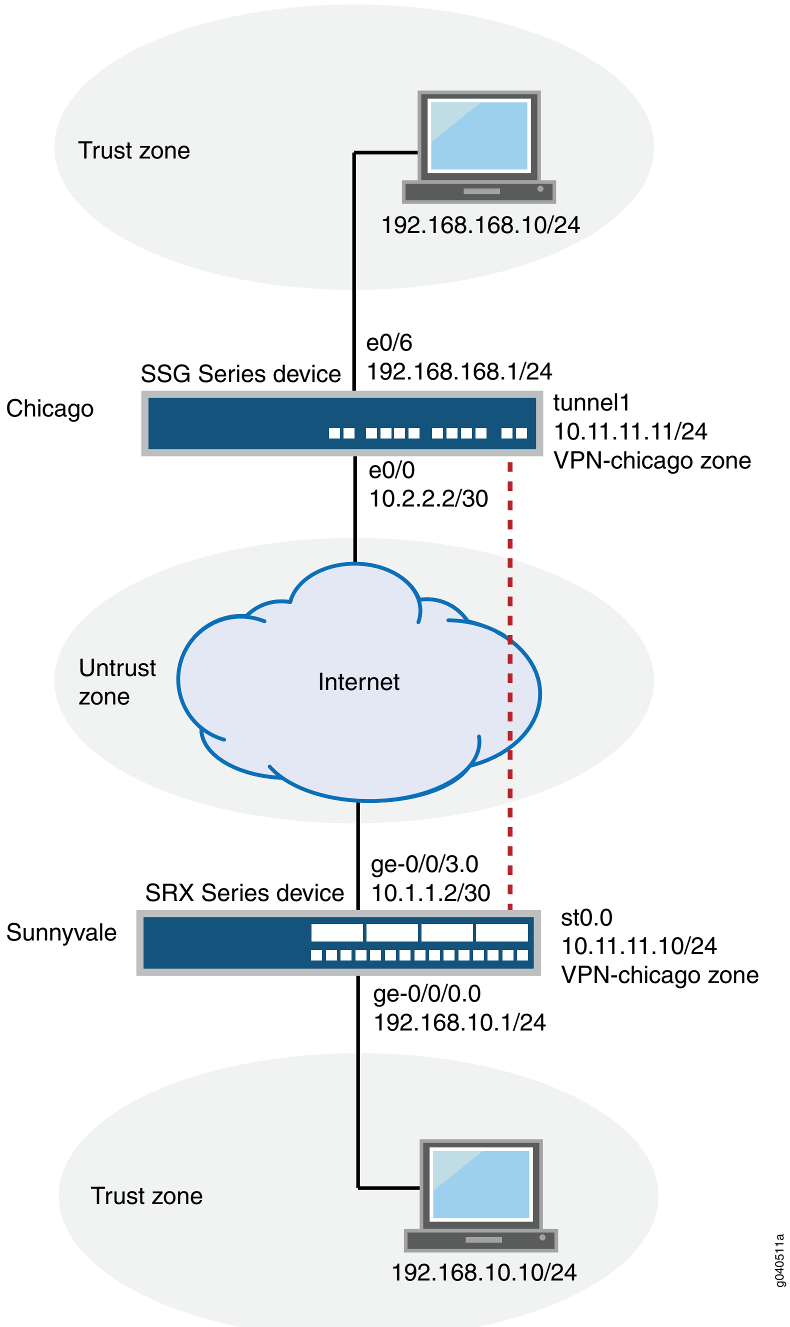
Before you begin, read “IPsec VPN Overview” on page 3.

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.

Figure 10 on page 51 shows an example of a route-based VPN topology. In this topology, the SRX Series device is located in Sunnyvale, and an SSG Series device (or a third-party device) is located in Chicago.

Figure 10: Route-Based VPN Topology



In this example, you configure interfaces, an IPv4 default route, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, security policy, and TCP-MSS parameters. See Table 9 on page 52 through Table 13 on page 53 for specific configuration parameters used in this example.

Table 9: 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 10.1.1.2.
	192.168.168.0/24	The next hop is st0.0.
Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.
	untrust	• IKE is the only allowed system service. • The ge-0/0/3.0 interface is bound to this zone.
	vpn-chicago	The st0.0 interface is bound to this zone.
Address book entries	sunnyvale	• This address is an entry in the address book **book1**, which is attached to a zone called **trust**. • The address for this address book entry is 192.168.10.0/24.
	chicago	• This address is an entry in the address book **book2**, which is attached to a zone called **vpn-chicago**. • The address for this address book entry is 192.168.168.0/24.

Table 10: IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw-chicago	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2

Table 11: IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2
VPN	ike-vpn-chicago	• IKE gateway reference: gw-chicago • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0

Table 12: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

The security policy permits traffic from the trust zone to the vpn-chicago zone.	vpn-tr-chi	• Match criteria:
		  ◦ source-address sunnyvale
		  ◦ destination-address chicago
		  ◦ application any
		• Action: permit

The security policy permits traffic from the vpn-chicago zone to the trust zone.	vpn-chi-tr	• Match criteria:
		  ◦ source-address chicago
		  ◦ destination-address sunnyvale
		  ◦ application any
		• Action: permit

Table 13: 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. NOTE: 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

Configuring Interface, Static Route, Security Zone, and Address Book Information

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 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 10.1.1.2
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 vpn-chicago interfaces st0.0
set security address-book book1 address sunnyvale 192.168.10.0/24
set security address-book book1 attach zone trust
set security address-book book2 address chicago 192.168.168.0/24
set security address-book book2 attach zone vpn-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 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 10.1.1.2
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 an address book and attach a zone to it.

   [edit security address-book book1]
   user@host# **set address sunnyvale 192.168.10.0/24**
   user@host# **set attach zone trust**

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 another address book and attach a zone to it.

    [edit security address-book book2]
    user@host# **set address chicago 192.168.168.0/24**
    user@host# **set attach zone vpn-chicago**

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-0/0/0 {
   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 10.1.1.2;
   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 {
   host-inbound-traffic {
      system-services {
         all;
      }
   }
   interfaces {
      ge-0/0/0.0;
```



```
    }
  }
  security-zone vpn-chicago {
    interfaces {
      st0.0;
    }
  }
[edit]
user@host# show security address-book
book1 {
  address sunnyvale 192.168.10.0/24;
  attach {
    zone trust;
  }
}
  book2 {
    address chicago 192.168.168.0/24;
    attach {
      zone vpn-chicago;
    }
  }
```

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

Configuring IKE

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 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-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
```

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. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike-phase1-policy]
user@host# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@host# set proposals ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike-phase1-policy]
user@host# set pre-shared-key ascii-text $ABC123
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@host# set gateway gw-chicago external-interface ge-0/0/3.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gw-chicago]
user@host# set ike-policy ike-phase1-policy
```

12. Define the IKE Phase 1 gateway address.

```
[edit security ike gateway gw-chicago]
user@host# set 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@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 {
  mode main;
  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;
}
```

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

Configuring IPsec

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 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 ike-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ike-vpn-chicago ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn ike-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 ike-vpn-chicago ike gateway gw-chicago
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn ike-vpn-chicago ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn ike-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 ike-vpn-chicago {
    bind-interface st0.0;
    ike {
      gateway gw-chicago;
      ipsec-policy ipsec-phase2-policy;
    }
  }
```

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

Configuring Security Policies

CLI Quick Configuration

To quickly configure this 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 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 chicago
user@host# set policy vpn-chi-tr match destination-address sunnyvale
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 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 flow tcp-mss ipsec-vpn mss 1350
```


Step-by-Step Procedure

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

To configure TCP-MSS information:

1. Configure TCP-MSS information.

   ```
   [edit]
   user@host# set security flow tcp-mss ipsec-vpn mss 1350
   ```

Results

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

```
[edit]
user@host# show security flow
tcp-mss {
  ipsec-vpn {
    mss 1350;
  }
}
```

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

Configuring the SSG Series Device

CLI Quick Configuration

For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts and Examples ScreenOS Reference Guide*, which is located at http://www.juniper.net/techpubs.

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 Main outgoing-interface ethernet0/0 preshare
  $ABC123 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

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status on page 64
- Verifying the IPsec Phase 2 Status on page 66
- Reviewing Statistics and Errors for an IPsec Security Association on page 67
- Testing Traffic Flow Across the VPN on page 68

Verifying the IKE Phase 1 Status

Purpose Verify the IKE Phase 1 status.

Action

NOTE: 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 device 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  Main
```

```
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: Main, 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
Phase 2 negotiations in progress: 0
```

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 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)
- Role information

NOTE: **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 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)
    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, 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 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, 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 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 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@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 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 device 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.

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Hub-and-Spoke VPN on page 70
- Example: Configuring a Policy-Based VPN on page 206

CHAPTER 5

Configuring Hub-and-Spoke VPNs

Understanding Hub-and-Spoke VPNs

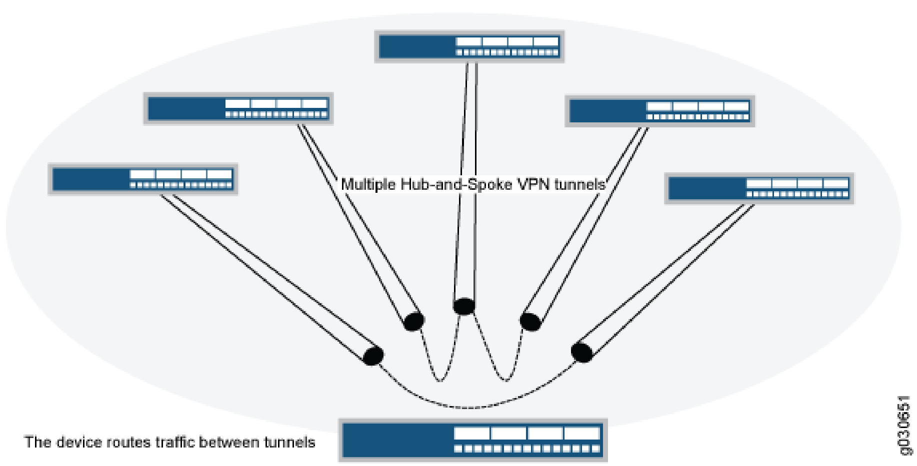
Supported Platforms SRX Series, vSRX

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 11 on page 69.)

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

NOTE: **SRX Series devices support only the route-based hub-and-spoke feature.**

Figure 11: Multiple Tunnels in a Hub-and-Spoke VPN Configuration



Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Hub-and-Spoke VPN on page 70

Example: Configuring a Hub-and-Spoke VPN

Supported Platforms SRX Series, vSRX

This example shows how to configure a hub-and-spoke IPsec VPN for an enterprise-class deployment.

Requirements

This example uses the following hardware:

- SRX240 device
- SRX5800 device
- SSG140 device

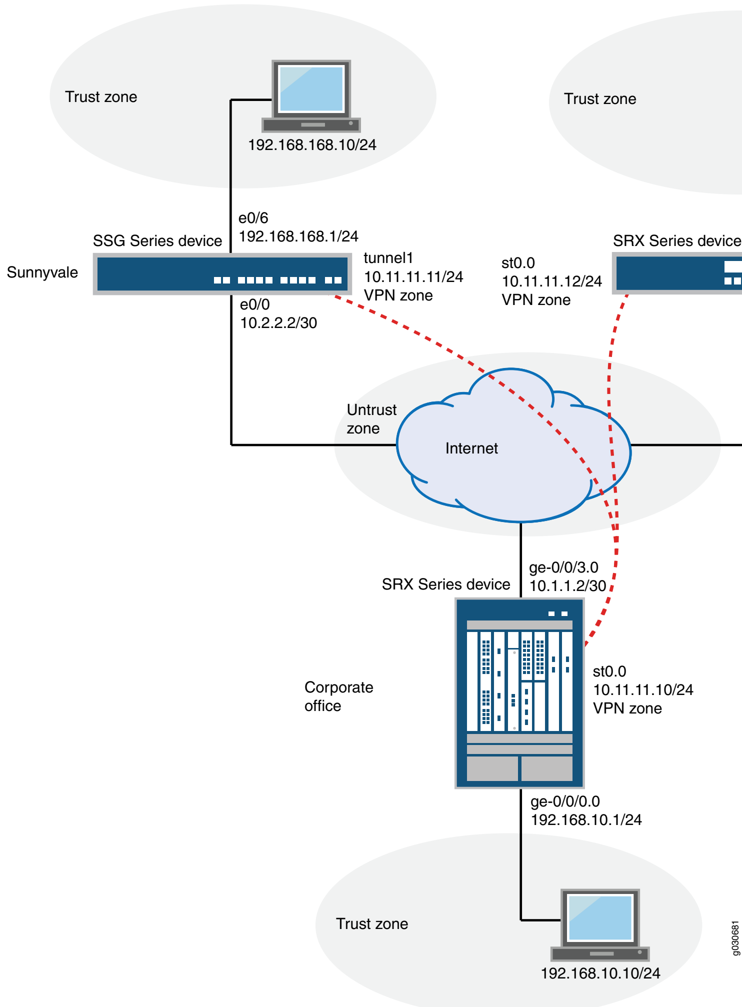
Before you begin, read “IPsec VPN Overview” on page 3.

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 12 on page 71 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 device is located at the Westford branch, and an SSG140 device is located at the Sunnyvale branch.

Figure 12: 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 14 on page 71 through Table 18 on page 76 for specific configuration parameters used in this example.

Table 14: 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

Table 14: Interface, Security Zone, and Address Book Information *(continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
		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. • The ge-0/0/0.0 interface is bound to this zone.
		untrust	• IKE is the only allowed system service. • The ge-0/0/3.0 interface is bound to this zone.
		vpn	The st0.0 interface is bound to this zone.
Spoke	Security zones	trust	• All system services are allowed. • The ge-0/0/3.0 interface is bound to this zone.
		untrust	• IKE is the only allowed system service. • 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. • 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. • The address for this address book entry is 192.168.168.0/24.
		westford-net	• This address is for the vpn zone's address book. • The address for this address book entry is 192.168.178.0/24.

Table 14: Interface, Security Zone, and Address Book Information *(continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
Spoke	Address book entries	local-net	• This address is for the trust zone's address book. • 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. • 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. • The address for this address book entry is 192.168.168.0/24.

Table 15: IKE Phase 1 Configuration Parameters

Hub or Spoke	Feature	Name	Configuration Parameters
Hub	Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
	Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
	Gateway	gw-westford	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.3.3.2
		gw-sunnyvale	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2

Table 15: IKE Phase 1 Configuration Parameters *(continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
Spoke	Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
	Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
	Gateway	gw-corporate	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/0.0 • Gateway address: 10.1.1.2

Table 16: IPsec Phase 2 Configuration Parameters

Hub or Spoke	Feature	Name	Configuration Parameters
Hub	Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
	Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2
	VPN	vpn-sunnyvale	• IKE gateway reference: gw-sunnyvale • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0
		vpn-westford	• IKE gateway reference: gw-westford • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0
Spoke	Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
	Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2

Table 16: IPsec Phase 2 Configuration Parameters *(continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
	VPN	vpn-corporate	• IKE gateway reference: gw-corporate • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0

Table 17: 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:
			  ◦ source-address local-net
			  ◦ destination-address sunnyvale-net
			  ◦ destination-address westford-net
			  ◦ application any

	The security policy permits traffic from the vpn zone to the trust zone.	spokes-to-local	Match criteria: • source-address sunnyvale-net • source-address westford-net • destination-address local-net • application any
	The security policy permits intrazone traffic.	spoke-to-spoke	Match criteria: • source-address any • destination-address any • application any

Spoke	The security policy permits traffic from the trust zone to the vpn zone.	to-corp	• Match criteria:
			  ◦ source-address local-net
			  ◦ destination-address corp-net
			  ◦ destination-address sunnyvale-net
			  ◦ application any

	The security policy permits traffic from the vpn zone to the trust zone.	from-corp	Match criteria: • source-address corp-net • source-address sunnyvale-net • destination-address local-net • application any

Table 17: 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: • source-address any • source-destination any • application any • Permit action: source-nat interface By specifying **source-nat interface**, the SRX Series device 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 18: 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. NOTE: 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

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
```

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 http://www.juniper.net/techpubs.

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

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

NOTE: 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 device 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

NOTE: **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 device.
- Auto— NHTB was not configured, but the entry was automatically populated into the NHTB table during Phase 2 negotiations between two SRX Series devices

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 device 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.

Related Documentation

- Understanding Hub-and-Spoke VPNs on page 69
- Example: Configuring a Route-Based VPN on page 50
- Example: Configuring a Policy-Based VPN on page 206

CHAPTER 6

Configuring VPNs for IKEv2

Understanding Internet Key Exchange Version 2

Supported Platforms SRX Series, vSRX

Internet Key Exchange version 2 (IKEv2) is the next generation standard for secure key exchange between peer VPN devices, as defined in RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*.

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.

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, acknowledgements, 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.

IKEv2 includes support for:

- Route-based VPNs.

NOTE: IKEv2 does not support policy-based VPNs.

- Site-to-site VPNs.
- Dead peer detection.
- Chassis cluster.
- 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. If a child SA is required, it is rekeyed. However, if child SAs are currently active, the corresponding IKE SA is rekeyed.

NOTE: On SRX Series devices, if an IPsec VPN tunnel is established using IKEv2, a small number of packet drops might be observed during CHILD_SA rekey as a result of "bad SPI" being logged. This occurs only when the SRX Series device is the responder for this rekey and the peer is a non-Juniper Networks device, and the latency between the peers is low and the packet rate is high. To avoid this issue, ensure that the SRX Series device always initiates the rekeys by setting its IPsec lifetime to a lower value than that of the peer.

- AutoVPN.
- Dynamic endpoint VPN.

IKEv2 does not support the following features:

- Policy-based VPN.
- Dialup tunnels.
- VPN monitoring.
- EAP.
- Multiple child SAs for the same traffic selectors for each QoS value.
- IP Payload Compression Protocol (IPComp).
- Traffic selectors.

Related Documentation

- Understanding IKEv2 Configuration Payload on page 107
- Example: Configuring a Route-Based VPN for IKEv2 on page 108

Understanding IKEv2 Configuration Payload

Supported Platforms SRX Series, vSRX

Configuration payload is an Internet Key Exchange version 2 (IKEv2) feature used to propagate provisioning information from a responder (or server) to an initiator (or client). 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 19 on page 107 describes the IKEv2 configuration attributes supported on SRX Series devices.

Table 19: 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
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_IP4_DHCP	6	Instructs the host to send any internal DHCP request to the address contained within the attribute. Multiple DHCP servers can be requested. The responder can respond with zero or more DHCP server attributes.	0 or 4 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.

In a route-based VPN, secure tunnel (st0) interfaces operate in either point-to-multipoint or point-to-point mode. Dynamic address assignment through the IKEv2 configuration payload is supported for point-to-multipoint interfaces only. 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.

Related Documentation

- Understanding Internet Key Exchange Version 2 on page 105
- Understanding Pico Cell Provisioning on page 125
- Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload on page 128
- Example: Configuring NAT-T with Dynamic Endpoint VPN on page 286

Example: Configuring a Route-Based VPN for IKEv2

Supported Platforms SRX Series, vSRX

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 on page 108
- Overview on page 108
- Configuration on page 111
- Verification on page 121

Requirements

This example uses the following hardware:

- SRX240 device
- SSG140 device

Before you begin, read “IPsec VPN Overview” on page 3.

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 20 on page 109 through Table 24 on page 110 for specific configuration parameters used in this example.

Table 20: Interface, Static Route, Security Zone, and Address Book Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	192.168.10.1/24
	ge-0/0/3.0	10.1.1.2/30
	st0.0 (tunnel interface)	10.11.11.10/24
Static routes	0.0.0.0/0 (default route)	The next hop is 10.1.1.1.
	192.168.168.0/24	The next hop is st0.0.
Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.
	untrust	• IKE is the only allowed system service. • The ge-0/0/3.0 interface is bound to this zone.
	vpn-chicago	The st0.0 interface is bound to this zone.
Address book entries	sunnyvale	• This address is for the trust zone's address book. • The address for this address book entry is 192.168.10.0/24.
	chicago	• This address is for the untrust zone's address book. • The address for this address book entry is 192.168.168.0/24.

Table 21: IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw-chicago	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2

Table 22: IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2
VPN	ipsec-vpn-chicago	• IKE gateway reference: gw-chicago • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0

Table 23: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

The security policy permits traffic from the trust zone to the vpn-chicago zone.	vpn-tr-chi	• Match criteria:
		  • source-address sunnyvale
		  • destination-address chicago
		  • application any
		• Action: permit

The security policy permits traffic from the vpn-chicago zone to the trust zone.	vpn-chi-tr	• Match criteria:
		  • source-address chicago
		  • destination-address sunnyvale
		  • application any
		• Action: permit

Table 24: 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. NOTE: 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

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.

Configuring IKE

CLI Quick Configuration

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

```
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text "$ABC123"
set security ike gateway gw-chicago external-interface ge-0/0/3.0
set security ike gateway gw-chicago ike-policy ike-phase1-policy
set security ike gateway gw-chicago address 10.2.2.2
set security ike gateway gw-chicago version v2-only
```

Step-by-Step Procedure

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

To configure IKE:

1. Create the IKE Phase 1 proposal.

   ```
   [edit security ike]
   user@host# set proposal ike-phase1-proposal
   ```

2. Define the IKE proposal authentication method.

   ```
   [edit security ike proposal ike-phase1-proposal]
   user@host# set authentication-method pre-shared-keys
   ```

3. Define the IKE proposal Diffie-Hellman group.

   ```
   [edit security ike proposal ike-phase1-proposal]
   user@host# set dh-group group2
   ```

4. Define the IKE proposal authentication algorithm.

   [edit security ike proposal ike-phase1-proposal]
   user@host# **set authentication-algorithm sha1**

5. Define the IKE proposal encryption algorithm.

   [edit security ike proposal ike-phase1-proposal]
   user@host# **set encryption-algorithm aes-128-cbc**

6. Create an IKE Phase 1 policy.

   [edit security ike]
   user@host# **set policy ike-phase1-policy**

7. Specify a reference to the IKE proposal.

   [edit security ike policy ike-phase1-policy]
   user@host# **set proposals ike-phase1-proposal**

8. Define the IKE Phase 1 policy authentication method.

   [edit security ike policy ike-phase1-policy]
   user@host# **set pre-shared-key ascii-text "$ABC123"**

9. Create an IKE Phase 1 gateway and define its external interface.

   [edit security ike]
   user@host# **set gateway gw-chicago external-interface ge-0/0/3.0**

10. Define the IKE Phase 1 policy reference.

    [edit security ike gateway gw-chicago]
    user@host# **set ike-policy ike-phase1-policy**

11. Define the IKE Phase 1 gateway address.

    [edit security ike gateway gw-chicago]
    user@host# **set address 10.2.2.2**

12. Define the IKE Phase 1 gateway version.

    [edit security ike gateway gw-chicago]
    user@host# **set version v2-only**

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

```
[edit]
user@host# show security ike
proposal ike-phase1-proposal {
```

```
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
  }
  policy ike-phase1-policy {
    proposals ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
  }
  gateway gw-chicago {
    ike-policy ike-phase1-policy;
    address 10.2.2.2;
    external-interface ge-0/0/3.0;
    version v2-only;
  }
```

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

Configuring IPsec

CLI Quick Configuration

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

```
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ipsec-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ipsec-vpn-chicago ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn ipsec-vpn-chicago bind-interface st0.0
```

Step-by-Step Procedure

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

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

   ```
   [edit]
   user@host# set security ipsec proposal ipsec-phase2-proposal
   ```

2. Specify the IPsec Phase 2 proposal protocol.

   ```
   [edit security ipsec proposal ipsec-phase2-proposal]
   user@host# set protocol esp
   ```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

   ```
   [edit security ipsec proposal ipsec-phase2-proposal]
   user@host# set authentication-algorithm hmac-sha1-96
   ```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

   ```
   [edit security ipsec proposal ipsec-phase2-proposal]
   user@host# set encryption-algorithm aes-128-cbc
   ```

5. Create the IPsec Phase 2 policy.

   ```
   [edit security ipsec]
   user@host# set policy ipsec-phase2-policy
   ```

6. Specify the IPsec Phase 2 proposal reference.

   ```
   [edit security ipsec policy ipsec-phase2-policy]
   user@host# set proposals ipsec-phase2-proposal
   ```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

   ```
   [edit security ipsec policy ipsec-phase2-policy]
   user@host# set perfect-forward-secrecy keys group2
   ```

8. Specify the IKE gateway.

   ```
   [edit security ipsec]
   user@host# set vpn ipsec-vpn-chicago ike gateway gw-chicago
   ```

9. Specify the IPsec Phase 2 policy.

   ```
   [edit security ipsec]
   user@host# set vpn ipsec-vpn-chicago ike ipsec-policy ipsec-phase2-policy
   ```

10. Specify the interface to bind.

   ```
   [edit security ipsec]
   user@host# set vpn ipsec-vpn-chicago bind-interface st0.0
   ```

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

```
[edit]
user@host# show security ipsec
proposal ipsec-phase2-proposal {
   protocol esp;
   authentication-algorithm hmac-sha1-96;
   encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
   perfect-forward-secrecy {
      keys group2;
   }
   proposals ipsec-phase2-proposal;
}
vpn ipsec-vpn-chicago {
```

```
    bind-interface st0.0;
    ike {
      gateway gw-chicago;
      ipsec-policy ipsec-phase2-policy;
    }
  }
```

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

Configuring Security Policies

CLI Quick Configuration

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

```
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match
  source-address sunnyvale
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match
  destination-address chicago
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match
  application any
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi then permit
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match
  source-address chicago
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match
  destination-address sunnyvale
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match
  application any
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr then permit
```

Step-by-Step Procedure

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

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the vpn-chicago zone.

   ```
   [edit security policies from-zone trust to-zone vpn-chicago]
   user@host# set policy vpn-tr-chi match source-address sunnyvale
   user@host# set policy vpn-tr-chi match destination-address chicago
   user@host# set policy vpn-tr-chi match application any
   user@host# set policy vpn-tr-chi then permit
   ```

2. Create the security policy to permit traffic from the vpn-chicago zone to the trust zone.

   ```
   [edit security policies from-zone vpn-chicago to-zone trust]
   user@host# set policy vpn-chi-tr match source-address sunnyvale
   user@host# set policy vpn-chi-tr match destination-address chicago
   user@host# set policy vpn-chi-tr match application any
   user@host# set policy vpn-chi-tr then permit
   ```


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

```
[edit]
user@host# show security policies
from-zone trust to-zone vpn-chicago {
  policy vpn-tr-vpn {
    match {
      source-address sunnyvale;
      destination-address chicago;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vpn-chicago to-zone trust {
  policy vpn-tr-vpn {
    match {
      source-address chicago;
      destination-address sunnyvale;
      application any;
    }
    then {
      permit;
    }
  }
}
```

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

Configuring TCP-MSS

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

```
set security flow tcp-mss ipsec-vpn mss 1350
```

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

To configure TCP-MSS information:

1. Configure TCP-MSS information.

```
[edit]
user@host# set security flow tcp-mss ipsec-vpn mss 1350
```

Results

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

```
[edit]
user@host# show security flow
tcp-mss {
  ipsec-vpn {
    mss 1350;
  }
}
```

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

Configuring the SSG Series Device

CLI Quick Configuration

For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts & Examples ScreenOS Reference Guide*, which is located at http://www.juniper.net/techpubs.

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

Confirm that the configuration is working properly.

Verifying the IKE Phase 1 Status

Purpose Verify the IKE Phase 1 status.

Action

NOTE: 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 device 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

NOTE: 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 device 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.

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Hub-and-Spoke VPN on page 70
- Example: Configuring a Policy-Based VPN on page 206
- Understanding Internet Key Exchange Version 2 on page 105

Understanding Pico Cell Provisioning

Supported Platforms SRX Series, vSRX

IKEv2 configuration payload can be used to propagate provisioning information from an IKE responder, such as an SRX Series device, 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 device, 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 device
   - 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 device 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.

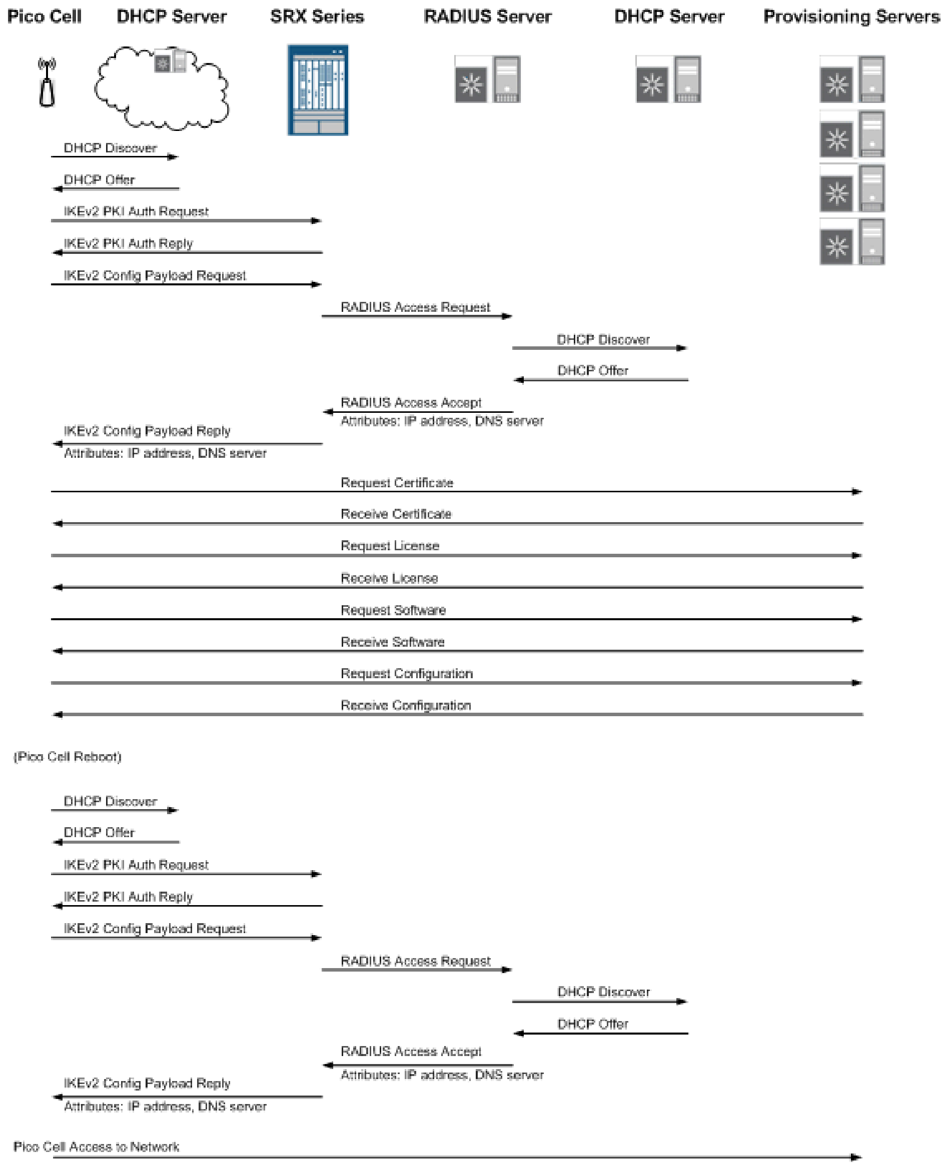
2. Pico cell provisioning—Using its assigned OAM traffic address, the pico cell requests its provisioning information—typically operator certificate, license, software, and configuration information—from servers within the protected network.

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 device: one for OAM traffic and the other for Third-Generation Partnership Project (3GPP) data traffic.

Figure 13 on page 127 shows a typical workflow for a pico cell deployment.

Figure 13: Typical Pico Cell Deployment Workflow



NOTE: The IKEv2 configuration payload feature is supported only for point-to-multipoint secure tunnel (st0) 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.

Related Documentation

- Understanding IKEv2 Configuration Payload on page 107
- Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload on page 128

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

Supported Platforms SRX Series, vSRX

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:

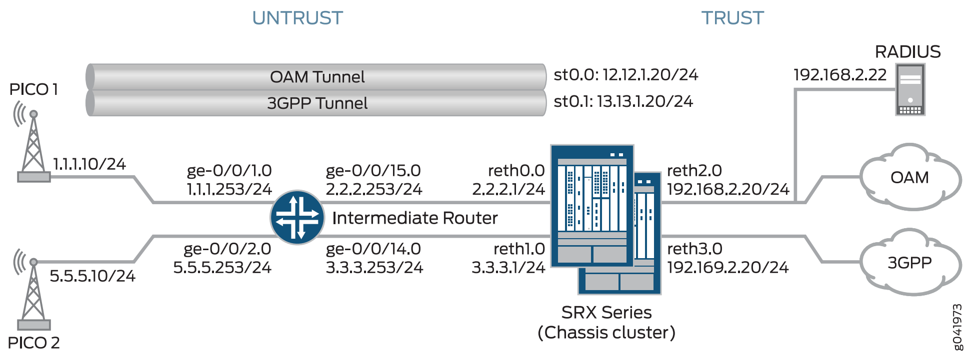
- Two SRX Series devices configured in a chassis cluster
- One SRX Series device configured as an intermediate router
- Two pico cell clients
- One RADIUS server configured with pico cell client provisioning information
- Junos OS Release 12.1X46-D10 or later for IKEv2 configuration payload support

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. The IKEv2 configuration payload feature is supported for IPv4 only.

Figure 14 on page 129 shows a topology in which the SRX Series supports pico cell provisioning using the IKEv2 configuration payload feature.

Figure 14: 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: 12.12.1.201/32, TSr: 192.168.2.0/24
Pico 1 3GPP	TSi: 13.13.1.201/32, TSr: 192.169.2.0/24, TSr: 13.13.0.0/16
Pico 2 OAM	TSi: 12.12.1.205/32, TSr: 192.168.2.0/24
Pico 2 3GPP	TSi: 13.13.1.205/32, TSr: 192.169.2.0/24, TSr: 13.13.0.0/16

NOTE: 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).

NOTE: The IKEv2 configuration payload feature is supported only for point-to-multipoint secure tunnel (st0) 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.

Table 25 on page 130 shows the Phase 1 and Phase 2 options configured on the SRX Series, including information for establishing both OAM and 3GPP tunnels.

Table 25: 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

Table 25: Phase 1 and Phase 2 Options for the SRX Series *(continued)*

Option	Value
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
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 25: 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	192.169.2.0/24
Remote proxy-identity	0.0.0.0/0
IPsec policy	IPSEC_POL

Certificates are stored on the pico cells and the SRX Series.

NOTE: 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

- Configuring the SRX Series on page 133
- Configuring the Intermediate Router on page 142

- Configuring the Pico Cell (Sample Configuration) on page 145
- Configuring the RADIUS Server (Sample Configuration) on page 147

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 2.2.2.1./24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 3.3.3.1/24
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 192.169.2.20/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 12.12.1.20/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 13.13.1.20/24
set routing-options static route 1.1.0.0/16 next-hop 2.2.2.253
set routing-options static route 5.5.0.0/16 next-hop 2.2.2.253
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 192.169.2.0/24
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 2.2.2.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 3.3.3.1/24
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 192.169.2.20/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 12.12.1.20/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 13.13.1.20/24
```

3. Configure routing options.

```
[edit routing-options]
user@host# set static route 1.1.0.0/16 next-hop 2.2.2.253
user@host# set static route 5.5.0.0/16 next-hop 2.2.2.253
```

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 192.169.2.0/24
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 2.2.2.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 3.3.3.1/24;
        }
    }
}
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 192.169.2.20/24;
        }
    }
}
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 1.1.0.0/16 next-hop 2.2.2.253;
    route 5.5.0.0/16 next-hop 2.2.2.253;
}
[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"; ## SECRET-DATA
    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 192.169.2.0/24;
            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 1.1.1.253/24
```

```
set interfaces ge-0/0/2 unit 0 family inet address 5.5.5.253/24
set interfaces ge-0/0/14 unit 0 family inet address 3.3.3.253/24
set interfaces ge-0/0/15 unit 0 family inet address 2.2.2.253/24
set routing-options static route 192.169.2.0/24 next-hop 2.2.2.1
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 1.1.1.253/24
   user@host# set ge-0/0/2 unit 0 family inet address 5.5.5.253/24
   user@host# set ge-0/0/14 unit 0 family inet address 3.3.3.253/24
   user@host# set ge-0/0/15 unit 0 family inet address 2.2.2.253/24
   ```

2. Configure routing options.

   ```
   [edit routing-options]
   user@host# set static route 192.169.2.0/24 next-hop 2.2.2.1
   ```

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 1.1.1.253/24;
      }
   }
}
ge-0/0/2 {
   unit 0 {
      family inet {
         address 5.5.5.253/24;
      }
   }
}
ge-0/0/14 {
   unit 0 {
      family inet {
         address 3.3.3.253/24;
      }
   }
}
ge-0/0/15 {
   unit 0 {
      family inet {
         address 2.2.2.253/24;
      }
   }
}
[edit]
user@host# show routing-options
static {
   route 192.169.2.0/24 next-hop 2.2.2.1;
}
[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:

```
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=2.2.2.1/24
        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=3.3.3.1/24
        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:

```
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=2.2.2.1/24
        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=3.3.3.1/24
```



```
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)

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.

NOTE: 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 device. This ensures the continuity of the client identity across the various protocols.

1. Review the RADIUS configuration for the Pico 1 OAM VPN. The RADIUS server has the following information:

```
DEFAULT  User-Name =~  "CN\=pico1\,\ C\=US\,\ ST\=CA\,\ L\=Sunnyvale$",
Cleartext-Password := "$ABC123"
        Service-Type = Framed-User,
        Framed-IP-Address = 12.12.1.201,
        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:

```
DEFAULT  User-Name =~  "C\=US\,\ ST\=CA\,\ L\=Sunnyvale\,\ O\=org\,\
OU=pico_cell,\ CN\=pico1\$", Cleartext-Password := "$ABC123"
        Service-Type = Framed-User,
        Framed-IP-Address = 13.13.1.201.10,
        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.

NOTE: The clear-text password is hard-coded and is not configurable. 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

Confirm that the configuration is working properly.

- Verifying the IKE Phase 1 Status for the SRX Series on page 148
- Verifying IPsec Security Associations for the SRX Series on page 150

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 1.1.1.1
1643848758 UP     9e31d4323195a195  4d142438106d4273  IKEv2 1.1.1.1

user@host# show security ike security-associations index 553329718 detail
node0:
--------------------------------------------------------------------------
IKE peer 1.1.1.1, 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: 2.2.2.1:500, Remote: 1.1.1.1:500
  Lifetime: Expires in 28738 seconds
  Peer ike-id: C=US, ST=CA, L=Sunnyvale, O=org, OU=pico_cell, CN=pico1
  Xauth assigned IP: 12.12.1.201
  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

NOTE: 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  1.1.1.1
  >214171651 ESP:aes-cbc-256/sha1 c0a54936 3529/        -  root 500  1.1.1.1
  <205520899 ESP:aes-cbc-256/sha1 84e49026 3521/        -  root 500  1.1.1.1
  >205520899 ESP:aes-cbc-256/sha1 c4ed1849 3521/        -  root 500  1.1.1.1


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: 3.3.3.1, Remote Gateway: 1.1.1.1
  Local Identity: list(any:0,ipv4_subnet(any:0-65535,[0..7]=192.169.2.0/24),
ipv4_subnet(any:0-65535,[0..7]=13.13.0.0/16))
  Remote Identity: ipv4(any:0,[0..3]=13.13.1.201)
    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: 2.2.2.1, Remote Gateway: 1.1.1.1
Local Identity: ipv4_subnet(any:0-65535,[0..7]=192.168.2.0/24)
Remote Identity: ipv4(any:0,[0..3]=12.12.1.201)
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 1.1.1.1.
- 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.

Related Documentation

- IPsec VPN Overview on page 3
- Understanding Internet Key Exchange Version 2 on page 105
- Understanding Certificates and PKI on page 353

Understanding IKEv2 Reauthentication

Supported Platforms SRX Series, vSRX

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.

Related Documentation

- Understanding Internet Key Exchange Version 2 on page 105

Understanding IKEv2 Fragmentation

Supported Platforms SRX Series, vSRX

- Overview on page 154
- Message Fragmentation on page 154

Overview

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.

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 devices, 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 devices, 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.

Related Documentation

- Understanding Internet Key Exchange Version 2 on page 105

CHAPTER 7

Configuring Secure Tunnel Interface in a Virtual Router

Understanding Virtual Router Support for Route-Based VPNs

Supported Platforms SRX Series

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 device
- Applications such as Application Layer Gateway (ALG), Intrusion Detection and Prevention (IDP), and Unified Threat Management (UTM)
- 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

Related Documentation

- Understanding Virtual Router Limitations on page 158
- Understanding Virtual Router Limitations on page 158

Understanding Virtual Router Limitations

Supported Platforms SRX Series, vSRX

The following features are not supported for virtual router (VR):

- Public key infrastructure (PKI) inside VR
- Chassis cluster active/active with VPN inside VR

When you configure VPN on SRX Series devices, 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.

Related Documentation

- Understanding Virtual Router Support for Route-Based VPNs on page 157
- IPsec VPN Overview on page 3

Example: Configuring an st0 Interface in a Virtual Router

Supported Platforms SRX Series

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 and Interfaces 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.

Configuration

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 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 default-policy permit-all
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 *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 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
```

3. 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"
```

4. 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
```

5. 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
```

6. 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
```

7. 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
```

8. Configure the VPN bind to tunnel interface.

```
[edit security ipsec]
user@host# set vpn first_vpn bind-interface st0.0
```

9. Configure the security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

10. 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
```

11. 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 {
    default-policy {
      permit-all;
    }
  }
  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

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.

Related Documentation

- Understanding Virtual Router Support for Route-Based VPNs on page 157

CHAPTER 8

Configuring Dual Stack Tunnels over an External Interface

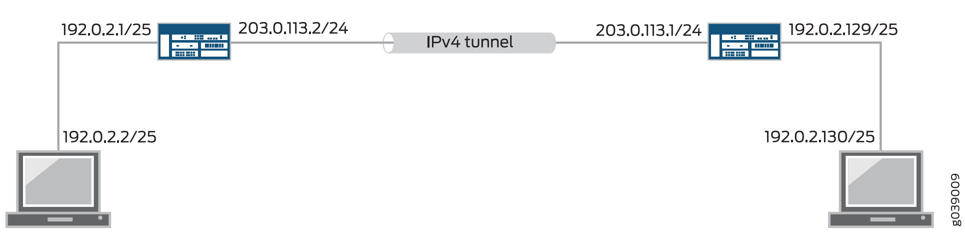
Understanding VPN Tunnel Modes

Supported Platforms SRX Series, vSRX

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 devices support four tunnel modes for route-based site-to-site VPNs.

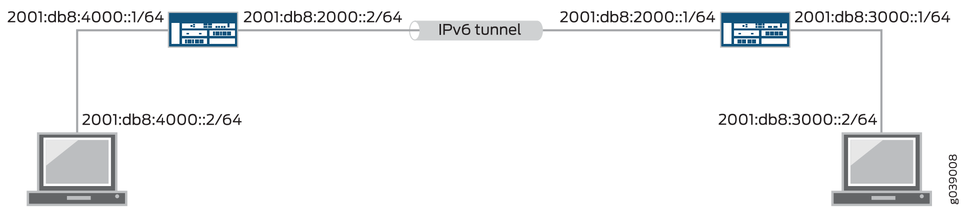
IPv4-in-IPv4 tunnels encapsulate IPv4 packets inside IPv4 packets, as shown in Figure 15 on page 165. The protocol fields for both the outer and the inner headers are IPv4.

Figure 15: IPv4-in-IPv4 Tunnel



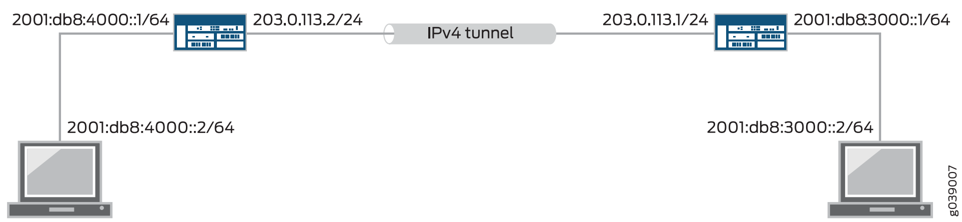
IPv6-in-IPv6 tunnels encapsulate IPv6 packets inside IPv6 packets, as shown in Figure 16 on page 166. The protocol fields for both the outer and inner headers are IPv6.

Figure 16: IPv6-in-IPv6 Tunnel



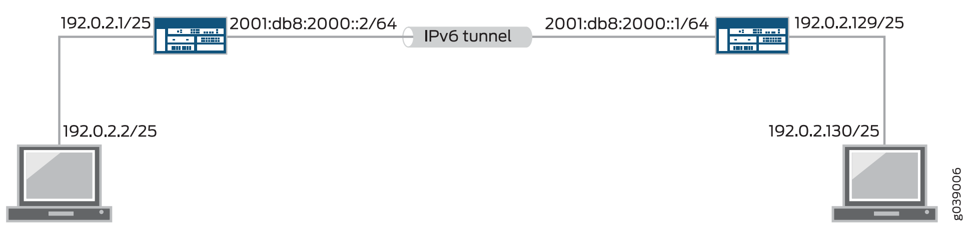
IPv6-in-IPv4 tunnels encapsulate IPv6 packets inside IPv4 packets, as shown in Figure 17 on page 166. The protocol field for the outer header is IPv4 and the protocol field for the inner header is IPv6.

Figure 17: IPv6-in-IPv4 Tunnel



IPv4-in-IPv6 tunnels encapsulate IPv4 packets inside IPv6 packets, as shown in Figure 18 on page 166. The protocol field for the outer header is IPv6 and the protocol field for the inner header is IPv4.

Figure 18: 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.

NOTE: 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.

Related Documentation

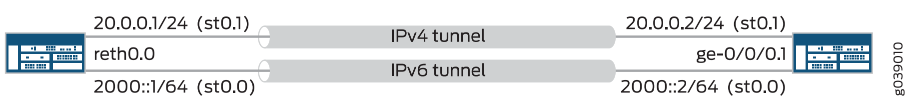
- VPN Feature Support for IPv6 Addresses on page 319
- Understanding Dual-Stack Tunnels over an External Interface on page 167
- Understanding IPv6 IKE and IPsec Packet Processing on page 323

Understanding Dual-Stack Tunnels over an External Interface

Supported Platforms SRX Series

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 19 on page 167, the physical interfaces reth0.0 and ge-0/0/0.1 support parallel IPv4 and IPv6 tunnels between two devices.

Figure 19: Dual-Stack Tunnels



NOTE: In Figure 19 on page 167, 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.

NOTE: The **local-address** value must be an IP address that is configured on an interface on the SRX Series device. 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.

Related Documentation

- Example: Configuring Dual-Stack Tunnels over an External Interface on page 168
- Understanding VPN Tunnel Modes on page 165
- VPN Feature Support for IPv6 Addresses on page 319

Example: Configuring Dual-Stack Tunnels over an External Interface

Supported Platforms SRX Series

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 Dual-Stack Tunnels over an External Interface" on page 167.

NOTE: The configuration shown in this example is only supported with route-based site-to-site VPNs.

Overview

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 26 on page 169 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 26: 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
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 27 on page 170 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 27: 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.

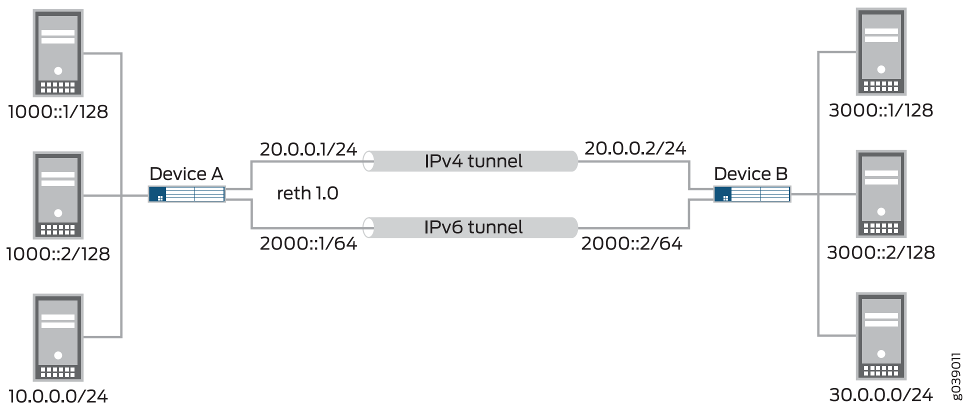
A static route is configured in the default (IPv4) routing table to route IPv4 traffic to 30.0.0.0/24 through st0.1.

NOTE: **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 20 on page 171, the SRX Series device A supports IPv4 and IPv6 tunnels to device B. IPv6 traffic to 3000::1/128 is routed through the IPv4 tunnel, while IPv6 traffic to 3000::2/128 and IPv4 traffic to 30.0.0.0/24 are routed through the IPv6 tunnel.

Figure 20: Dual-Stack Tunnel Example



Configuration

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 *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

Confirm that the configuration is working properly.

- Verifying IKE Phase 1 Status on page 176
- Verifying IPsec Phase 2 Status on page 176
- Verifying Routes on page 176

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

- Understanding Dual-Stack Tunnels over an External Interface on page 167
- Understanding VPN Tunnel Modes on page 165

CHAPTER 9

Configuring Traffic Selectors in Route-Based VPNs

Understanding Traffic Selectors in Route-Based VPNs

Supported Platforms SRX Series, vSRX

A traffic selector (also known as a *proxy ID* in IKEv1) 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 an SA.

Traffic Selector Configuration

To configure a traffic selector, use the **traffic-selector** configuration statement at the [**edit security ipsec vpn *vpn-name***] hierarchy level. The traffic selector is defined with the mandatory **local-ip *ip-address*** and **remote-ip *ip-address*** 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 or subnetwork 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.

NOTE: **Traffic selectors on AutoVPN hubs can only be configured with IPv4 addresses. IPv4-in-IPv6, IPv6-in-IPv6, or IPv6-in-IPv4 tunnel modes are not supported for AutoVPN with traffic selectors.**

When traffic selectors are configured, static routes are automatically added during configuration processing or when traffic selectors are negotiated; this process is known as auto route insertion (ARI). These routes might conflict with those that are populated through routing protocols. We recommend that you do not configure routing protocols on st0 interfaces that are bound to VPNs where traffic selectors are configured.

When a traffic selector is deleted, all corresponding IPsec SAs, routes, and tunnel sessions are cleared. This might affect traffic passing through these tunnels.

When a traffic selector is modified, deleted, or added, traffic selectors that follow it in the configuration are affected. The tunnels, SAs, and routes are cleared and reinstalled. Traffic selectors that precede the new or modified traffic selector in the configuration are unaffected.

For example, three traffic selectors are configured for the same VPN in the following order:

1. ts-red
2. ts-blue
3. ts-green

Changes in the traffic selector configuration have the following results:

Action	Result
Modify local or remote IP address in ts-blue.	Tunnels, SAs, and routes for ts-blue and ts-green are cleared. Tunnel, SA, and route for ts-red are not affected.
Delete ts-blue.	Tunnel, SA, and route for ts-green are cleared. Tunnel, SA, and route for ts-red are not affected.
Insert ts-white after ts-blue.	Tunnels, SAs, and routes for ts-white and ts-green are cleared. Tunnels, SAs, and routes for ts-red and ts-blue are not affected.

Traffic Selector Flexible Matches

During IKE negotiation, the responder can accept from the initiator a proposed traffic selector that is a subset of the traffic selector configured on the responder. There can be a wide subnetwork configured in a traffic selector on the hub and a narrow portion of the subnetwork configured in a traffic selector on each spoke. For example, consider the following traffic selectors configured on an AutoVPN hub and spoke:

	Local IP	Remote IP
Spoke:	10.30.1.0/24	10.40.1.0/24
Hub:	10.40.0.0/16	10.30.0.0/16

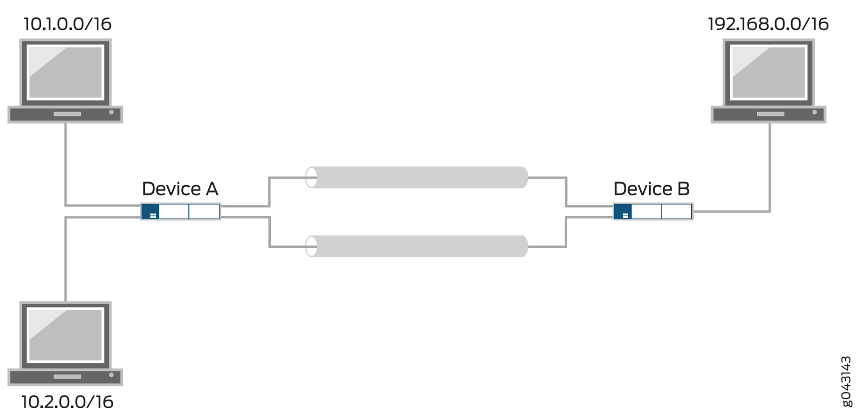
The initiator (the spoke) negotiates with the responder (the hub) using the /24 netmask. Because the traffic selector IP addresses configured on the spoke are subsets of the traffic selector configured on the hub, the hub accepts the negotiation. The /24 subnetworks are used to match traffic for the negotiated tunnel and the 10.30.1.0/24 static route to the spoke is installed through ARI in the hub's routing table.

Traffic selector flexible matches are supported for both IKEv1 and IKEv2.

Multiple Tunnels for Traffic Selector Configuration

A single traffic selector configuration can result in multiple tunnels. In Figure 21 on page 181, subnetworks 10.1.0.0/16 and 10.2.0.0/16 are behind device A while subnetwork 192.168.0.0/16 is behind device B.

Figure 21: Multiple Tunnels for Traffic Selector Configuration



Two traffic selectors are configured on device A and one traffic selector is configured on device B, as follows:

	Local IP address	Remote IP address

Device A traffic selectors		
TS1:	10.1.0.0/16	192.168.0.0/16
TS2:	10.2.0.0/16	192.168.0.0/16
Device B traffic selector		
TS:	192.168.0.0/16	10.0.0.0/8

Device A initiates two tunnels with device B using its proposed traffic selectors. Both of device A's proposed traffic selectors match the configured traffic selector on device B. Device B creates two different tunnels for its single traffic selector configuration.

Limitations

Traffic selectors cannot be configured with the following features:

- Policy-based VPNs
- Shared IKE IDs
- VPN monitoring
- Different address families configured for the local and remote IP addresses
- A remote address of 0.0.0.0/0 (IPv4) or 0::0 (IPv6)
- IKEv2 site-to-site VPN
- Dynamic routing protocols configured on st0 interfaces

Related Documentation

- Understanding Auto Route Insertion on page 198
- Understanding AutoVPN with Traffic Selectors on page 623
- Understanding VPN Tunnel Modes on page 165
- Understanding Traffic Selectors and Overlapping IP Addresses on page 199

Example: Configuring Traffic Selectors in a Route-Based VPN

Supported Platforms SRX Series, vSRX

This example shows how to configure traffic selectors for a route-based VPN.

- Requirements on page 183
- Overview on page 183
- Configuration on page 185
- Verification on page 195

Requirements

Before you begin, read “Understanding Traffic Selectors in Route-Based VPNs” on page 179.

Overview

This example configures traffic selectors to allow traffic to flow between subnetworks on SRX_A and subnetworks on SRX_B.

Table 28 on page 183 shows the traffic selectors used in this example. Traffic selectors are configured with other Phase 2 options (shown in Table 30 on page 184).

Table 28: 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
TS3-ipv4	192.168.10.0/24	192.168.20.0/24	TS3-ipv4	192.168.20.0/24	192.168.10.0/24

Table 29 on page 183 shows the Phase 1 options used in this example. The Phase 1 option configuration on each device includes an IKE gateway configuration to the IPv6 peer.

Table 29: Phase 1 Options for Traffic Selector Configurations

Option	SRX_A	SRX_B
IKE proposal	phase1_psk_proposal	phase1_psk_proposal
Authentication method	preshared keys	preshared keys
DH group	group2	group2
Authentication algorithm	SHA 1	SHA 1
Encryption algorithm	3DES CBC	3DES CBC
Lifetime	180 seconds	180 seconds
IKE policy	ike_psk_policy	ike_psk_policy
Mode	main	main
Proposal	phase1_psk_proposal	phase1_psk_proposal
Preshared key	$ABC123	$ABC123

Table 29: Phase 1 Options for Traffic Selector Configurations *(continued)*

Option	SRX_A	SRX_B
IKE gateway	ike-gateway-to-he-srx	ike-gateway-to-branch-srx
IKE policy	ike_psk_policy	ike_psk_policy
Gateway address	2001:db8:2000::2	2001:db8:2000::1
External interface	ge-0/0/1.0	ge-0/0/1.0
Local address	2001:db8:2000::1	2001:db8:2000::2
IKE version	v1-only (default)	v1-only (default)

Table 30 on page 184 shows the Phase 2 options used in this example. Traffic selectors shown in Table 28 on page 183 are configured with the Phase 2 options.

Table 30: Phase 2 Options for Traffic Selector Configurations

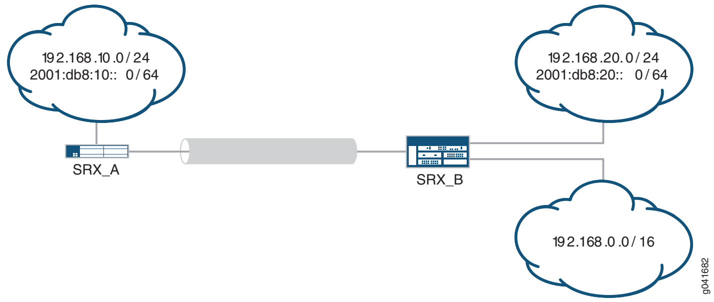
Option	SRX_A	SRX_B
IPsec proposal	phase2-proposal	phase2-proposal
Protocol	ESP	ESP
Authentication algorithm	HMAC SHA-1 96	HMAC SHA-1 96
Encryption algorithm	3DES CBC	3DES CBC
IPsec policy	ipsec-policy	ipsec-policy
Proposal	phase2-proposal	phase2-proposal
VPN	ipsec-vpn-to-he-srx	ipsec-vpn-to-branch-srx
Bind interface	st0.1	st0.1
IKE gateway	ike-gateway-to-he-srx	ike-gateway-to-branch-srx
IPsec policy	ipsec-policy	ipsec-policy

NOTE: On both devices, 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 22 on page 185, 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 22: Traffic Selector Configuration Example



Configuration

- Configuring SRX_A on page 185
- Configuring SRX_B on page 190

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 phase1_psk_proposal authentication- method pre-shared-keys
set security ike proposal phase1_psk_proposal dh-group group2
set security ike proposal phase1_psk_proposal authentication- algorithm sha1
set security ike proposal phase1_psk_proposal encryption-algorithm 3des-cbc
set security ike proposal phase1_psk_proposal lifetime-seconds 180
set security ike policy ike_psk_policy mode main
set security ike policy ike_psk_policy proposals phase1_psk_proposal
set security ike policy ike_psk_policy pre-shared-key ascii-text "$ABC123"
set security ike gateway ike-gateway-to-he-srx ike-policy ike_psk_policy
set security ike gateway ike-gateway-to-he-srx address 2001:db8:2000::2
set security ike gateway ike-gateway-to-he-srx external-interface ge-0/0/1.0
set security ike gateway ike-gateway-to-he-srx local-address 2001:db8:2000::1
set security ipsec proposal phase2-proposal protocol esp
set security ipsec proposal phase2-proposal authentication- algorithm hmac-sha1-96
set security ipsec proposal phase2-proposal encryption-algorithm 3des-cbc
set security ipsec policy ipsec_policy proposals phase2-proposal
set security ipsec vpn ipsec-vpn-to-he-srx bind-interface st0.1
```

```
set security ipsec vpn ipsec-vpn-to-he-srx ike ipsec-policy ipsec_policy
set security ipsec vpn ipsec-vpn-to-he-srx ike gateway ike- gateway-to-he-srx
set security ipsec vpn ipsec-vpn-to-he-srx traffic-selector TS1- ipv6 local-ip
   2001:db8:10::0/64 remote-ip 2001:db8:20::0/64
set security ipsec vpn ipsec-vpn-to-he-srx traffic-selector TS2- ipv4 local-ip
   192.168.10.0/24 remote-ip 192.168.0.0/16
set security ipsec vpn ipsec-vpn-to-he-srx traffic-selector TS3- ipv4 local-ip
   192.168.10.0/24 remote-ip 192.168.20.0/24
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 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 st0.1
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* 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 phase1_psk_proposal]
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
user@host# set lifetime-seconds 180

[edit security ike policy ike_psk_policy]
user@host# set mode main
user@host# set proposals phase1_psk_proposal
user@host# set pre-shared-key ascii-text "$ABC123"
```

```
[edit security ike gateway ike-gateway-to-he-srx]
user@host# set ike-policy ike_psk_policy
user@host# set address 2001:db8:2000::2
user@host# set external-interface ge-0/0/1.0
user@host# set local-address 2001:db8:2000::1
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal phase2-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 phase2-proposal

[edit security ipsec vpn ipsec-vpn-to-he-srx]
user@host# set bind-interface st0.1
user@host# set ike gateway ike-gateway-to-he-srx
user@host# set ike ipsec-policy ipsec_policy
user@host# set traffic-selector TS1-ipv6 local-ip 2001:db8:10::0/64 remote-ip
  2001:db8:20::0/64
user@host# set traffic-selector TS2-ipv4 local-ip 192.168.10.0/24 remote-ip
  192.168.0.0/16
user@host# set traffic-selector TS3-ipv4 local-ip 192.168.10.0/24 remote-ip
  192.168.20.0/24
```

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 all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
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 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 phase1_psk_proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 180;
  }
  policy ike_psk_policy {
    mode main;
    proposals phase1_psk_proposal;
    pre-shared-key ascii-text
    "$ABC123"; ## SECRET-DATA
  }
  gateway ike-gateway-to-he-srx {
    ike-policy ike_psk_policy;
    address 2001:db8:2000::2;
    external-interface ge-0/0/1.0;
    local-address 2001:db8:2000::1;
  }
  [edit]
  user@host# show security ipsec
  proposal phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
```



```
}
policy ipsec_policy {
   proposals phase2-proposal;
}
vpn ipsec-vpn-to-he-srx {
   bind-interface st0.1;
   ike {
      ipsec-policy ipsec_policy;
      gateway ike-gateway-to-he-srx;
   }
   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;
   }
   traffic-selector TS3-ipv4 {
      local-ip 192.168.10.0/24;
      remote-ip 192.168.20.0/24;
   }
}
[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;
   }
}
security-zone untrust {
   host-inbound-traffic {
      system-services {
         all;
      }
      protocols {
         all;
      }
   }
   interfaces {
      st0.1;
```

```
        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.

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 192.168.1.1/24
set security ike proposal phase1_psk_proposal authentication-method pre-shared-keys
set security ike proposal phase1_psk_proposal dh-group group2
set security ike proposal phase1_psk_proposal authentication-algorithm sha1
set security ike proposal phase1_psk_proposal encryption-algorithm 3des-cbc
set security ike proposal phase1_psk_proposal lifetime-seconds 180
set security ike policy ike_psk_policy mode main
set security ike policy ike_psk_policy proposals phase1_psk_proposal
set security ike policy ike_psk_policy pre-shared-key ascii-text "$ABC123"
set security ike gateway ike-gateway-to-branch-srx ike-policy ike_psk_policy
set security ike gateway ike-gateway-to-branch-srx address 2001:db8:2000::1
set security ike gateway ike-gateway-to-branch-srx external-interface ge-0/0/1.0
set security ike gateway ike-gateway-to-branch-srx local-address 2001:db8:2000::2
set security ipsec proposal phase2-proposal protocol esp
set security ipsec proposal phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal phase2-proposal encryption-algorithm 3des-cbc
set security ipsec policy ipsec_policy proposals phase2-proposal
set security ipsec vpn ipsec-vpn-to-branch-srx bind-interface st0.1
set security ipsec vpn ipsec-vpn-to-branch-srx ike ipsec-policy ipsec_policy
set security ipsec vpn ipsec-vpn-to-branch-srx ike gateway ike-gateway-to-branch-srx
set security ipsec vpn ipsec-vpn-to-branch-srx traffic-selector TS1-ipv6 local-ip
  2001:db8:20::0/64 remote-ip 2001:db8:10::0/64
set security ipsec vpn ipsec-vpn-to-branch-srx traffic-selector TS2-ipv4 local-ip
  192.168.0.0/16 remote-ip 192.168.10.0/24
set security ipsec vpn ipsec-vpn-to-branch-srx traffic-selector TS3-ipv4 local-ip
  192.168.20.0/24 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 all
```



```
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.1
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 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 192.168.1.1/24
```

4. Configure Phase 1 options.

```
[edit security ike proposal phase1_psk_proposal]
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
user@host# set lifetime-seconds 180

[edit security ike policy ike_psk_policy]
user@host# set mode main
user@host# set proposals phase1_psk_proposal
user@host# set pre-shared-key ascii-text "$ABC123"

[edit security ike gateway ike-gateway-to-branch-srx]
user@host# set ike-policy ike_psk_policy
user@host# set address 2001:db8:2000::1
user@host# set external-interface ge-0/0/1.0
user@host# set local-address 2001:db8:2000::2
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal phase2-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 phase2-proposal**

[edit security ipsec vpn ipsec-vpn-to-branch-srx]
user@host# **set bind-interface st0.1**
user@host# **set ike gateway ike-gateway-to-branch-srx**
user@host# **set ike ipsec-policy ipsec_policy**
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**
user@host# **set traffic-selector TS3-ipv4 local-ip 192.168.20.0/24 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**
   user@host# **set interfaces ge-1/1/1.0**

   [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 st0.1**
   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 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 192.168.1.1/24;
        }
    }
}
st0 {
    unit 1 {
        family inet;
        family inet6;
    }
}
[edit]
user@host# show security ike
proposal phase1_psk_proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 180;
}
policy ike_psk_policy {
    mode main;
    proposals phase1_psk_proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway ike-gateway-to-branch-srx {
    ike-policy ike_psk_policy;
    address 2001:db8:2000::1;
    external-interface ge-0/0/1.0;
    local-address 2001:db8:2000::2;
}
[edit]
user@host# show security ipsec
proposal phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
}
policy ipsec_policy {
    proposals phase2-proposal;
}
vpn ipsec-vpn-to-branch-srx {
    bind-interface st0.1;
    ike {
```



```
            ipsec-policy ipsec_policy;
            gateway ike-gateway-to-branch-srx;
        }
        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;
        }
        traffic-selector TS3-ipv4 {
            local-ip 192.168.20.0/24;
            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 {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            st0.1;
            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

Confirm that the configuration is working properly.

NOTE: The sample outputs shown are on SRX-A.

- Verifying IPsec Phase 2 Status on page 195
- Verifying Traffic Selectors on page 197
- Verifying Routes on page 197

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/ sha1 3d75aeff 2984/ unlim -  root 500   2001:db8:2000::2
  >268173313 ESP:3des/ sha1 a468fece 2984/ unlim -  root 500   2001:db8:2000::2
  <268173316 ESP:3des/ sha1 417f3cea 3594/ unlim -  root 500   2001:db8:2000::2
  >268173316 ESP:3des/ sha1 a4344027 3594/ unlim -  root 500   2001:db8:2000::2
  <268173317 ESP:3des/ sha1 cc9fb573 3556/ unlim -  root 500   2001:db8:2000::2
  >268173317 ESP:3des/ sha1 a4bde69b 3556/ 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: ipsec-vpn-to-he-srx
  Local Gateway: 2001:db8:2000::1, Remote Gateway: 2001:db8:2000::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-sha1-96, Encryption: 3des-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-sha1-96, Encryption: 3des-cbc
  Anti-replay service: counter-based enabled, Replay window size: 64

ID: 268173316 Virtual-system: root, VPN Name: ipsec-vpn-to-he-srx
Local Gateway: 2001:db8:2000::1, Remote Gateway: 2001:db8:2000::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-sha1-96, Encryption: 3des-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-sha1-96, Encryption: 3des-cbc
  Anti-replay service: counter-based enabled, Replay window size: 64

ID: 268173317 Virtual-system: root, VPN Name: ipsec-vpn-to-he-srx
Local Gateway: 2000::1, Remote Gateway: 2000::2
Traffic Selector Name: TS3-ipv4
Local Identity: ipv4(192.168.10.0-192.168.10.255)
Remote Identity: ipv4(192.168.0.0-192.168.255.255)
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: cc9fb573, AUX-SPI: 0
                           , VPN Monitoring: -
  Hard lifetime: Expires in 3548 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 2925 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: a4bde69b, AUX-SPI: 0
```



```
                                    , VPN Monitoring: -
    Hard lifetime: Expires in 3548 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2925 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 **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        *[Static/5] 00:00:32
                    > via st0.1
192.168.20.0/24        *[Static/5] 00:00:32
                    > via st0.1
2001:db8:20::0/64      *[Static/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.

Related Documentation

- Understanding Traffic Selectors in Route-Based VPNs on page 179

Understanding Auto Route Insertion

Supported Platforms SRX Series, vSRX

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.

NOTE: 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.

NOTE: 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.

Related Documentation

- Understanding Traffic Selectors in Route-Based VPNs on page 179
- Understanding AutoVPN with Traffic Selectors on page 623

Understanding Traffic Selectors and Overlapping IP Addresses

Supported Platforms SRX Series, vSRX

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.

NOTE: 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.

Related Documentation

- Understanding Traffic Selectors in Route-Based VPNs on page 179

PART 3

Configuring Policy-Based IPsec VPNs

CHAPTER 10

Configuring Policy-Based VPNs

Understanding Policy-Based IPsec VPNs

Supported Platforms SRX Series, vSRX

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.

NOTE: **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.**

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Route-Based VPN on page 50
- Example: Configuring a Hub-and-Spoke VPN on page 70
- Example: Configuring a Policy-Based VPN on page 206

Example: Configuring a Policy-Based VPN

Supported Platforms SRX Series, vSRX

This example shows how to configure a policy-based IPsec VPN to allow data to be securely transferred between a branch office and the corporate office.

Requirements

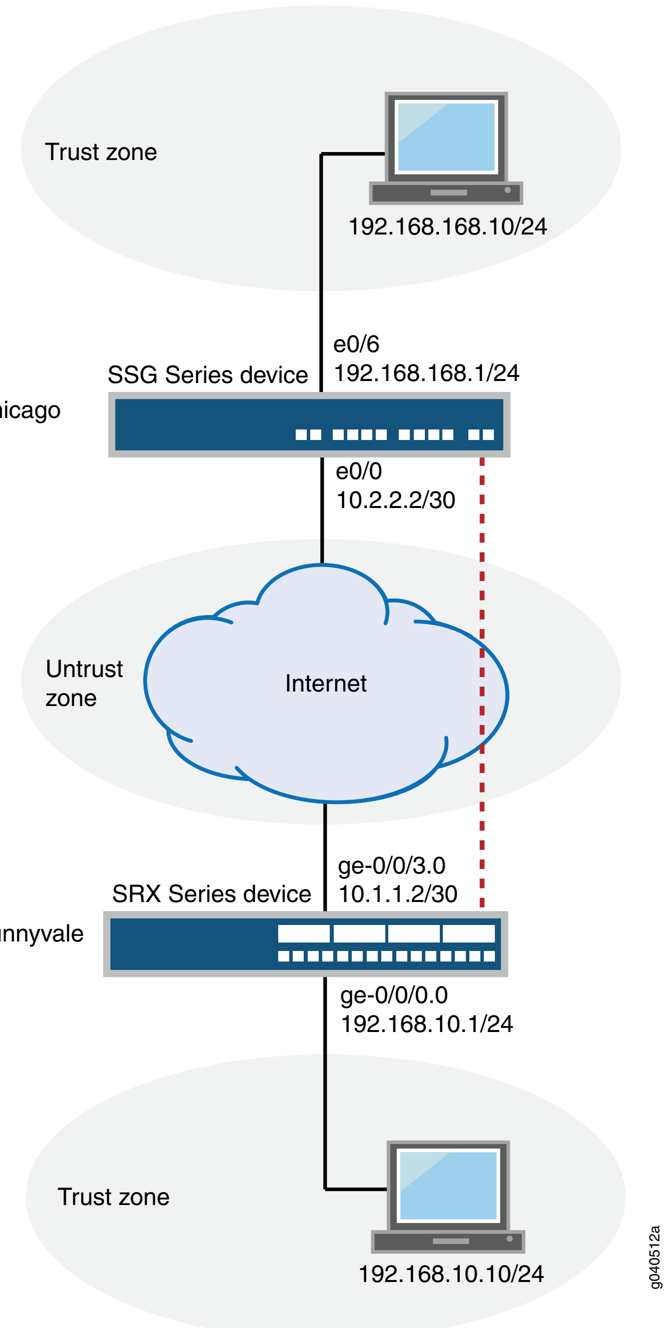
Before you begin, read "IPsec VPN Overview" on page 3.

Overview

In this example, you configure a 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 23 on page 207 shows an example of a policy-based VPN topology. In this topology, the SRX Series device is located in Sunnyvale, and an SSG Series device (or it can be another third-party device) is located in Chicago.

Figure 23: 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, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, security policy, and TCP-MSS parameters. See Table 31 on page 208 through Table 35 on page 210.

Table 31: Interface, 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
Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.
	untrust	• IKE is the only allowed system service. • The ge-0/0/3.0 interface is bound to this zone.
Address book entries	sunnyvale	• This address is an entry in the address book **book1**, which is attached to a zone called **trust**. • The address for this address book entry is 192.168.10.0/24.
	chicago	• This address is an entry in the address book **book2**, which is attached to a zone called **untrust**. • The address for this address book entry is 192.168.168.0/24.

Table 32: IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw-chicago	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2

Table 33: IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2
VPN	ike-vpn-chicago	• IKE gateway reference: gw-chicago • IPsec policy reference: ipsec-phase2-policy

Table 34: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

This security policy permits traffic from the trust zone to the untrust zone.	vpn-tr-untr	• Match criteria:
		  ◦ source-address sunnyvale
		  ◦ destination-address chicago
		  ◦ application any
		• Permit action: tunnel ipsec-vpn ike-vpn-chicago
		• Permit action: tunnel pair-policy vpn-untr-tr

This security policy permits traffic from the untrust zone to the trust zone.	vpn-untr-tr	• Match criteria:
		  ◦ source-address chicago
		  ◦ destination-address sunnyvale
		  ◦ application any
		• Permit action: tunnel ipsec-vpn ike-vpn-chicago
		• Permit action: tunnel pair-policy vpn-tr-untr

This security policy permits all traffic from the trust zone to the untrust zone.
 	
NOTE: You must put the 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 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 vpn-tr-untr policy.
	permit-any	• Match criteria:
		  ◦ source-address any
		  ◦ source-destination any
		  ◦ application any
		• Action: permit

Table 35: 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. NOTE: 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

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-0/0/0 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 routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
set security zones security-zone untrust interfaces ge-0/0/3.0
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 address-book book1 address sunnyvale 192.168.10.0/24
set security address-book book1 attach zone trust
set security address-book book2 address chicago 192.168.168.0/24
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-0/0/0 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
```

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 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 security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/0.0
```

8. Specify allowed system services for the security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

9. Create an address book and attach it to a zone.

```
[edit security address-book book1]
user@host# set address sunnyvale 192.168.10.0/24
user@host# set attach zone trust
```

10. Create another address book and attach it to a zone.

```
[edit security address-book book2]
user@host# set address chicago 192.168.168.0/24
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-0/0/0 {
  unit 0 {
    family inet {
```

```
            address 192.168.10.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.1.2/30
        }
    }
}
[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-0/0/3.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
[edit]
user@host# show security address-book
book1 {
    address sunnyvale 192.168.10.0/24;
    attach {
        zone trust;
    }
}
    book2 {
        address chicago 192.168.168.0/24;
        attach {
            zone untrust;
        }
    }
```

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

Configuring IKE

CLI Quick Configuration

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

```
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy 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-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
```

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. Set the IKE Phase 1 policy mode.

   [edit security ike policy ike-phase1-policy]
   user@host# **set mode main**

8. Specify a reference to the IKE proposal.

   [edit security ike policy ike-phase1-policy]
   user@host# **set proposals ike-phase1-proposal**

9. Define the IKE Phase 1 policy authentication method.

   [edit security ike policy ike-phase1-policy]
   user@host# **set pre-shared-key ascii-text “$ABC123”**

10. Create an IKE Phase 1 gateway and define its external interface.

    [edit security ike gateway gw-chicago]
    user@host# **set external-interface ge-0/0/3.0**

11. Define the IKE Phase 1 policy reference.

    [edit security ike gateway gw-chicago]
    user@host# **set ike-policy ike-phase1-policy**

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 {
   mode main;
   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;
}
```

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 ike-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ike-vpn-chicago ike 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 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 ike-vpn-chicago ike gateway gw-chicago
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn ike-vpn-chicago ike ipsec-policy 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 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-chicago {
  ike {
    gateway gw-chicago;
    ipsec-policy ipsec-phase2-policy;
  }
}
```

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

Configuring Security Policies

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

```
set security policies from-zone trust to-zone untrust policy vpn-tr-untr match
  source-address sunnyvale
set security policies from-zone trust to-zone untrust policy vpn-tr-untr match
  destination-address chicago
set security policies from-zone trust to-zone untrust policy vpn-tr-untr match application
  any
```



```
set security policies from-zone trust to-zone untrust policy vpn-tr-untr then permit tunnel
   ipsec-vpn ike-vpn-chicago
set security policies from-zone trust to-zone untrust policy vpn-tr-untr then permit tunnel
   pair-policy vpn-untr-tr
set security policies from-zone untrust to-zone trust policy vpn-untr-tr match
   source-address chicago
set security policies from-zone untrust to-zone trust policy vpn-untr-tr match
   destination-address sunnyvale
set security policies from-zone untrust to-zone trust policy vpn-untr-tr match application
   any
set security policies from-zone untrust to-zone trust policy vpn-untr-tr then permit tunnel
   ipsec-vpn ike-vpn-chicago
set security policies from-zone untrust to-zone trust policy vpn-untr-tr then permit tunnel
   pair-policy 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 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 vpn-tr-untr match source-address sunnyvale
user@host# set policy vpn-tr-untr match destination-address chicago
user@host# set policy vpn-tr-untr match application any
user@host# set policy vpn-tr-untr then permit tunnel ipsec-vpn ike-vpn-chicago
user@host# set policy vpn-tr-untr then permit tunnel pair-policy 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 vpn-untr-tr match source-address chicago
user@host# set policy vpn-untr-tr match destination-address sunnyvale
user@host# set policy vpn-untr-tr match application any
user@host# set policy vpn-untr-tr then permit tunnel ipsec-vpn ike-vpn-chicago
user@host# set policy vpn-untr-tr then permit tunnel pair-policy 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 vpn-untr-tr match destination-address any
user@host# set policy vpn-untr-tr match application any
user@host# set policy vpn-untr-tr 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 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 vpn-tr-untr {
      match {
         source-address sunnyvale;
         destination-address chicago;
         application any;
      }
      then {
         permit {
            tunnel {
               ipsec-vpn ike-vpn-chicago;
               pair-policy vpn-untr-tr;
            }
         }
      }
   }
   policy permit-any {
      match {
         source-address any;
         destination-address any;
         application any;
      }
      then {
         permit
      }
   }
}
from-zone untrust to-zone trust {
   policy vpn-untr-tr {
      match {
         source-address chicago;
         destination-address sunnyvale;
         application any;
      }
      then {
         permit {
            tunnel {
               ipsec-vpn ike-vpn-chicago;
               pair-policy 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.

Configuring the SSG Series Device

CLI Quick Configuration

For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts and Examples ScreenOS Reference Guide*, which is located at http://www.juniper.net/techpubs.

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 interface ethernet0/6 zone Trust
set interface ethernet0/0 zone Untrust
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 flow tcp-mss 1350
set address Trust “local-net” 192.168.168.0 255.255.255.0
set address Untrust "corp-net" 192.168.10.0 255.255.255.0
set ike gateway corp-ike address 10.1.1.2 Main outgoing-interface ethernet0/0 preshare 395psksecr3t sec-level standard
set vpn corp-vpn gateway corp-ike replay tunnel idletime 0 sec-level standard
set policy id 11 from Trust to Untrust “local-net” “corp-net” “ANY” tunnel vpn “corp-vpn” pair-policy 10
set policy id 10 from Untrust to Trust “corp-net” “local-net” “ANY” tunnel vpn “corp-vpn” pair-policy 11
set policy id 1 from Trust to Untrust “ANY” “ANY” “ANY” nat src permit
set route 0.0.0.0/0 interface ethernet0/0 gateway 10.2.2.1

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status on page 220
- Verifying the IPsec Phase 2 Status on page 222
- Reviewing Statistics and Errors for an IPsec Security Association on page 223

Verifying the IKE Phase 1 Status

Purpose Verify the IKE Phase 1 status.

Action

NOTE: 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 policy-based VPNs, a separate host must generate the traffic; traffic initiated from the SRX Series device 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 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
4       10.2.2.2         UP     5e1db3f9d50b0de6  e50865d9ebf134f8  Main
```

```
user@host> show security ike security-associations index 4 detail
IKE peer 10.2.2.2, Index 4,
  Role: Responder, State: UP
  Initiator cookie: 5e1db3f9d50b0de6, Responder cookie: e50865d9ebf134f8
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.1.2:500, Remote: 10.2.2.2:500
  Lifetime: Expires in 28770 seconds
  Algorithms:
```



```
 Authentication        : sha1
 Encryption            : aes-128-cbc
 Pseudo random function: hmac-sha1
Traffic statistics:
 Input bytes    :                 852
 Output bytes   :                 856
 Input packets  :                   5
 Output packets :                   4
Flags: Caller notification sent
IPSec security associations: 1 created, 0 deleted
Phase 2 negotiations in progress: 0
```

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 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

NOTE: 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    Gateway         Port  Algorithm         SPI      Life:sec/kb  Mon vsys
  <2    10.2.2.2         500   ESP:aes-128/sha1   a63eb26f 3565/ unlim   -   0
  >2    10.2.2.2         500   ESP:aes-128/sha1   a1024ed9 3565/ unlim   -   0
```

```
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: vpnpolicy-unt-tr

    Direction: inbound, SPI: 2789126767, AUX-SPI: 0
    Hard lifetime: Expires in 3558 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 2986 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: 2701283033,, AUX-SPI: 0
    Hard lifetime: Expires in 3558 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 2986 seconds
    Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc
    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 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 3565/ unlim value indicates that the Phase 2 lifetime expires in 3565

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 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 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.

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 *index_number*** command, using the index number of the VPN for which you want to see statistics.

```
user@host> show security ipsec statistics index 2
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 if the other error counters are incrementing.

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Route-Based VPN on page 50
- Example: Configuring a Hub-and-Spoke VPN on page 70

PART 4

Configuring VPNs with NAT-T

CHAPTER 11

Configuring Route-Based and Policy-Based VPNs with NAT-T

Understanding NAT-T

Supported Platforms SRX Series, vSRX

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.

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.

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring a Route-Based VPN with Only the Responder Behind a NAT Device on page 228
- Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device on page 257
- Example: Configuring NAT-T with Dynamic Endpoint VPN on page 286

Example: Configuring a Route-Based VPN with Only the Responder Behind a NAT Device

Supported Platforms SRX Series, vSRX

This example shows how to configure a route-based VPN with a responder behind a NAT device to allow data to be securely transferred between a branch office and the corporate office.

Requirements

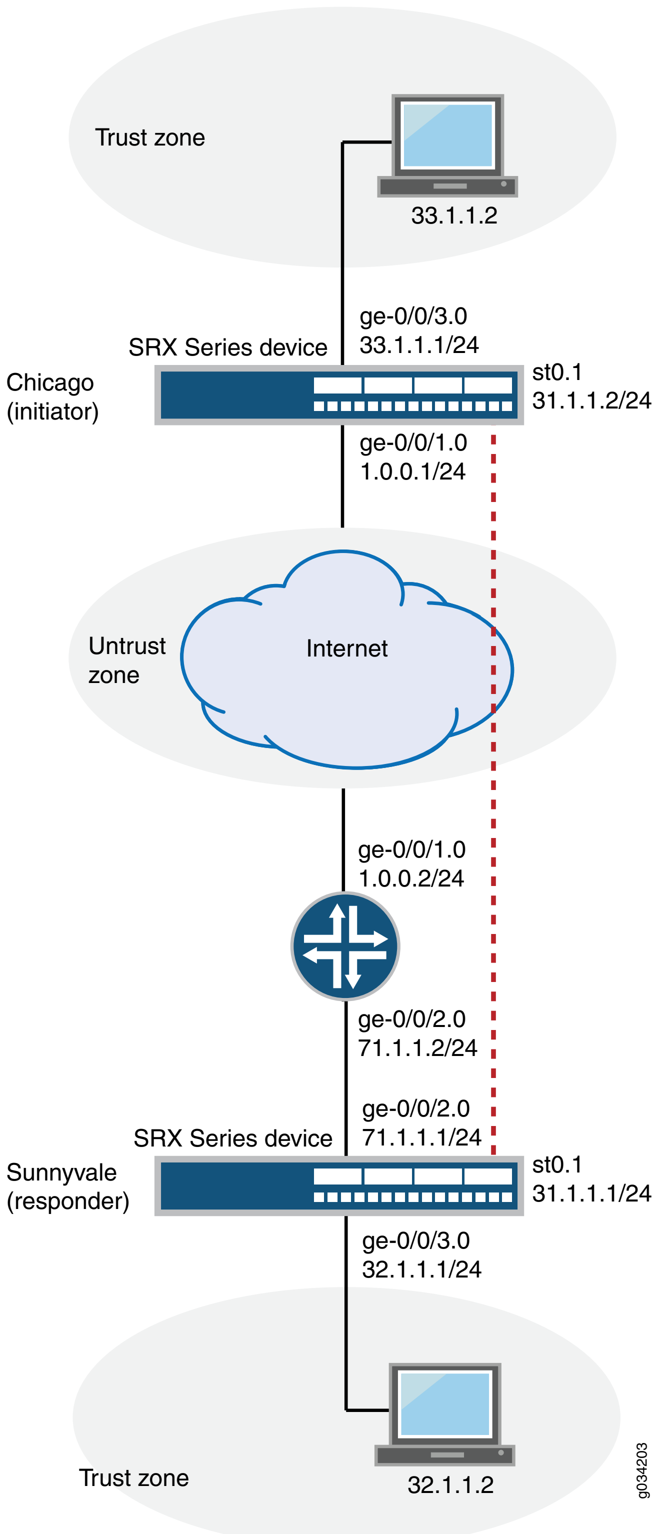
Before you begin, read "IPsec VPN Overview" on page 3.

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.

Figure 24 on page 230 shows an example of a topology for route-based VPN with only the responder behind a NAT device.

Figure 24: Route-Based VPN Topology with Only the Responder Behind a NAT Device



In this example, you configure interfaces, routing options, security zones, and security policies for both an initiator in Chicago and a responder in Sunnyvale. Then you configure IKE Phase 1 and IPsec Phase 2 parameters.

Packets sent from the initiator with a destination address 1.1.1.1/32 are translated to the destination address 71.1.1.1/32 on the NAT device.

See Table 36 on page 231 through Table 38 on page 232 for specific configuration parameters used for the initiator in the examples.

Table 36: Interface, Routing Options, Zones, and Security Policies for the Initiator

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/1	1.0.0.1/24
	ge-0/0/3	33.1.1.1/24
	st0.1 (tunnel interface)	31.1.1.2/24
Static routes	32.1.1.0/24	The next hop is st0.1.
	1.1.1.1/32	The next hop is 1.0.0.2.
Security zones	untrust	• Only IKE system service is allowed. • The ge-0/0/1.0 and the st0.1 interfaces are bound to this zone.
	trust	• All system services are allowed. • All protocols are allowed. • The ge-0/0/3.0 interface is bound to this zone.
Security policies	to-sunnyvale	Permit traffic from 33.1.1.1/24 in the trust zone to 32.1.1.1/24 in the untrust zone.
	from-sunnyvale	Permit traffic from 32.1.1.1/24 in the untrust zone to 33.1.1.1/24 in the trust zone.

Table 37: IKE Phase 1 Configuration Parameters for the Initiator

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: 3des-cbc
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text

Table 37: IKE Phase 1 Configuration Parameters for the Initiator *(continued)*

Feature	Name	Configuration Parameters
Gateway	gw1	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 1.1.1.1 • Local peer (initiator): branch_natt1@example.net • Remote peer (responder): responder_natt1@example.net

Table 38: IPsec Phase 2 Configuration Parameters for the Initiator

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • Perfect forward secrecy (PFS) keys: group2
VPN	vpn1	• IKE gateway reference: gw1 • IPsec policy reference: ipsec_pol • Bind to interface: st0.1 • Establish tunnels immediately

See Table 39 on page 232 through Table 41 on page 233 for specific configuration parameters used for the responder in the examples.

Table 39: Interface, Routing Options, Zones, and Security Policies for the Responder

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/2	71.1.1.1/24
	ge-0/0/3	32.1.1.1/24
	st0.1 (tunnel interface)	31.1.1.1/24
Static routes	0.0.0.0/0 (default route)	The next hop is 71.1.1.2.
	33.1.1.0/24	The next hop is st0.1.
Security zones	untrust	• Only IKE system service is allowed. • The ge-0/0/2.0 and the st0.1 interfaces are bound to this zone.
	trust	• All system services are allowed. All protocols are allowed. • The ge-0/0/3.0 interface is bound to this zone.

Table 39: Interface, Routing Options, Zones, and Security Policies for the Responder *(continued)*

Feature	Name	Configuration Parameters
Security policies	to-chicago	Permit traffic from 32.1.1.1/24 in the trust zone to 33.1.1.1/24 in the untrust zone.
	from-chicago	Permit traffic from 33.1.1.1/24 in the untrust zone to 32.1.1.1/24 in the trust zone.

Table 40: IKE Phase 1 Configuration Parameters for the Responder

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: 3des-cbc
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw1	• IKE policy reference: ike_pol • External interface: ge-0/0/2.0 • Gateway address: 1.0.0.1 • Local peer (responder): responder_natt1@example.net • Remote peer (initiator): branch_natt1@example.net

Table 41: IPsec Phase 2 Configuration Parameters for the Responder

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • PFS keys: group2
VPN	vpn1	• IKE gateway reference: gw1 • IPsec policy reference: ipsec_pol • Bind to interface: st0.1 • Establish tunnels immediately

Configuration

- Configuring Interface, Routing Options, Security Zones, and Security Policies for the Initiator on page 234
- Configuring IKE for the Initiator on page 238

Configuring Interface, Routing Options, Security Zones, and 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 interfaces ge-0/0/1 unit 0 family inet address 1.0.0.1/24
set interfaces ge-0/0/3 unit 0 family inet address 33.1.1.1/24
set interfaces st0 unit 1 family inet address 31.1.1.2/24
set routing-options static route 32.1.1.0/24 next-hop st0.1
set routing-options static route 1.1.1.1/32 next-hop 1.0.0.2
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces st0.1
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 address-book book1 address Chicago-lan 33.1.1.1/24
set security address-book book1 attach zone trust
set security address-book book2 address Sunnyvale-lan 32.1.1.1/24
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy to-sunnyvale match
  source-address Chicago-lan
set security policies from-zone trust to-zone untrust policy to-sunnyvale match
  destination-address Sunnyvale-lan
set security policies from-zone trust to-zone untrust policy to-sunnyvale match application
  any
set security policies from-zone trust to-zone untrust policy to-sunnyvale then permit
set security policies from-zone untrust to-zone trust policy from-sunnyvale match
  source-address Sunnyvale-lan
set security policies from-zone untrust to-zone trust policy from-sunnyvale match
  destination-address Chicago-lan
set security policies from-zone untrust to-zone trust policy from-sunnyvale match
  application any
set security policies from-zone untrust to-zone trust policy from-sunnyvale 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 interface, static route, security zone, and security policy information:

1. Configure Ethernet interface information.

   ```
   [edit]
   ```


user@host# **set interfaces ge-0/0/1 unit 0 family inet address 1.0.0.1/24**
user@host# **set interfaces ge-0/0/3 unit 0 family inet address 33.1.1.1/24**
user@host# **set interfaces st0 unit 1 family inet address 31.1.1.2/24**

2. Configure static route information.

   [edit]
   user@host# **set routing-options static route 32.1.1.0/24 next-hop st0.1**
   user@host# **set routing-options static route 1.1.1.1/32 next-hop 1.0.0.2**

3. Configure the untrust security zone.

   [edit]
   user@host# **set security zones security-zone untrust**

4. Assign interfaces to the untrust security zone.

   [edit security zones security-zone untrust]
   user@host# **set interfaces ge-0/0/1.0**
   user@host# **set interfaces st0.1**

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# **set security zones security-zone trust host-inbound-traffic protocols all**

7. Assign an interface to the trust security zone.

   [edit security zones security-zone trust]
   user@host# **set interfaces ge-0/0/3.0**

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 address books.

   [edit security address-book]
   user@host# **set book1 address Chicago-lan 33.1.1.1/24**
   user@host# **set book1 attach zone trust**
   user@host# **set book2 address Sunnyvale-lan 32.1.1.1/24**
   user@host# **set book2 attach zone untrust**

10. Create security policies.

    [edit security security-policies from-zone trust to-zone untrust]
    user@host# **set policy to-sunnyvale match source-address Chicago-lan**

```
user@host# set policy to-sunnyvale match destination-address Sunnyvale-lan
user@host# set policy to-sunnyvale match application any
user@host# set policy to-sunnyvale then permit

[edit security security-policies from-zone untrust to-zone trust]
user@host# set policy from-sunnyvale match source-address Sunnyvale-lan
user@host# set policy from-sunnyvale match destination-address Chicago-lan
user@host# set policy from-sunnyvale match application any
user@host# set policy from-sunnyvale then permit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, **show security zones**, **show security address-book**, 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 1.0.0.1/24;
      }
   }
}
ge-0/0/3 {
   unit 0 {
      family inet {
         address 33.1.1.1/24;
      }
   }
}
st0 {
   unit 1 {
      family inet {
         address 31.1.1.2/24
      }
   }
}

[edit]
user@host# show routing-options
static {
   route 32.1.1.0/24 next-hop st0.1;
   route 1.1.1.1/32 next-hop 1.0.0.2;
}

[edit]
user@host# show security zones
security-zone untrust {
   host-inbound-traffic {
      system-services {
         ike;
      }
   }
```



```
  interfaces {
    st0.1;
    ge-0/0/1.0;
  }
}
security-zone trust {
  host-inbound-traffic {
    system-services {
      all;
    }
    protocols {
    all;
    }
  }
  interfaces {
    ge-0/0/3.0;
    }
  [edit]
[edit]
user@host# show security address-book
  book1 {
    address Chicago-lan 33.1.1.1/24;
    attach {
      zone trust;
    }
  }
  book2 {
    address Sunnyvale-lan 32.1.1.1/24;
    attach {
      zone untrust;
    }
  }
[edit]
user@host# show security policies
  from-zone trust to-zone untrust {
    policy to-sunnyvale {
      match {
        source-address Chicago-lan;
        destination-address Sunnyvale-lan;
        application any;
      }
      then {
        permit;
      }
    }
  }
  from-zone untrust to-zone trust {
    policy from-sunnyvale {
      match {
        source-address Sunnyvale-lan;
        destination-address Chicago-lan;
        application any;
      }
      then {
        permit;
      }
```

```
    }
  }
```

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 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 1.1.1.1
set security ike gateway gw1 local-identity user-at-hostname branch_natt1@example.net
set security ike gateway gw1 remote-identity user-at-hostname
  responder_natt1@example.net
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 the IKE Phase 1 proposal.

   ```
   [edit security ike]
   user@host# set 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 sha1
   ```

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]
user@host# set policy ike_pol
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike_pol]
user@host# set mode main
```

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]
user@host# set pre-shared-key ascii-text “$ABC123”
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike gateway gw1]
user@host# set external-interface ge-0/0/1.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gw1]
user@host# set ike-policy ike_pol
```

12. Define the IKE Phase 1 gateway address.

```
[edit security ike gateway gw1]
user@host# set address 1.1.1.1
```

13. Set **local-identity** of the local peer.

```
[edit security ike gateway gw1]
user@host# set local-identity user-at-hostname branch_natt1@example.net
```

14. Set **remote-identity** of the responder. This is the IKE identifier.

```
[edit security ike gateway gw1]
user@host# set remote-identity user-at-hostname responder_natt1@example.net
```

15. Define the external interface.

```
[edit security ike gateway gw1]
user@host# set 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@host# 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 "$ABC123";
}
gateway gw1 {
   ike-policy ike_poly;
   address 1.1.1.1;
   local-identity user-at-hostname branch_natt1@example.net;
   remote-identity user-at-hostname responder_natt1@example.net;
   external-interface ge-0/0/1.0;
}
```

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-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 st0.1
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@host# set 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-sha1-96**

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# **set policy ipsec_pol**

6. Specify IPsec Phase 2 to use perfect forward secrecy (PFS).

   [edit security ipsec policy ipsec_pol]
   user@host# **set perfect-forward-secrecy keys group2**

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 vpn1 ike gateway gw1**

9. Specify the IPsec Phase 2 policy.

   [edit security ipsec]
   user@host# **set vpn vpn1 ike ipsec-policy ipsec_pol**

10. Specify the interface to bind.

    [edit security ipsec]
    user@host# **set vpn vpn1 bind-interface st0.1**

11. Specify that the tunnel be brought up immediately without waiting for a verification packet to be sent.

    [edit security ipsec]
    user@host# **set 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@host# 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 st0.1;
    ike {
      gateway gw1;
      ipsec-policy ipsec_pol;
      }
    establish-tunnels immediately;
  }
  proposals ipsec_prop;
  }
}
```

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

Configuring Interfaces, Routing Options, Security Zones, and 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 interfaces ge-0/0/2 unit 0 family inet address 71.1.1.1/24
set interfaces ge-0/0/3 unit 0 family inet address 32.1.1.1/24
set interfaces st0 unit 1 family inet address 31.1.1.1/24
set routing-options static route 0.0.0.0/0 next-hop 71.1.1.2
set routing-options static route 33.1.1.0/24 next-hop st0.1
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone 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 address-book book1 address Sunnyvale-lan 32.1.1.1/24
set security address-book book1 attach zone trust
set security address-book book2 address Chicago-lan 33.1.1.1/24
set security address-book book2 attach zone untrust
```



```
set security policies from-zone trust to-zone untrust policy to-chicago match
   source-address Sunnyvale-lan
set security policies from-zone trust to-zone untrust policy to-chicago match
   destination-address Chicago-lan
set security policies from-zone trust to-zone untrust policy to-chicago match application
   any
set security policies from-zone trust to-zone untrust policy to-chicago then permit
set security policies from-zone untrust to-zone trust policy from-chicago match
   source-address Chicago-lan
set security policies from-zone untrust to-zone trust policy from-chicago match
   destination-address Sunnyvale-lan
set security policies from-zone untrust to-zone trust policy from-chicago match application
   any
set security policies from-zone untrust to-zone trust policy from-chicago 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 interface, static route, security zones, policies and gateways:

1. Configure Ethernet interface information.

   ```
   [edit]
   user@host# set interfaces ge-0/0/2 unit 0 family inet address 71.1.1.1/24
   user@host# set interfaces ge-0/0/3 unit 0 family inet address 32.1.1.1/24
   user@host# set interfaces st0 unit 1 family inet address 31.1.1.1/24
   ```

2. Configure static route information.

   ```
   [edit]
   user@host# set routing-options static route 0.0.0.0/0 next-hop 71.1.1.2
   user@host# set routing-options static route 33.1.1.0/24 next-hop st0.1
   ```

3. Configure the untrust security zone.

   ```
   [edit ]
   user@host# set security zones security-zone untrust
   ```

4. Assign interfaces to the untrust security zone.

   ```
   [edit security zones security-zone untrust]
   user@host# set security zones security-zone untrust interfaces ge-0/0/2.0
   user@host# set security zones security-zone untrust interfaces st0.1
   ```

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# **set security zones security-zone trust host-inbound-traffic protocols all**

7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/3.0
```

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 address books.

```
[edit security address-book]
user@host# set book1 address Sunnyvale-lan 32.1.1.1/24
user@host# set book1 attach zone trust
user@host# set book2 address Chicago-lan 33.1.1.1/24
user@host# set book2 attach zone untrust
```

10. Create security policies.

```
[edit security security-policies from-zone trust to-zone untrust]
user@host# set policy to-chicago match source-address Sunnyvale-lan
user@host# set policy to-chicago match destination-address Chicago-lan
user@host# set policy to-chicago match application any
user@host# set policy to-chicago then permit

[edit security security-policies from-zone untrust to-zone trust]
user@host# set policy from-chicago match source-address Chicago-lan
user@host# set policy from-chicago match destination-address Sunnyvale-lan
user@host# set policy from-chicago match application any
user@host# set policy from-chicago then permit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, **show security zones**, **show security address-book**, 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 71.1.1.1/24;
      }
   }
}
ge-0/0/3 {
   unit 0 {
      family inet {
```



```
            address 32.1.1.1/24;
        }
    }
}
st0 {
    unit 1 {
        family inet {
            address 31.1.1.1/24
        }
    }
}
```

```
[edit]
user@host# show routing-options
static {
    route 0.0.0.0/0 next-hop 71.1.1.2;
    route 33.1.1.0/24 next-hop st0.1;
}
```

```
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        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 address-book
    book1 {
        address Sunnyvale-lan 32.1.1.1/24;
        attach {
            zone trust;
        }
    }
    book2 {
        address Chicago-lan 33.1.1.1/24;
        attach {
            zone untrust;
```

```
    }
  }
[edit]
user@host# show security policies
  from-zone trust to-zone untrust {
    policy to-chicago {
      match {
        source-address Sunnyvale-lan;
        destination-address Chicago-lan;
        application any;
      }
      then {
        permit;
      }
    }
  }
  from-zone untrust to-zone trust {
    policy from-chicago {
      match {
        source-address Chicago-lan;
        destination-address Sunnyvale-lan;
        application any;
      }
      then {
        permit;
      }
    }
  }
```

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 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 1.0.0.1
set security ike gateway gw1 local-identity user-at-hostname
  responder_natt1@example.net
set security ike gateway gw1 remote-identity user-at-hostname branch_natt1@example.net
set security ike gateway gw1 external-interface ge-0/0/2.0
```


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 the IKE Phase 1 proposal.

   ```
   [edit security ike]
   user@host# set 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 sha1
   ```

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]
   user@host# set policy ike_pol
   ```

7. Set the IKE Phase 1 policy mode.

   ```
   [edit security ike policy ike_pol]
   user@host# set mode main
   ```

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]
   user@host# set pre-shared-key ascii-text “$ABC123”
   ```

10. Create an IKE Phase 1 gateway and define its external interface.

   ```
   [edit security ike gateway gw1]
   ```

user@host# **set external-interface ge-0/0/2.0**

11. Define the IKE Phase 1 policy reference.

    [edit security ike gateway gw1]
    user@host# **set ike-policy ike_pol**

12. Define the IKE Phase 1 gateway address.

    [edit security ike gateway gw1]
    user@host# **set address 1.0.0.1**

13. Set **local-identity** of the responder.

    [edit security ike gateway gw1]
    user@host# **set local-identity user-at-hostname responder_natt1@example.net**

14. Set **remote-identity** of the responder. This is the IKE identifier.

    [edit security ike gateway gw1]
    user@host# **set remote-identity user-at-hostname branch_natt1@example.net**

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 sha1;
  encryption-algorithm 3des-cbc;
}
policy ike_pol {
  mode main;
  proposals ike_prop;
  pre-shared-key ascii-text “$ABC123”;
}
gateway gw1 {
  ike-policy ike_pol;
  address 1.0.0.1;
  local-identity user-at-hostname "responder_natt1@example.net";
  remote-identity user-at-hostname "branch_natt1@example.net";
  external-interface ge-0/0/2.0;
}
```

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-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 st0.1
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@host# set 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-sha1-96
   ```

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# set policy ipsec_pol
   ```

6. Specify IPsec Phase 2 to use perfect forward secrecy (PFS).

   ```
   [edit security ipsec policy ipsec_pol]
   user@host# set perfect-forward-secrecy keys group2
   ```


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 security ipsec vpn vpn1 ike gateway gw1
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn vpn1 ike ipsec-policy ipsec_pol
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn vpn1 bind-interface st0.1
```

11. Specify that the tunnel be brought up immediately without waiting for a verification packet to be sent.

```
[edit security ipsec]
user@host# set 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@host# 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 st0.1;
  ike {
    gateway gw1;
    ipsec-policy ipsec_pol;
    }
  establish-tunnels immediately;
}
```

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

Verification

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

NOTE: Before starting the verification process, you must send traffic from a host in the 33.1.1.0 network to a host in the 32.1.1.0 network. For route-based VPNs, traffic can be initiated by the SRX Series device 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 33.1.1.2 to 32.1.1.2.

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

106321  UP     d31d6833108fd69f  9ddfe2ce133086aa  Main          1.1.1.1

user@host> show security ike security-associations index 1 detail
IKE peer 1.1.1.1, Index
  Initiator cookie: d31d6833108fd69f, Responder cookie: 9ddfe2ce133086aa
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 1.0.0.1:500, Remote: 1.1.1.1:500
  Lifetime: Expires in 28785 seconds
  Peer ike-id: responder_natt1@example.net
  Xauth assigned IP: responder_natt1@example.net
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                  0
   Output bytes  :                  0
   Input  packets:                  0
   Output packets:                  0
  Flags: IKE SA is created
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 0
```



```
Negotiation type: Quick mode, Role: Initiator, Message ID: 0
Local: 1.0.0.1:500, Remote: 1.1.1.1:500
Local identity: branch_natt1@example.net
Remote identity: responder_natt1@example.net
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 500 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 500.
  - 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

NOTE: 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
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <131073 ESP:3des/sha1 ac23df79 2532/ unlim   -   root 500  1.1.1.1
  >131073 ESP:3des/sha1 cbc9281a 2532/ unlim   -   root 500  1.1.1.1
```

```
user@host> show security ipsec security-associations detail
Virtual-system: root
  Local Gateway: 1.0.0.1, Remote Gateway: 1.1.1.1
  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: IKEv1
    DF-bit: clear
    Direction: inbound, SPI: ac23df79, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3186 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2578 seconds
    Mode: Tunnel, 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: cbc9281a, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3186 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2578 seconds
    Mode: Tunnel, 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 a NAT address of 1.1.1.1.
- Both peers in the IPsec SA pair are using port 500.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 2532/ unlim value indicates that the Phase 2 lifetime expires in 2532

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

5802591 UP     d31d6833108fd69f  9ddfe2ce133086aa  Main           1.0.0.1


user@host> show security ike security-associations index 1 detail
IKE peer 1.0.0.1, Index 5802591,
  Role: Responder, State: UP
  Initiator cookie: d31d6833108fd69f, Responder cookie: 9ddfe2ce133086aa
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 71.1.1.1:500, Remote: 1.0.0.1:500
  Lifetime: Expires in 25704 seconds
  Peer ike-id: branch_natt1@example.net
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Flags: IKE SA is created
  IPSec security associations: 8 created, 2 deleted
  Phase 2 negotiations in progress: 0

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 71.1.1.1:500, Remote: 1.0.0.1:500
    Local identity: responder_natt1@example.net
    Remote identity: branch_natt1@example.net
    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 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 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

NOTE: **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
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <131073 ESP:3des/sha1 a5224cd9 3571/ unlim   -   root 500  1.0.0.1
  >131073 ESP:3des/sha1 82a86a07 3571/ unlim   -   root 500  1.0.0.1
```

```
user@host> show security ipsec security-associations detail
Virtual-system: root
  Local Gateway: 71.1.1.1, Remote Gateway: 1.0.0.1
  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: IKEv1
    DF-bit: clear
    Direction: inbound, SPI: a5224cd9, AUX-SPI: 0
                          , VPN Monitoring: -
    Hard lifetime: Expires in 3523 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2923 seconds
    Mode: Tunnel, 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: 82a86a07, AUX-SPI: 0
                          , VPN Monitoring: -
    Hard lifetime: Expires in 3523 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2923 seconds
    Mode: Tunnel, 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 1.0.0.1.
- Both peers in the IPsec SA pair are using port 500.
- 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.

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.

Related Documentation

- IPsec VPN Overview on page 3
- Understanding NAT-T on page 227
- Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device on page 257

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

Supported Platforms SRX Series, vSRX

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 VPN Overview" on page 3.

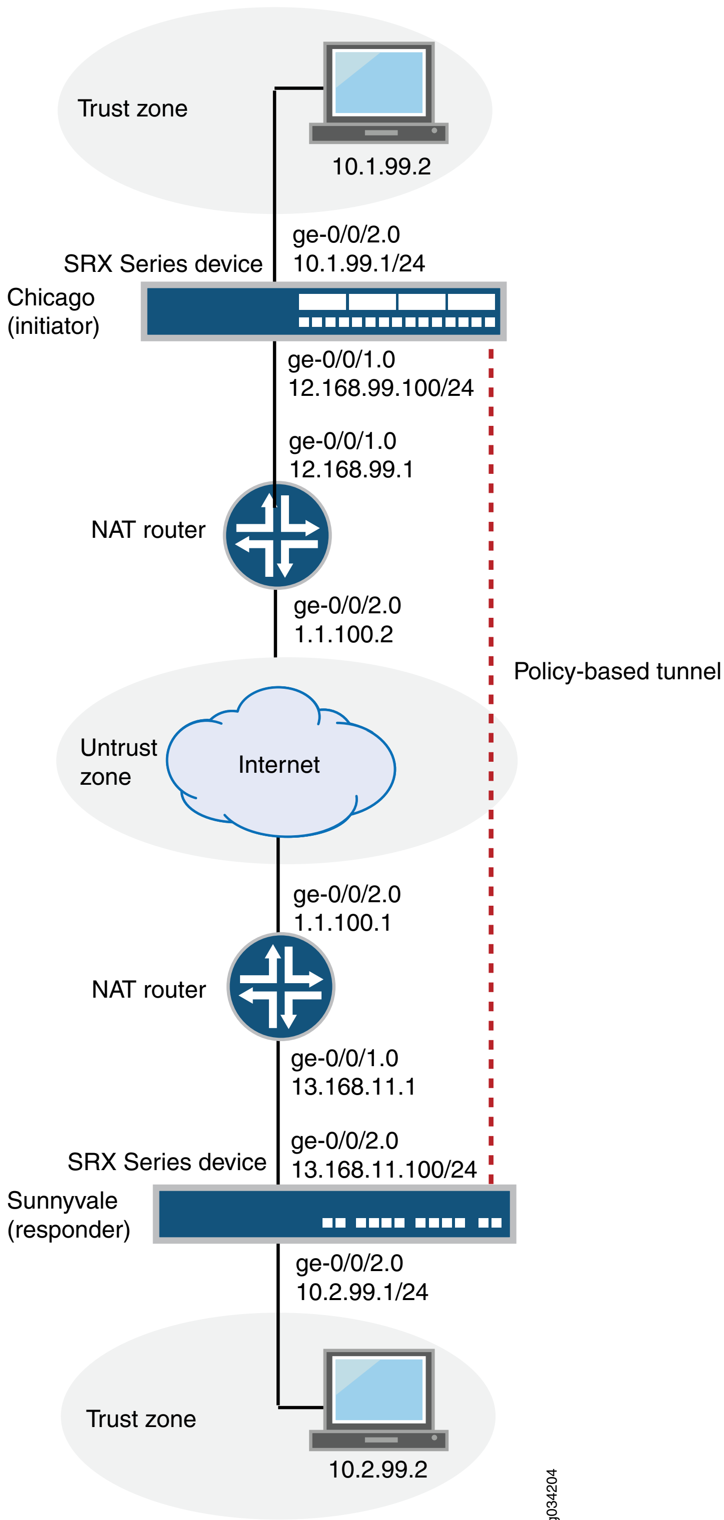
Overview

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 25 on page 259 shows an example of a topology for a VPN with both an initiator and a responder behind a NAT device.

Figure 25: 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 NAT device and mapped to public IP address 1.1.100.1.

See Table 42 on page 260 through Table 45 on page 261 for specific configuration parameters used for the initiator in the examples.

Table 42: Interface, Routing Options, and Security Zones for the Initiator

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/1	12.168.99.100/24
	ge-0/0/2	10.1.99.1/24
Static routes	10.2.99.0/24 (default route)	The next hop is 12.168.99.1.
	13.168.11.0/24	The next hop is 12.168.99.1.
	1.1.100.0/24	12.168.99.1
Security zones	trust	• All system services are allowed. • All protocols are allowed. • The ge-0/0/2.0 interface is bound to this zone.
	untrust	• All system services are allowed. • All protocols are allowed. • The ge-0/0/1.0 interface is bound to this zone.

Table 43: IKE Phase 1 Configuration Parameters for the Initiator

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: md5 • Encryption algorithm: 3des-cbc
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gate	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 1.1.100.23 • Local peer is hostname chicago • Remote peer is hostname sunnyvale

Table 44: IPsec Phase 2 Configuration Parameters for the Initiator

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-md5-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • Perfect forward secrecy (PFS): group1
VPN	first_vpn	• IKE gateway reference: gate • IPsec policy reference: ipsec_pol

Table 45: 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:
		  • source-address any
		  • destination-address any
		  • application any
		• Action: permit tunnel ipsec-vpn first_vpn

The security policy permits tunnel traffic from the untrust zone to the trust zone.	pol1	• Match criteria:
		  • source-address any
		  • destination-address any
		  • application any
		• Action: permit tunnel ipsec-vpn first_vpn

See Table 46 on page 261 through Table 49 on page 263 for specific configuration parameters used for the responder in the examples.

Table 46: Interface, Routing Options, and Security Zones for the Responder

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/2	13.168.11.100/24
	ge-0/0/3	10.2.99.1/24
Static routes	10.1.99.0/24 (default route)	The next hop is 13.168.11.1.
	12.168.99.0/24	The next hop is 13.168.11.1.
	1.1.100.0/24	13.168.11.1

Table 46: Interface, Routing Options, and Security Zones for the Responder *(continued)*

Feature	Name	Configuration Parameters
Security zones	trust	• All system services are allowed. • All protocols are allowed. • The ge-0/0/3.0 interface is bound to this zone.
	untrust	• All system services are allowed. • All protocols are allowed. • The ge-0/0/2.0 interface is bound to this zone.

Table 47: IKE Phase 1 Configuration Parameters for the Responder

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: md5 • Encryption algorithm: 3des-cbc
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gate	• IKE policy reference: ike_pol • External interface: ge-0/0/2.0 • Gateway address: 1.1.100.22 • Always send dead-peer detection • Local peer is hostname sunnyvale • Remote peer is hostname chicago

Table 48: IPsec Phase 2 Configuration Parameters for the Responder

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-md5-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • Perfect forward secrecy (PFS): group1
VPN	first_vpn	• IKE gateway reference: gate • IPsec policy reference: ipsec_pol • Establish tunnels immediately

Table 49: 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: ◦ source-address any ◦ destination-address any ◦ application any • Action: permit tunnel ipsec-vpn first_vpn
The security policy permits tunnel traffic from the untrust zone to the trust zone.	pol1	• Match criteria: ◦ source-address any ◦ destination-address any ◦ application any • Action: permit tunnel ipsec-vpn first_vpn

Configuration

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-0/0/1 unit 0 family inet address 12.168.99.100/24
set interfaces ge-0/0/2 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 13.168.11.0/24 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-0/0/2.0
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
```

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-0/0/1 unit 0 family inet address 12.168.99.100/24
   user@host# set interfaces ge-0/0/2 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 13.168.11.0/24 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-0/0/2.0
   ```

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. Configure the untrust security zone.

   ```
   [edit security zones security-zone untrust]
   user@host# set host-inbound-traffic protocols all
   ```

7. Assign an interface to the untrust security zone.

   ```
   [edit security zones security-zone untrust]
   user@host# set interfaces ge-0/0/1.0
   ```

8. Specify system services for the untrust security zone.

   ```
   [edit security zones security-zone untrust]
   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-0/0/1 {
   unit 0 {
      family inet {
         address 12.168.99.100/24;
      }
   }
}
ge-0/0/2 {
   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 13.168.11.0/24 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 untrust {
   host-inbound-traffic {
      system-services {
         all;
      }
      protocols{
      all;
      }
   }
   interfaces {
      ge-0/0/1.0.;
   }
}
security-zone trust {
   host-inbound-traffic {
      system-services {
         all;
      }
      protocols {
      all;
      }
   }
   interfaces {
      ge-0/0/2.0;
   }
```

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 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 gate ike-policy ike_pol
set security ike gateway gate address 1.1.100.23
set security ike gateway gate external-interface ge-0/0/1.0
set security ike gateway gate local-identity hostname chicago
set security ike gateway gate remote-identity hostname sunnyvale
```

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_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# **set policy ike_pol**

7. Set the IKE Phase 1 policy mode.

   [edit security ike policy ike_pol]
   user@host# **set mode main**

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-0/0/1.0**

11. Create an IKE Phase 1 gateway address.

    [edit security ike gateway]
    set gate address 1.1.100.23

12. Define the IKE Phase 1 policy reference.

    [edit security ike gateway]
    set gate ike-policy ike_pol

13. Set **local-identity** for the local peer.

    [edit security ike gateway gate]
    user@host# **set local-identity hostname chicago**

14. Set **remote-identity** for the responder. This is the responder's local identity.

    [edit security ike gateway gate]
    user@host# **set remote-identity hostname sunnyvale**

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 main;
    proposals ike_prop;
    pre-shared-key ascii-text "$ABC123";
  }
  gateway gate {
    ike-policy ike_pol;
    address 1.1.100.23;
    local-identity hostname chicago;
    remote-identity hostname sunnyvale;
    external-interface ge-0/0/1.0;
  }
```

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
```

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_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;
    }
  }
}
```

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 source-address
  any
set security policies from-zone untrust to-zone trust policy pol1 match destination-address
  any
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 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
```

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 {
          source-address any;
          destination-address any;
          application any;
        }
        then {
          permit;
            tunnel {
          ipsec-vpn first_vpn;
        }
      }
    }
```

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-0/0/2 unit 0 family inet address 13.168.11.100/24
set interfaces ge-0/0/3 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 12.168.99.0/24 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 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 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
```

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-0/0/2 unit 0 family inet address 13.168.11.100/24**
   user@host# **set interfaces ge-0/0/3 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 12.168.99.0/24 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. Configure the untrust security zone.

   [edit]
   user@host# **set security zones security-zone untrust host-inbound-traffic protocols all**

4. Assign an interface to the untrust security zone.

   [edit security zones security-zone untrust]
   user@host# **set interfaces ge-0/0/2.0**

5. Specify allowed system services for the untrust security zone.

   [edit security zones security-zone untrust]
   user@host# **set host-inbound-traffic system-services all**

6. Configure the trust security zone.

   [edit]
   user@host# **set security zones security-zone trust host-inbound-traffic protocols all**

7. Assign an interface to the trust security zone.

   [edit security zones security-zone trust]
   user@host# **set interfaces ge-0/0/3.0**

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**

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-0/0/2 {
  unit 0 {
    family inet {
      address 13.168.11.100/24;
    }
  }
}
ge-0/0/3 {
  unit 0 {
    family inet {
      address 10.2.99.1/244;
    }
  }
}
```

```
[edit]
user@host# show routing-options
static {
  route 10.1.99.0/24 next-hop 13.168.11.1;
  route 12.168.99.0/24 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 {
    system-services {
      all;
      protocols {
      all;
    }
  }
  interfaces {
    ge-0/0/2.0;
  }
}
security-zone trust {
  host-inbound-traffic {
    system-services {
      all;
    }
    protocols {
    all;
    }
  }
  interfaces {
    ge-0/0/3.0;
  }
```

```
}
```

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 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 gate ike-policy ike_pol
set security ike gateway gate address 1.1.100.22
set security ike gateway gate dead-peer-detection probe-idle-tunnel
set security ike gateway gate external-interface ge-0/0/2.0
set security ike gateway gate local-identity hostname sunnyvale
set security ike gateway gate remote-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# set proposal ike-phase1-proposal
   ```

2. Define the IKE proposal authentication method.

   ```
   [edit security ike proposal ike_prop]
   user@host# set authentication-method pre-shared-key
   ```

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]
   user@host# **set policy ike_pol**

7. Set the IKE Phase 1 policy mode.

   [edit security ike policy ike_pol]
   user@host# **set mode main**

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 proposals ike_prop set 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 security ike gateway gate external-interface ge-0/0/2.0**

11. Define the IKE Phase 1 policy reference.

    [edit security ike gateway]
    user@host# **set gate ike-policy ike_pol**

12. Create an IKE Phase 1 gateway address.

    [edit security ike gateway]
    user@host# **set gate address 1.1.100.22**

13. Set **local-identity** for the local peer (initiator).

    [edit security ike gateway gate]
    user@host# **set local-identity hostname sunnyvale**

14. Set **remote-identity** for the responder. This is the responder's local identity.

    [edit security ike gateway gate]
    user@host# **set remote-identity hostname chicago**

15. Set dead peer detection to detect whether the peer is up or down.

    [edit security ike gateway gate]
    user@host# **set dead-peer-detection probe-idle-tunnel**

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 main;
   proposals ike_prop;
   pre-shared-key ascii-text "$ABC123";
}
gateway gate {
   ike-policy ike_pol;
   address 1.1.100.22;
   dead-peer-detection probe-idle-tunnel;
   external-interface ge-0/0/2.0;
   local-identity hostname sunnyvale;
   remote-identity hostname chicago;
}
```

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
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# set 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. 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**

6. Create the IPsec Phase 2 policy.

   [edit security ipsec]
   user@host# **set policy ipsec_pol**

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**

10. Specify that the tunnel be brought up immediately without a verification packet.

   [edit security ipsec]
   user@host# **set security ipsec vpn first_vpn 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@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 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 source-address any
set security policies from-zone untrust to-zone trust policy pol1 match destination-address any
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 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
```

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 {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
        tunnel {
      ipsec-vpn first_vpn;
    }
  }
}
```

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

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status for the Initiator on page 280
- Verifying IPsec Security Associations for the Initiator on page 282

- Verifying the IKE Phase 1 Status for the Responder on page 283
- Verifying IPsec Security Associations for the Responder on page 285

Verifying the IKE Phase 1 Status for the Initiator

Purpose Verify the IKE Phase 1 status.

Action

NOTE: 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 device 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

5137403 UP     b3a24bc00e963c51  7bf96bcc6230e484  Main           1.1.100.23

user@host> show security ike security-associations index 1 detail
Index    State  Initiator cookie  Responder cookie  Mode           Remote Address

1400579286 UP  487cfb570908425c  7710c8487f9ff20c  Main           1.1.100.22


{primary:node0}[edit]
root@poway# run show security ike security-associations detail
node0:

IKE peer 1.1.100.22, Index 1400579286,
  Location: FPC 5, PIC 0, KMD-Instance 4
  Role: Initiator, State: UP
  Initiator cookie: 487cfb570908425c, Responder cookie: 7710c8487f9ff20c
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 13.168.11.100:4500, Remote: 1.1.100.22:4500
  Lifetime: Expires in 28622 seconds
  Peer ike-id: sunnyvale
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-md5-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-md5
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
```



```
 Output packets:                    0
IPSec security associations: 0 created, 0 deleted
Phase 2 negotiations in progress: 0
```

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

NOTE: 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
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <2    ESP:3des/md5    2bf24122 3390/ unlim   -   root 4500  1.1.100.23
  >2    ESP:3des/md5    2baef146 3390/ unlim   -   root 4500  1.1.100.23
```

```
user@host> show security ipsec security-associations detail
Local Gateway: 12.168.99.100, 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
    Policy-name: pol1

    Location: FPC 5, PIC 0, KMD-Instance 4
    Direction: inbound, SPI: 2bf24122, AUX-SPI: 0
                          , VPN Monitoring: -
    Hard lifetime: Expires in 3388 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2801 seconds
    Mode: Tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-md5-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Location: FPC 5, PIC 0, KMD-Instance 4
    Direction: outbound, SPI: 2baef146, AUX-SPI: 0
                          , VPN Monitoring: -
    Hard lifetime: Expires in 3388 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2801 seconds
    Mode: Tunnel, 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 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

5802591 UP     d31d6833108fd69f  9ddfe2ce133086aa  Main           1.0.0.1


user@host> show security ike security-associations index 1 detail
IKE peer 1.1.100.23, Index 1400579287,
  Location: FPC 5, PIC 0, KMD-Instance 4
  Role: Responder, State: UP
  Initiator cookie: 487cfb570908425c, Responder cookie: 7710c8487f9ff20c
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 12.168.99.100:4500, Remote: 1.1.100.23:4500
  Lifetime: Expires in 28587 seconds
  Peer ike-id: chicago
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-md5-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-md5
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 0

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 71.1.1.1:4500, Remote: 1.0.0.1:4500
    Local identity: branch_natt1@example.net
```

```
Remote identity: limits_natt1@example.net
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

NOTE: 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
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <131073 ESP:3des/sha1 a5224cd9 3571/ unlim   -   root 4500  1.0.0.1
  >131073 ESP:3des/sha1 82a86a07 3571/ unlim   -   root 4500  1.0.0.1
```

```
user@host> show security ipsec security-associations detail
Virtual-system: root
  Local Gateway: 71.1.1.1, Remote Gateway: 1.0.0.1
  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: IKEv1
    DF-bit: clear
    Direction: inbound, SPI: a5224cd9, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3523 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2923 seconds
    Mode: Tunnel, 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: 82a86a07, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3523 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2923 seconds
    Mode: Tunnel, 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 a NAT address of 1.0.0.1.
- 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.

Related Documentation

- IPsec VPN Overview on page 3
- Understanding NAT-T on page 227
- Example: Configuring a Route-Based VPN with Only the Responder Behind a NAT Device on page 228

Example: Configuring NAT-T with Dynamic Endpoint VPN

Supported Platforms SRX Series, vSRX

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:

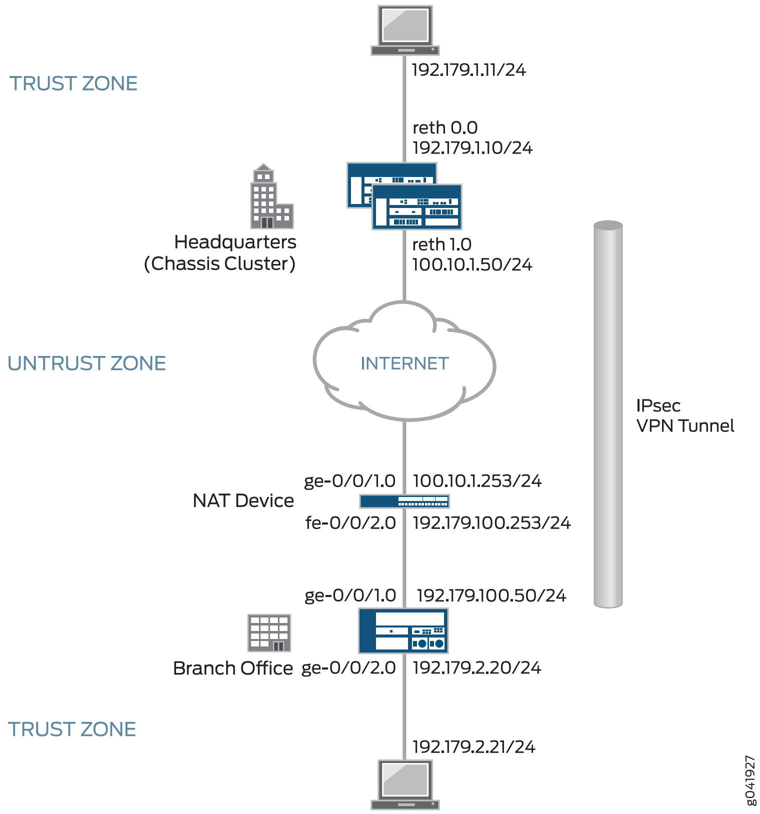
- Two SRX Series devices configured in a chassis cluster
- One SRX Series device providing NAT
- One SRX Series device providing branch office network access
- Junos OS Release 12.1X46-D10 or later for IKEv2 NAT-T support

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 26 on page 287 shows an example of a topology with NAT-Traversal (NAT-T) and dynamic endpoint VPN.

Figure 26: 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.

NOTE: 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*.

NOTE: Starting with Junos OS Release 12.1X46-D10, 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.

NOTE: 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

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 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 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-0/0/2.0
set security nat source rule-set DYNAMIC from zone untrust
set security nat source rule-set DYNAMIC to zone trust
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 trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0

[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 fe-0/0/2.0
```

3. Configure NAT.

```
[edit security nat source rule-set DYNAMIC]
user@host# set from zone untrust
user@host# set to zone trust
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

Confirm that the configuration is working properly.

- Verifying the IKE Phase 1 Status for the Responder on page 300
- Verifying IPsec Security Associations for the Responder on page 301

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

NOTE: **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.

Release History Table

Release	Description
12.1X46-D10	Starting with Junos OS Release 12.1X46-D10, 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

- IPsec VPN Overview on page 3
- *Security Policies Overview*
- Understanding NAT-T on page 227
- Example: Configuring a Route-Based VPN with Only the Responder Behind a NAT Device on page 228
- Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device on page 257

PART 5

Configuring IPsec VPN Tunnels with Chassis Clusters

CHAPTER 12

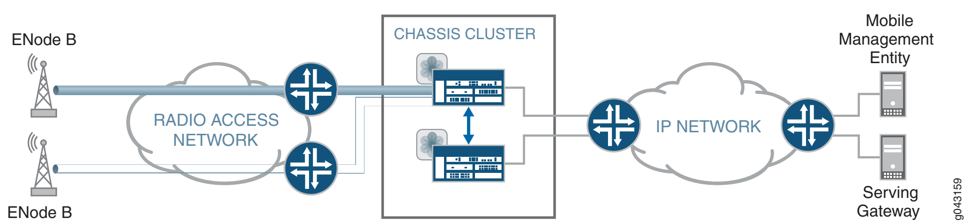
Configuring IPsec VPN Tunnels with Chassis Clusters

Understanding Dual Active-Backup IPsec VPN Chassis Clusters

Supported Platforms SRX Series, vSRX

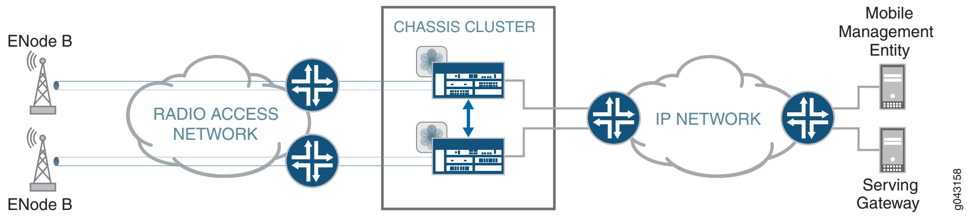
In an active/passive chassis cluster, all VPN tunnels terminate on the same node, as shown in Figure 27 on page 307.

Figure 27: 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 28 on page 308. This deployment is known as *dual active-backup IPsec VPN chassis clusters* This feature is supported only on SRX1500, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.

Figure 28: 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.
- IKEv2 only. IKEv1 is not supported.
- 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 "Understanding VPN Support for Inserting Services Processing Cards" on page 34.
- 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 are only supported on SRX5400, SRX5600, and SRX5800 chassis clusters. Dual active-backup IPsec VPN chassis clusters cannot be configured with the following features:

- VPNs with manual or preshared keys.
- 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.

Related Documentation

- *Chassis Cluster Feature Guide for SRX Series Devices*

Understanding Loopback Interface for a High Availability VPN

Supported Platforms SRX Series

An Internet Key Exchange (IKE) gateway needs an external interface to communicate with a peer device. In a high availability 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 alternate physical interface to reach the peer gateway.

This feature allows the loopback interface to be configured for 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.

On SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the lo0 pseudointerface can be configured in any redundancy group; for example, RG0, RG1, RG2, and so on. However, on SRX5400, SRX5600, and SRX5800 devices, the lo0 pseudointerface cannot be configured in RG0 when it is used as an IKE gateway external interface. Because a VPN is only supported in an active-passive chassis cluster environment on SRX5400, SRX5600, and SRX5800 devices, the lo0 pseudointerface can be configured in such a setup for RG1. 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 decides the anchor SPU.

Related Documentation

- IPsec VPN Overview on page 3

Example: Configuring Redundancy Groups for Loopback Interfaces

Supported Platforms SRX Series

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 devices
- An SSG140 device or equivalent
- Two switches
- Junos OS Release 12.1x44-D10 or later for SRX Series Services Gateways

Before you begin:

Understand chassis cluster redundant Ethernet interfaces. See *Chassis Cluster Feature Guide for SRX Series Devices*.

Overview

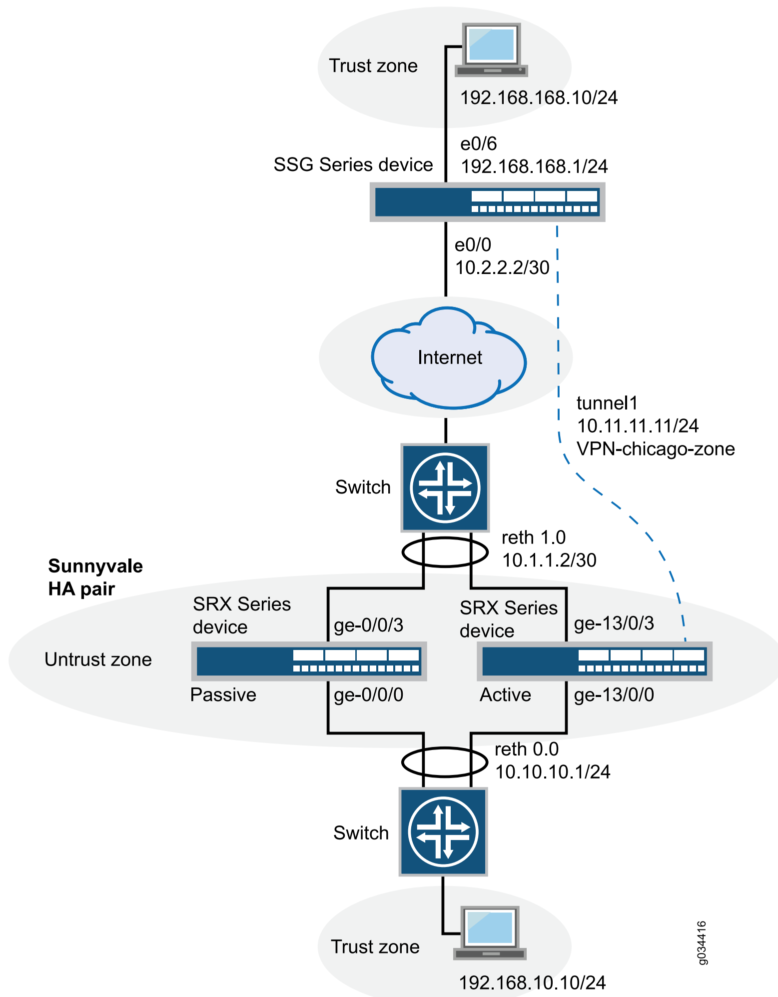
An Internet Key Exchange (IKE) gateway needs an external interface to communicate with a peer device. In a chassis cluster setup, the node on which the external interface is active selects a Services Processing Unit (SPU) to support the VPN tunnel. IKE and IPsec packets are processed on that SPU. Therefore, 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.

NOTE: **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 29 on page 311 shows an example of a loopback chassis cluster VPN topology. In this topology, the SRX Series chassis cluster device is located in Sunnyvale, California. The SRX Series 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 29: Loopback Interface for Chassis Cluster VPN



lo0.1 10.3.3.3/30 Untrust zone	A logical interface on loopback that may be active on either node in the cluster depending on the activeness of its RG.
st0.0 10.11.11.10/24 vpn-chicago-zone	A logical interface on the secure tunnel interface for IPSec VPN tunnel.

Configuration

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

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.

Related Documentation

- Understanding Loopback Interface for a High Availability VPN on page 309

PART 6

Configuring IPv6 IPsec VPNs

CHAPTER 13

Configuring IPv6 IPsec VPNs

VPN Feature Support for IPv6 Addresses

Supported Platforms SRX Series, vSRX

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 50 on page 319 defines the support for IPv6 addresses in VPN features.

Table 50: IPv6 Address Support in VPN Features

Feature	Supported	Not Supported	Exceptions
IKE and IPsec Support:			
IKEv1 and IKEv2	X		Unless specified, all supported features are applicable for IKEv1 and IKEv2.
Route-based VPN	X		–
Policy-based VPN	X	X	IPv6 policy-based VPNs are not supported on SRX Series devices 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.

Table 50: IPv6 Address Support in VPN Features *(continued)*

Feature	Supported	Not Supported	Exceptions
Site-to-site VPN	X		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	X		
Dialup VPN	X		
AutoVPN		X	–
Group VPN		X	–
Point-to-point tunnel interfaces	X		–
Point-to-multipoint tunnel interfaces		X	–
Hub-and-spoke scenario for site-to-site VPNs	X		–
Numbered and unnumbered tunnel interfaces	X		–
Unicast static and dynamic (RIP, OSPF, BGP) routing	X		–
Multicast dynamic routing (PIM)		X	–
Virtual router	X		–
Logical system		X	–
Automatic and manual SA and key management	X		–
Multiple SPUs	X		–
Chassis cluster	X		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	X		–
SNMP MIB	X		–

Table 50: IPv6 Address Support in VPN Features *(continued)*

Feature	Supported	Not Supported	Exceptions
Local address selection	X		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	X		–
Xauth or modecfg over IPv6		X	–
SPC insert	X		–
ISSU	X		–
DNS name as IKE gateway address	X		As with IPv4 tunnels, peer gateway address changes in the DNS name are not supported with IPv6 tunnels.
Preshared key or certificate authentication	X		–
NAT-Traversal (NAT-T) for IPv4 IKE peers	X		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	X		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 devices.	X		–
Generic proposals and policies for IPv6 and IPv4	X		–
General IKE ID	X		–
ESP and AH transport modes		X	These modes are not supported for IPv4.
ESP and AH tunnel modes	X		AH tunnel mode with mutable extension headers and options is not supported.
Extended sequence number		X	–

Table 50: IPv6 Address Support in VPN Features *(continued)*

Feature	Supported	Not Supported	Exceptions
Single proxy ID pairs	X		–
Multiple traffic selector pairs	X		Supported with IKEv1 only.
Lifetime of IKE or IPsec SA, in seconds	X		–
Lifetime of IKE SA, in kilobytes	X		–
VPN monitoring		X	Configuration with IPv6 tunnels cannot be committed.
DF bit	X		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	X		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	X		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	X		–
VPN session affinity	X		–
Multicast traffic		X	–
Tunnel IP services (Screen, NAT, ALG, IPS, AppSecure)	X		–
Packet reordering for IPv6 fragments over tunnel		X	–
PKI Support:			
PKI in virtual router	X		–
RSA signature authentication (512-, 1024-, 2048-, or 4096-bit key size)	X		–
DSA signature authentication (1024-, 2048-, or 4096-bit key size)	X		–

Table 50: IPv6 Address Support in VPN Features *(continued)*

Feature	Supported	Not Supported	Exceptions
ECDSA signatures	X		–
Certificate chain authentication		X	–
Automatic or manual enrollment over IPv4	X		–
Automatic or manual revocation over IPv4	X		–
Automatic or manual enrollment over IPv6		X	–
Automatic or manual revocation over IPv6		X	–
IPv6 addresses within PKI certificate fields		X	–

Related Documentation

- Understanding VPN Tunnel Modes on page 165
- IPsec VPN Overview on page 3

Understanding IPv6 IKE and IPsec Packet Processing

Supported Platforms SRX Series, vSRX

This topic includes the following sections:

- IPv6 IKE Packet Processing on page 323
- IPv6 IPsec Packet Processing on page 325

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 51 on page 324 lists the ID types and their assigned values in the identification payload.

Table 51: ISAKMP ID Types and Their Values

ID Type	Value
RESERVED	0
ID_IPV4_ADDR	1
ID_FQDN	2
ID_USER_FQDN	3
ID_IPV4_ADDR_SUBNET	4
ID_IPV6_ADDR	5
ID_IPV6_ADDR_SUBNET	6
ID_IPV4_ADDR_RANGE	7
ID_IPV6_ADDR_RANGE	8
ID_DER_ASN1_DN	9
ID_DER_ASN1_GN	10
ID_KEY_ID	11
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.

NOTE: **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.

NOTE: Packet reordering for IPv6 fragments over a tunnel is not supported.

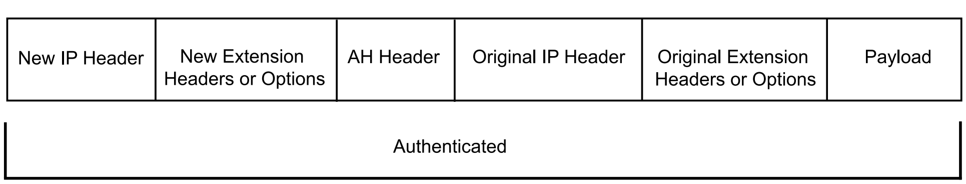
Devices with IPv6 addressing do not perform fragmentation. IPv6 hosts should either perform path MTU discovery or send packets smaller than the IPv6 minimum MTU size of 1280 bytes.

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 30 on page 325.

Figure 30: IPv6 AH Tunnel Mode



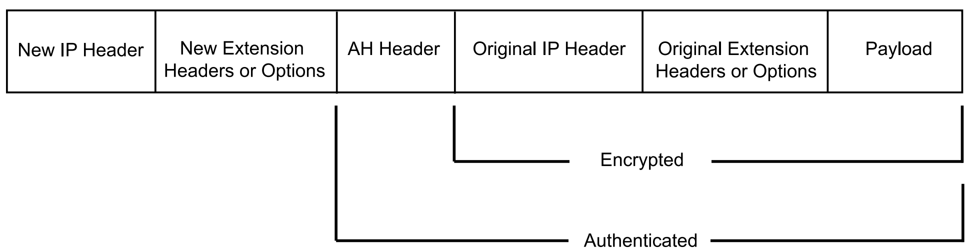
Unlike ESP, the AH authentication algorithm covers the outer header as well as any new extension headers and options.

NOTE: AH tunnel mode on SRX Series devices does not support IPv4 mutable options or IPv6 mutable extension headers. See Table 52 on page 326.

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 31 on page 326.

Figure 31: 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 devices. Table 52 on page 326 shows the IPv4 options or IPv6 extension headers that are supported with the ESP or AH protocol on SRX Series devices. If an unsupported IPsec packet is received, ICV calculation fails and the packet is dropped.

Table 52: 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
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.

NOTE: **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 53 on page 327 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 53: 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.

Table 53: IPv6 Header Construction for IPv6-in-IPv6 and IPv4-in-IPv6 Tunnel Modes *(continued)*

Header Fields	Outer Header at Encapsulator	Inner Header at Decapsulator
dest address	Constructed.	No change.
Extension headers	Never copied.	No change.

Table 54 on page 328 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 54: IPv4 Header Construction for IPv6-in-IPv4 and IPv4-in-IPv4 Tunnel Modes

Header Fields	Outer Header	Inner Header
version	4.	No change.
header length	Constructed.	No change.
DS field	Copied from the inner header.	No change.
ECN field	Copied from the inner header.	Constructed.
total length	Constructed.	No change.
ID	Constructed.	No change.
flags (DF, MF)	Constructed.	No change.
fragment offset	Constructed.	No change.
TTL	64.	Decrement.
protocol	AH, ESP	No change.
checksum	Constructed.	Constructed.
src address	Constructed.	No change.
dest address	Constructed.	No change.
options	Never copied.	No change.

For IPv6-in-IPv4 tunnel mode, the Don’t Fragment (DF) bit is cleared by default. If the **df-bit set** or **df-bit copy** options are configured at the [**edit security ipsec vpn *vpn-name***] hierarchy level for the corresponding IPv4 VPN, the DF bit is set in the outer IPv4 header.

For IPv4-in-IPv4 tunnel mode, the DF bit in the outer IPv4 header is based on the **df-bit** option configured for the inner IPv4 header. If **df-bit** is not configured for the inner IPv4 header, the DF bit is cleared in the outer IPv4 header.

Related Documentation

- IPsec VPN Overview on page 3
- IPv6 IPsec Configuration Overview on page 329
- Example: Configuring an IPv6 IPsec Manual VPN on page 329

IPv6 IPsec Configuration Overview

Supported Platforms SRX Series, vSRX

Juniper Networks supports manual and autokey IKE with preshared keys configurations for IPv6 IPsec VPN.

- Manual VPN—In a manual VPN configuration, the secret keys and security associations (SAs) are manually configured on the tunnel endpoints using the manual key mechanism. To create an IPv6 IPsec manual VPN, see “Example: Configuring an IPv6 IPsec Manual VPN” on page 329.
- 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. For more information on Phase 1 negotiations, see “Understanding Phase 1 of IKE Tunnel Negotiation” on page 18.
  - Phase 2—In this phase, the participants negotiate the IPsec SAs for authenticating and encrypting the IPv6 data packets. For more information on Phase 2 negotiations, see “Understanding Phase 2 of IKE Tunnel Negotiation” on page 20.

Related Documentation

- Understanding IPv6 IKE and IPsec Packet Processing on page 323
- Example: Configuring an IPv6 IPsec Manual VPN on page 329

Example: Configuring an IPv6 IPsec Manual VPN

Supported Platforms SRX Series, vSRX

This example shows how to configure an IPv6 IPsec manual VPN.

Requirements

Before you begin:

- Understand how VPNs work. See "IPsec VPN Overview" on page 3.
- Understand IPv6 IPsec packet processing. See "Understanding IPv6 IKE and IPsec Packet Processing" on page 323.

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

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

To confirm that the configuration is working properly, perform this task:

- Verifying Security Algorithms on page 332

Verifying Security Algorithms

Purpose Determine if security algorithms are applied or not.

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

Related Documentation

- Understanding IPv6 IKE and IPsec Packet Processing on page 323
- IPv6 IPsec Configuration Overview on page 329

Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN

Supported Platforms SRX Series, vSRX

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.

NOTE: IPv6 policy-based VPNs are supported only on standalone SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

- Requirements on page 332
- Overview on page 332
- Configuration on page 336
- Verification on page 345

Requirements

This example uses the following hardware:

- SRX300 device

Before you begin:

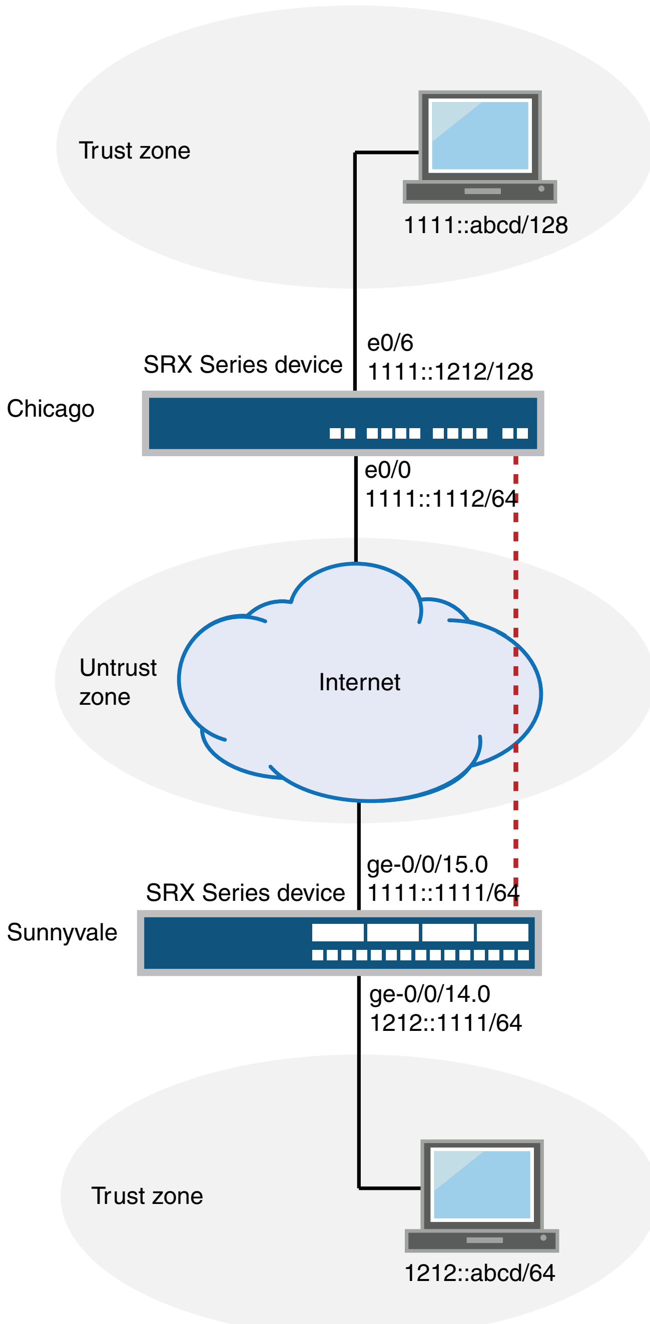
- Understand how VPNs work. See "IPsec VPN Overview" on page 3.
- Understand IPv6 IKE and IPsec packet processing. See "Understanding IPv6 IKE and IPsec Packet Processing" on page 323.

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 32 on page 333 shows an example of an IPv6 IKE policy-based VPN topology. In this topology, one SRX Series device is located in Sunnyvale, and another SRX Series device (this can be a second SRX Series device or a third-party device) is located in Chicago.

Figure 32: 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 55 on page 334 through Table 59 on page 336.

Table 55: Interface, Security Zone, and Address Book Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/14.0	1212::1111/64
	ge-0/0/15.0	1111::1111/64
Security zones	trust	• All system services are allowed. • The ge-0/0/14.0 interface is bound to this zone.
	untrust	• IKE is the only allowed system service. • 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. • The address for this address book entry is 1212::abcd/64.
	chicago	• This address is for the untrust zone's address book. • The address for this address book entry is 1111::abcd/128.

Table 56: IPv6 IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipv6-ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: **aes-128-cbc**
Policy	ipv6-ike-phase1-policy	• Mode: Aggressive • Proposal reference: ipv6-ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw-chicago	• IKE policy reference: ipv6-ike-phase1-policy • External interface: ge-0/0/15.0 • Gateway address: 1111::1112/64

Table 57: IPv6 IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipv6-ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
Policy	ipv6-ipsec-phase2-policy	• Proposal reference: ipv6-ipsec-phase2-proposal • PFS: Diffie-Hellman group2
VPN	ipv6-ike-vpn-chicago	• IKE gateway reference: gw-chicago • IPsec policy reference: ipv6-ipsec-phase2-policy

Table 58: 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:
		  ◦ source-address sunnyvale
		  ◦ destination-address chicago
		  ◦ application any
		• Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago
		• 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:
		  ◦ source-address chicago
		  ◦ destination-address sunnyvale
		  ◦ application any
		• Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago
		• Permit action: tunnel pair-policy ipv6-vpn-tr-untr

This security policy permits all traffic from the trust zone to the untrust zone.
 	
NOTE: 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:
		  ◦ source-address any
		  ◦ source-destination any
		  ◦ application any
		• Action: permit

Table 59: 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. NOTE: 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

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 1212::1111/64
set interfaces ge-0/0/15 unit 0 family inet6 address 1111::1111/64
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
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 1212::abcd/64
set security address-book book1 attach zone trust
set security address-book book2 address chicago 1111::abcd/64
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 1212::1111/64
user@host# set interfaces ge-0/0/15 unit 0 family inet6 address 1111::1111/64
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
```

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 1212::abcd/64
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 1111::abcd/64
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 1212::1111/64;
      }
   }
}
ge-0/0/15 {
   unit 0 {
      family inet6 {
         address 1111::1111/64;
      }
   }
}

[edit]
user@host# show routing-options
static {
   route 0.0.0.0/0 next-hop 1.1.1.1;
}

[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 1212::abcd/64;
   attach {
      zone trust;
   }
}
   book2 {
      address chicago 1111::abcd/64;
      attach {
         zone untrust;
      }
   }
```

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

Configuring IKE

CLI Quick Configuration

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

```
set security ike proposal 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 1111111111111111
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 1111::1112/64
```

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 1111::1112**

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 "$9$jrHP5QFn/ApPfBIEhr1Yg4aDik.P5z3Dj9Apu1I7—dbgoJGD";
      ## SECRET-DATA
}
gateway gw-chicago {
   ike-policy ipv6-ike-phase1-policy;
   address 1111::1112;
   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:

- 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

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status on page 346
- Verifying the IPsec Phase 2 Status on page 347

Verifying the IKE Phase 1 Status

Purpose Verify the IKE Phase 1 status.

Action

NOTE: 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 device 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 1212::abcd/64 to 1111::abcd/128.

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        1111::1112        UP     e48efd6a444853cf  0d09c59aafb720be  Aggressive


user@host> show security ike security-associations index 5 detail
IKE peer 1111::1112, Index 5,
  Role: Initiator, State: UP
  Initiator cookie: e48efd6a444853cf, Responder cookie: 0d09c59aafb720be
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 1111::1111:500, Remote: 1111::1112:500
  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: 1111::1111:500, Remote: 1111::1112:500
    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

NOTE: 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   1111::1112
  2     ESP:aes-128/sha1 9a4db486 3597/ unlim   -   root 500   1111::1112
```

```
user@host> show security ipsec security-associations index 2 detail
  Virtual-system: Root
  Local Gateway: 111::1111, Remote Gateway: 1111::1112
  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.

NOTE: For some third-party vendors, the proxy ID must be manually entered to match.

Related Documentation

- Understanding IPv6 IKE and IPsec Packet Processing on page 323
- IPv6 IPsec Configuration Overview on page 329
- Example: Configuring an IPv6 IPsec Manual VPN on page 329

PART 7

Configuring Public Key Infrastructure

CHAPTER 14

Managing Digital Certificates with PKI

Understanding Certificates and PKI

Supported Platforms SRX Series, vSRX

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 Public Key Infrastructure (PKI) provides an infrastructure for digital certificate management.

This topic includes the following sections:

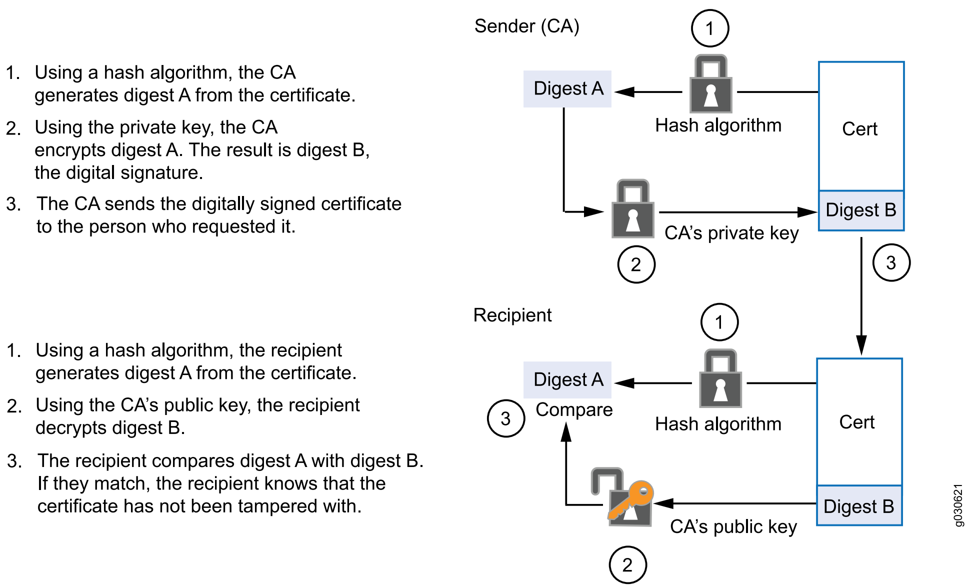
Certificate Signatures and Verification

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 33 on page 354 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 33 on page 354 illustrates this process.

NOTE: 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 33: Digital Signature Verification



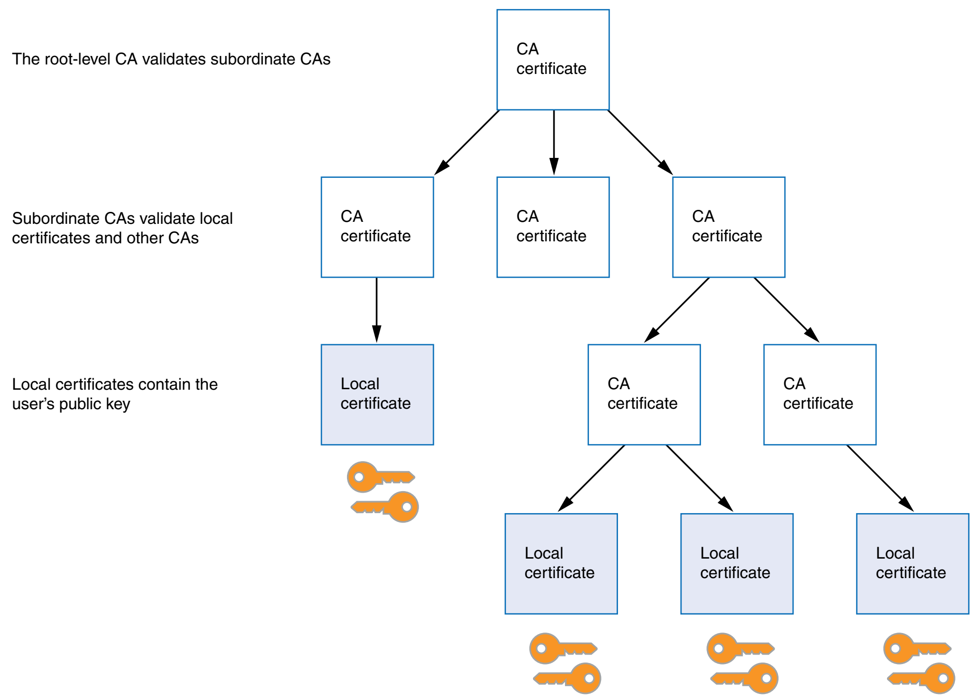
When Digital Signature Algorithm (DSA) signatures are used, the SHA-1 hash algorithm is used to generate the digest. When Rivest-Shamir-Adleman (RSA) signatures are used, SHA-1 is the default hash algorithm used to generate the digest; you can specify the SHA-256 hash algorithm with the **digest** option 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.

Public Key Infrastructure

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.

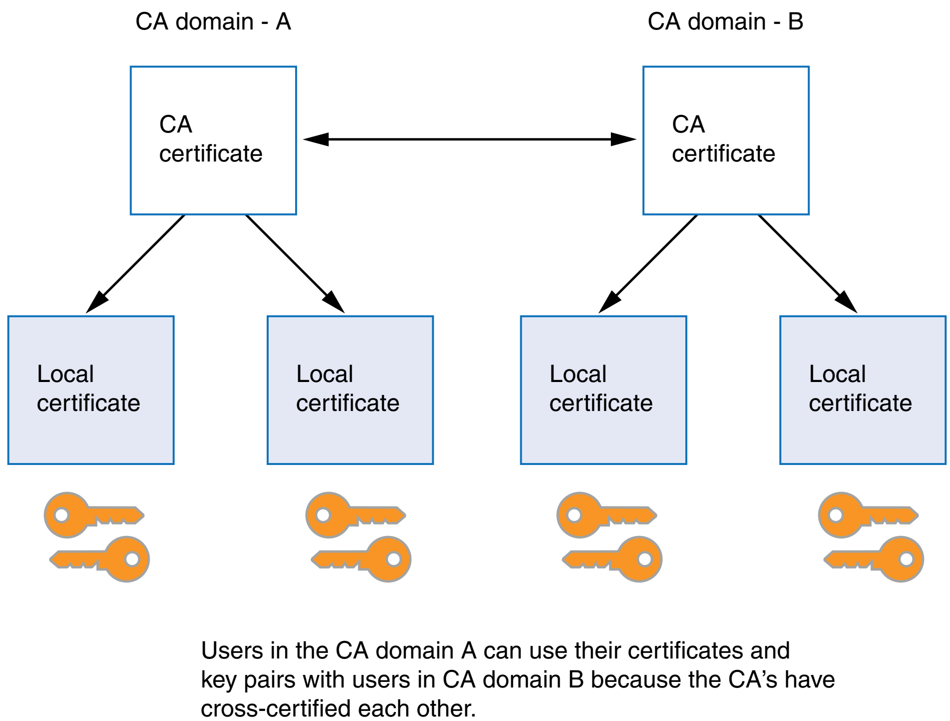
Figure 34 on page 355 shows the structure of a single-domain certificate authority with multiple hierarchy levels.

Figure 34: 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 35 on page 356.

Figure 35: Cross-Certification



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:

- 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.

  - CA certificate — When the certificate is issued by the CA and loaded into the Junos 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)

For convenience and practicality, PKI must be transparently managed and implemented. Toward this goal, Junos OS supports the following features:

- Generates a public-private key pair.
- Loads multiple local certificates from different CAs.
- Uses either Simple Certificate Enrollment Protocol (SCEP) or CMPv2 for certificate enrollment.
- Delivers a certificate when establishing an IPsec tunnel.
- Validates a certificate path upward through a single level of CA authorities.
- Supports the Public-Key Cryptography Standards #7 (PKCS #7) cryptographic . As a result, the device can accept X.509 certificates and certificate revocation lists (CRLs) packaged within a PKCS #7 envelope.

NOTE: Junos OS supports a PKCS #7 file size of up to 7 KB.

- Retrieves CRLs online retrieval through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP).

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.

Related Documentation

- Digital Certificates Configuration Overview on page 361
- Understanding Certificate Chains on page 445
- IPsec VPN Overview on page 3

Cryptographic Key Handling Overview

Supported Platforms SRX Series, vSRX

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.

NOTE: 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

Related Documentation

- Understanding Certificates and PKI on page 353

Understanding CMPv2 and SCEP Certificate Enrollment

Supported Platforms SRX Series, vSRX

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 60 on page 358 describes the differences between the CMPv2 and SCEP certificate enrollment protocols.

Table 60: 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.

Related Documentation

- Understanding Certificate Enrollment with CMPv2 on page 359
- Understanding Certificates and PKI on page 353
- Enrolling Digital Certificates Online: Configuration Overview on page 361

Understanding Certificate Enrollment with CMPv2

Supported Platforms SRX Series, vSRX

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 on page 359
- End-Entity Certificate with Issuer CA Certificate on page 360
- End-Entity Certificate with CA Certificate Chain on page 360

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.

NOTE: If the Initialization Response message needs to be authenticated by a CA certificate, the CA certificate must be enrolled prior to any end-entity certificate enrollment.

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.

NOTE: CA certificate reenrollment is not supported.

A CA might issue a new CA certificate prior to the expiration of the current CA certificate. If a new CA certificate arrives during certificate reenrollment with a new public key, the new CA certificate is not saved in the device.

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.

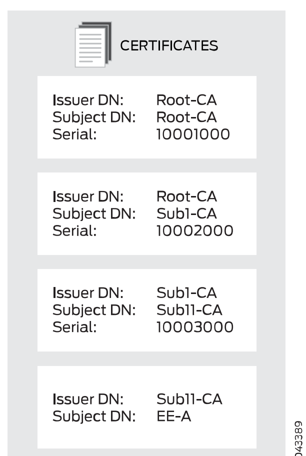
The Initialization Response message can contain an end-entity certificate as well as a self-signed issuer CA certificate. The end-entity certificate is first stored in the certificate store, and then the CA certificate is checked. If the CA certificate is found and the subject distinguished name (DN) 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 36 on page 360, the Initialization Response message includes the end-entity certificate and three CA certificates in a certificate chain.

Figure 36: 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**.

Related Documentation

- Understanding Certificate Authority Profiles on page 375
- Understanding Certificate Chains on page 445
- Understanding CMPv2 and SCEP Certificate Enrollment on page 358

Digital Certificates Configuration Overview

Supported Platforms SRX Series, vSRX

You can obtain CA and local certificates manually, or online using Simple Certificate Enrollment Protocol (SCEP) or CMPv2. Certificates are verifiable and renewable, and you can delete them when they are no longer needed.

Manual certificate processing includes generation of a PKCS10 request, submission to the CA, retrieval of the signed certificate, and manually loading of 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.

This topic includes the following sections:

- Enrolling Digital Certificates Online: Configuration Overview on page 361
- Manually Generating Digital Certificates: Configuration Overview on page 362

Enrolling Digital Certificates Online: Configuration Overview

Supported Platforms SRX Series, vSRX

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 "Example: Generating a Public-Private Key Pair" on page 374.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 375.
3. For SCEP only, enroll the CA certificate. See "Enrolling a CA Certificate Online Using SCEP" on page 378.
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 379.
5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 381.

Manually Generating Digital Certificates: Configuration Overview

Supported Platforms SRX Series, vSRX

To obtain digital certificates manually:

1. Generate a key pair on the device. See "Example: Generating a Public-Private Key Pair" on page 374.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 375.
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 382.
4. Load the certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 384.
5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 381.
6. If necessary, load the certificate's CRL on the device. See "Example: Manually Loading a CRL onto the Device" on page 435.
7. If necessary, configure the CA profile with CRL locations. See "Example: Configuring a Certificate Authority Profile with CRL Locations" on page 436

Related Documentation

- Understanding Certificates and PKI on page 353
- Example: Verifying Certificate Validity on page 438
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 436
- Deleting Certificates (CLI Procedure) on page 385

CHAPTER 15

Configuring Digital Certificate Validation

Understanding Digital Certificate Validation

Supported Platforms SRX Series, vSRX

During IKE negotiation, the PKI daemon on an SRX Series device 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 Devices

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 device.

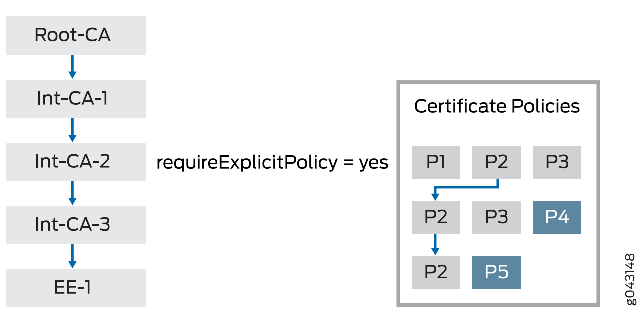
On the SRX Series device, policy OIDs are configured in an IKE policy with the **policy-oids** configuration statement at the [**edit security ike policy** ***policy-name*** **certificate**] hierarchy level. You can configure up to five policy OIDs. For a peer's certificate to be validated successfully,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

No Policy OIDs Configured on SRX Series Devices

If no policy OID is configured on the SRX Series device, 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 37 on page 364 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.

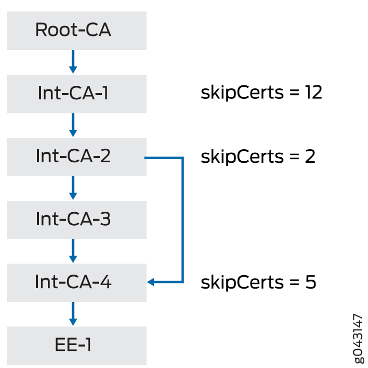
Figure 37: Policy Validation with requireExplicitPolicy Field



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 38 on page 365 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 38: Policy Validation with skipCerts Field



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

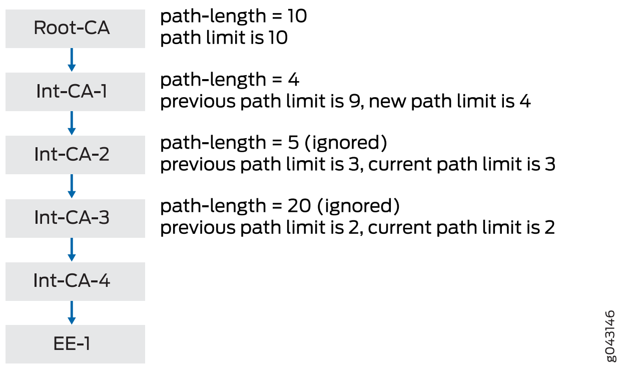
Path Length Validation

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 39 on page 366 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 39: 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.

EE Certificates

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.

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.

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 aborted.

CRL Signature Validation

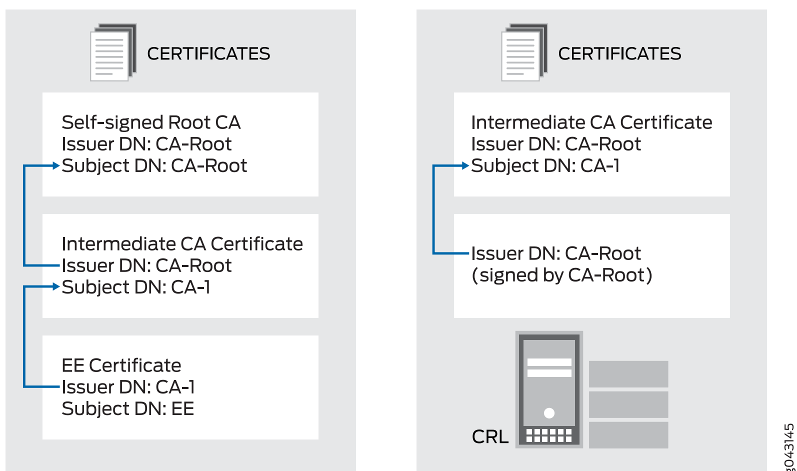
In Phase 1 negotiations, participants check the certificate revocation list (CRL) to see if certificates received during an IKE exchange are still valid. The CRL is periodically downloaded for CA profiles configured with CRL as the certificate revocation check. Downloaded CRL files must be verified before 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 40 on page 367 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 40: 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.

Related Documentation

- Example: Improving Digital Certificate Validation by Configuring Policy OIDs on an SRX Series Device on page 368
- Understanding Certificates and PKI on page 353

Example: Improving Digital Certificate Validation by Configuring Policy OIDs on an SRX Series Device

Supported Platforms SRX Series, vSRX

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 device. This example shows how to configure policy OIDs in the IKE policy on an SRX Series device.

NOTE: You must ensure that at least one of the policy OIDs configured on the SRX Series device 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 devices.
- Configure an IPsec VPN tunnel. See "IPsec VPN with Autokey IKE Configuration Overview" on page 22. 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 device is lc-igloo-root.

Configuration

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:

- 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

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 device. Check the **policy-oids** values in the peer's certificate chain with the values configured on the SRX Series device.

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 device.

Related Documentation

- Understanding Digital Certificate Validation on page 363

CHAPTER 16

Generating a Public-Private Key Pair

Understanding Public Key Cryptography

Supported Platforms SRX Series, vSRX

The public-private key pairs used in public key cryptography play an important role in the use of digital certificates. A public-private key pair encrypts and decrypts data. Data encrypted with a public key, which the owner makes available to the public, can be decrypted with the corresponding private key only, which the owner keeps secret and protected. For example, if Alice wants to send Bob an encrypted message, Alice can encrypt it with Bob's public key and send it to him. Bob then decrypts the message with his private key.

The reverse process is also useful: encrypting data with a private key and decrypting it with the corresponding public key. This process is known as creating a digital signature. For example, if Alice wants to present her identity as the sender of a message, she can encrypt the message with her private key and send the message to Bob. Bob then decrypts the message with Alice's public key, thus verifying that Alice is indeed the sender.

When you generate a public-private key pair, the device automatically saves the key pair in a file in the certificate store, where it is subsequently used in certificate request commands. The generated key pair is saved as *certificate-id*.

NOTE: The default RSA and DSA key size is 1024 bits. Simple Certificate Enrollment Protocol (SCEP) supports RSA certificates only. CMPv2 supports RSA, DSA, and ECDSA certificate types.

NOTE: If the device renews a great number of certificates at once, thus using up keys rapidly, it might run out of pregenerated keys and have to generate them promptly for each new request. In this case, the generation of keys might affect the performance of the device, especially in a high-availability environment where the performance of the device might slow down for a number of minutes.

Related Documentation

- Understanding Certificates and PKI on page 353
- Example: Generating a Public-Private Key Pair on page 374
- Digital Certificates Configuration Overview on page 361

Example: Generating a Public-Private Key Pair

Supported Platforms SRX Series, vSRX

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 ca-ipsec.

Configuration

Step-by-Step Procedure

To generate a public-private key pair:

- Create a certificate key pair.

  [edit]
  user@host> **request security pki generate-key-pair certificate-id ca-ipsec**

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
```

Related Documentation

- Understanding Public Key Cryptography on page 373
- Example: Verifying Certificate Validity on page 438
- Digital Certificates Configuration Overview on page 361

CHAPTER 17

Configuring Certificate Authority Profiles

Understanding Certificate Authority Profiles

Supported Platforms SRX Series, vSRX

A certificate authority (CA) profile configuration contains information specific to a CA. You can have multiple CA profiles on the device. For example, you might have one profile for Microsoft and one for Entrust. Each profile is associated with a CA certificate. If you want to load a new CA certificate without removing the older one, you must create a new CA profile (for example, Microsoft-2008).

NOTE: The following CAs are supported: Entrust, Microsoft, and Verisign. SCEP only supports the Microsoft CA.

Related Documentation

- Understanding Certificates and PKI on page 353
- Example: Configuring a CA Profile on page 375

Example: Configuring a CA Profile

Supported Platforms SRX Series, vSRX

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. 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

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
     revocation-check crl refresh-interval 48 url http://www.my-ca.com/my-crl.crl
   ```

2. Specify the enrollment retry value.

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

3. 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
   ```

4. 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.

Related Documentation

- Understanding Certificate Authority Profiles on page 375
- Digital Certificates Configuration Overview on page 361

CHAPTER 18

Configuring CA and Local Certificates

Understanding Online CA Certificate Enrollment

Supported Platforms SRX Series, vSRX

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.

Related Documentation

- Understanding Public Key Cryptography on page 373
- Understanding Certificates and PKI on page 353
- Enrolling a CA Certificate Online Using SCEP on page 378
- Example: Enrolling a Local Certificate Online Using SCEP on page 379

Understanding Local Certificate Requests

Supported Platforms SRX Series, vSRX

When you create a local certificate request, the device generates a CA 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)

NOTE: 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.

Related Documentation

- Understanding Certificates and PKI on page 353
- Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server on page 382

Enrolling a CA Certificate Online Using SCEP

Supported Platforms SRX Series, vSRX

Before you begin:

1. Generate a public and private key pair. See "Example: Generating a Public-Private Key Pair" on page 374.
2. Create a CA profile. See "Example: Configuring a CA Profile" on page 375.

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**.

Related Documentation

- Understanding Online CA Certificate Enrollment on page 377
- Digital Certificates Configuration Overview on page 361
- Example: Enrolling a Local Certificate Online Using SCEP on page 379
- Example: Using SCEP to Automatically Renew a Local Certificate on page 381

Example: Enrolling a Local Certificate Online Using SCEP

Supported Platforms SRX Series, vSRX

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

- Requirements on page 379
- Overview on page 379
- Configuration on page 380
- Verification on page 380

Requirements

Before you begin:

- Generate a public and private key pair. See “Example: Generating a Public-Private Key Pair” on page 374.
- Configure a certificate authority profile. See “Example: Configuring a CA Profile” on page 375.
- For SCEP, enroll the CA certificate. See “Enrolling a CA Certificate Online Using SCEP” on page 378.

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 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

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
   ```

   NOTE: 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).

Verification

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

Release History Table

Release	Description
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40 the **scep** keyword is supported and required.

Related Documentation

- Digital Certificates Configuration Overview on page 361
- Enrolling Digital Certificates Online: Configuration Overview on page 361

Example: Using SCEP to Automatically Renew a Local Certificate

Supported Platforms SRX Series, vSRX

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 “Enrolling Digital Certificates Online: Configuration Overview” on page 361.
- Obtain a local certificate. See “Example: Enrolling a Local Certificate Online Using SCEP” on page 379.

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

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 15.1X49-D40 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.

Release History Table

Release	Description
15.1X49-D40	Starting in Junos OS 15.1X49-D40 the **scep** keyword is supported and required.

Related Documentation

- Enrolling Digital Certificates Online: Configuration Overview on page 361

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

Supported Platforms SRX Series, vSRX

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

- Requirements on page 383
- Overview on page 383
- Configuration on page 383
- Verification on page 383

Requirements

Generate a public and private key. See “Example: Generating a Public-Private Key Pair” on page 374.

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

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)
```


Related Documentation

- Understanding Local Certificate Requests on page 377
- Digital Certificates Configuration Overview on page 361

Understanding Certificate Loading

Supported Platforms SRX Series, vSRX

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.

NOTE: You can load multiple EE certificates onto the device.

Related Documentation

- Understanding Certificates and PKI on page 353
- Example: Loading CA and Local Certificates Manually on page 384

Example: Loading CA and Local Certificates Manually

Supported Platforms SRX Series, vSRX

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

Requirements

Before you begin:

- Generate a public-private key pair. See "Example: Generating a Public-Private Key Pair" on page 374.
- Create a CA profile. See "Understanding Certificate Authority Profiles" on page 375.

NOTE: 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 382.

Overview

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

Configuration

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
```

Related Documentation

- Understanding Certificate Loading on page 384
- Digital Certificates Configuration Overview on page 361
- Example: Using SCEP to Automatically Renew a Local Certificate on page 381
- Example: Verifying Certificate Validity on page 438
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 436

Deleting Certificates (CLI Procedure)

Supported Platforms SRX Series, vSRX

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.

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

Related Documentation

- Digital Certificates Configuration Overview on page 361

Example: Configuring PKI

Supported Platforms SRX Series, vSRX

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

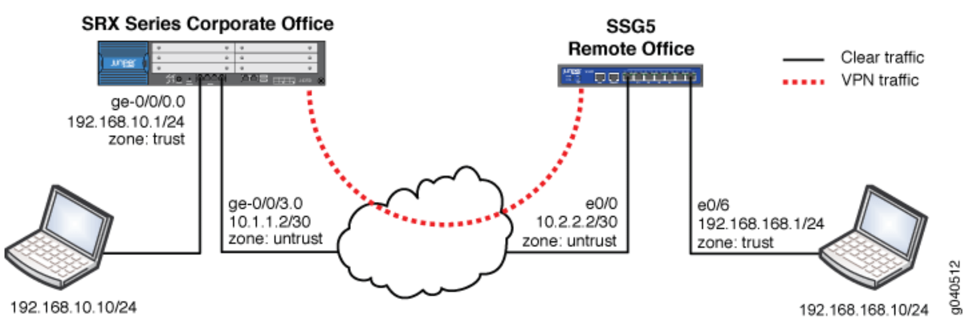
Before you begin:

- Ensure that the internal LAN interface of the SRX Series device 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 41 on page 387 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 41: Network Topology Diagram



NOTE: **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.

NOTE: 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.

NOTE: 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.

NOTE: 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.

NOTE: 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.
- **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

  NOTE: **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.

  NOTE: **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).

NOTE: 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.

NOTE:

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.

NOTE:

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

- Loading CA and Local Certificates on page 394
- Configuring the IPsec VPN with the Certificates on page 396

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 *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.

   [edit]
   user@host# **set security pki ca-profile ms-ca revocation-check crl**

   NOTE: 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. 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**

4. 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**

   NOTE: Currently you can configure only one URL. Support for backup URL configuration is not available.

5. 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**

6. 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.

   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/hz6Cza10BxakUxfGxF7UWYWHaWFFYLqo6vXNO8r
OS5Yak7rWANAsMob3E2X/1adlQIRi4QFTjkBqGI+MTEDGnqFsJBqrB6oyqGtdcSU
u0qUivMvgKQVCx8hpx99J3EBTurfWL1pCNlBmZggNogb6MbwESOCAwEAAaAwMC4G
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)
```

   NOTE: 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. Generate the PKCS10 certificate request to be sent to the CA.

   user@host> **request security pki generate-certificate-request certificate-id *id-name* subject subject-name (domain-name *domain-name* | ip-address *device-ip* | email *email-id*) filename *filename***

3. 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

   NOTE: 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)
```

NOTE: 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 41 on page 387

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
```


NOTE: 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

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 400.

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 405.
- 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 405.

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.

NOTE: 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.

NOTE: 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 405.

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 405.

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 403.
- 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

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**

NOTE: 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.

NOTE: 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
```

NOTE: 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.

NOTE: 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.

NOTE: 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 399.

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 VPN Overview on page 3
- Understanding Certificates and PKI on page 353

CHAPTER 19

Managing Certificate Revocation

Understanding Online Certificate Status Protocol

Supported Platforms SRX Series, vSRX

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 device. Based on the response from the server, the VPN connection is allowed or denied. OCSP responses are not cached on SRX Series devices.

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 devices 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 device. 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 device verifies the OCSP response signature using the CA certificate enrolled in the SRX Series device. 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 device. The OCSP response is validated using the responder's CA certificate. The SRX Series device 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 device 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 device.

Related Documentation

- Comparison of Online Certificate Status Protocol and Certificate Revocation List on page 419
- Improving Security by Configuring OCSP for Certificate Revocation Status on page 420

Understanding Certificate Revocation Lists

Supported Platforms SRX Series, vSRX

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, participants check the CRL list to see if certificates received during an IKE exchange are still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, the device tries to download it automatically from the CRL distribution point of the local certificate. If the device fails to connect to the URL in the certificate distribution point (CDP), it tries to retrieve the CRL from the URL configured in the CA profile.

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.

Related Documentation

- Understanding Certificates and PKI on page 353
- Example: Manually Loading a CRL onto the Device on page 435
- Example: Verifying Certificate Validity on page 438
- Deleting a Loaded CRL (CLI Procedure) on page 439
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 436

Comparison of Online Certificate Status Protocol and Certificate Revocation List

Supported Platforms MX Series, SRX Series, vSRX

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.

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

Related Documentation

- Understanding Online Certificate Status Protocol on page 417
- *Example: Improving Security by Configuring OCSP for Certificate Revocation Status*

Improving Security by Configuring OCSP for Certificate Revocation Status

Supported Platforms SRX Series, vSRX

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

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 61 on page 421 shows the Phase 1 options used in this example.

Table 61: Phase 1 Options for OCSP Configuration Example

Option	Peer A	Peer B
IKE proposal	ike_prop	ike_prop
Authentication method	RSA signatures	RSA signatures
DH group	group2	group2
Authentication algorithm	SHA 1	SHA 1
Encryption algorithm	3DES CBC	3DES CBC
IKE policy	ike_policy	ike_policy
Mode	aggressive	aggressive
Proposal	ike_prop	ike_prop
Certificate	local-certificate localcert1	local-certificate localcert1
IKE gateway	jsr_gateway	jsr_gateway
Policy	ike_policy	ike_policy
Gateway address	198.51.100.50	192.0.2.50
Remote identity	localcert11.example.net	-
Local identity	-	localcert11.example.net
External interface	reth1	ge-0/0/2.0
Version	v2	v2

Table 62 on page 421 shows the Phase 2 options used in this example.

Table 62: 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

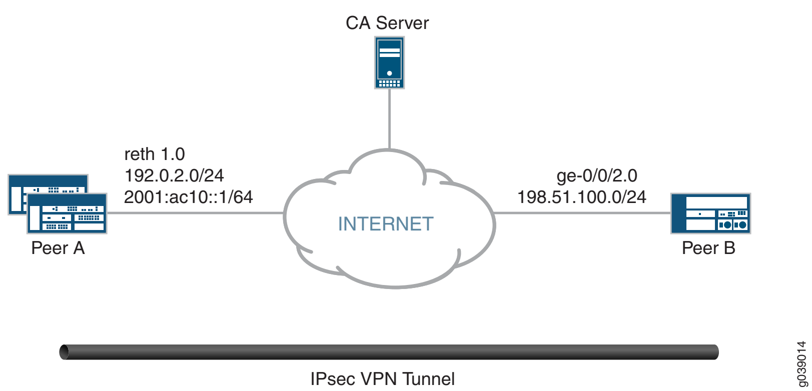
Table 62: Phase 2 Options for OCSP Configuration Example *(continued)*

Option	Peer A	Peer B
Lifetime seconds	1200	1200
Lifetime kilobytes	150,000	150,000
IPsec policy	ipsec_policy	ipsec_policy
PFC keys	group2	group2
Proposal	ipsec_prop	ipsec_prop
VPN	test_vpn	test_vpn
Bind interface	st0.1	st0.1
IKE gateway	jsr_gateway	jsr_gateway
Policy	ipsec_policy	ipsec_policy
Establish tunnels	-	immediately

Topology

Figure 42 on page 422 shows the peer devices that are configured in this example.

Figure 42: OCSP Configuration Example



Configuration

- Configuring Peer A on page 423
- Configuring Peer B on page 427

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 *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 *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

Confirm that the configuration is working properly.

- Verifying CA Certificates on page 431
- Verifying Local Certificates on page 432

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)
```


NOTE: 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

- Understanding Online Certificate Status Protocol on page 417
- Understanding Certificates and PKI on page 353

Example: Manually Loading a CRL onto the Device

Supported Platforms SRX Series, vSRX

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

- Requirements on page 435
- Overview on page 435
- Configuration on page 436
- Verification on page 436

Requirements

Before you begin:

1. Generate a public and private key pair. See "Example: Generating a Public-Private Key Pair" on page 374.
2. Generate a certificate request. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 382.
3. Configure a certificate authority (CA) profile. See "Example: Configuring a CA Profile" on page 375.
4. Load your certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 384.

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.)

NOTE: 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

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
```

NOTE: 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.

Related Documentation

- Understanding Certificate Revocation Lists on page 418
- Digital Certificates Configuration Overview on page 361
- Example: Verifying Certificate Validity on page 438
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 436
- Deleting a Loaded CRL (CLI Procedure) on page 439

Example: Configuring a Certificate Authority Profile with CRL Locations

Supported Platforms SRX Series, vSRX

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 “Example: Generating a Public-Private Key Pair” on page 374.
2. Create a CA profile or profiles containing information specific to a CA. See “Example: Configuring a CA Profile” on page 375.
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 382.
4. Load the certificate onto the device. See “Example: Loading CA and Local Certificates Manually” on page 384.
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 435.

Overview

In Phase 1 negotiations, you check the CRL list to see if the certificate that you received during an IKE exchange is still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, Junos OS tries to retrieve the CRL through the LDAP or HTTP CRL location defined within the CA certificate itself. If no URL address is defined in the CA certificate, the device uses the URL of the server that you define for that CA certificate. If you do not define a CRL URL for a particular CA certificate, the device gets the CRL from the URL in the CA profile configuration.

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

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

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.

Related Documentation

- Understanding Certificate Revocation Lists on page 418
- Example: Manually Loading a CRL onto the Device on page 435
- Example: Verifying Certificate Validity on page 438
- Deleting a Loaded CRL (CLI Procedure) on page 439
- Deleting Certificates (CLI Procedure) on page 385

Example: Verifying Certificate Validity

Supported Platforms SRX Series, vSRX

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

- Requirements on page 438
- Overview on page 438
- Configuration on page 438
- Verification on page 439

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

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
```

NOTE: 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.

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

Related Documentation

- Understanding Certificate Revocation Lists on page 418
- Example: Manually Loading a CRL onto the Device on page 435
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 436
- Deleting a Loaded CRL (CLI Procedure) on page 439

Deleting a Loaded CRL (CLI Procedure)

Supported Platforms SRX Series, vSRX

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.

Related Documentation

- Understanding Certificate Revocation Lists on page 418
- Example: Manually Loading a CRL onto the Device on page 435
- Example: Verifying Certificate Validity on page 438
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 436

CHAPTER 20

Generating Self-Signed Certificates

Understanding Self-Signed Certificates

Supported Platforms SRX Series, vSRX

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.

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

This topic includes the following sections:

Generating Self-Signed Certificates

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.

Automatically Generating Self-Signed Certificates

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.

Manually Generating Self-Signed Certificates

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.

Related Documentation

- Understanding Certificates and PKI on page 353
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 443
- Example: Manually Generating Self-Signed Certificates on page 442

Example: Manually Generating Self-Signed Certificates

Supported Platforms SRX Series, vSRX

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

Requirements

Before you begin, generate a public private key pair. See "Example: Generating a Public-Private Key Pair" on page 374.

Overview

For a manually generated self-signed certificate, you specify the 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, which refers a key-pair of the same certificate-id.

Configuration

Step-by-Step Procedure

To generate the self-signed certificate manually:

- Create the self-signed certificate.

  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**

Verification

To verify the certificate was properly generated and loaded, enter the **show security pki local-certificate** operational mode command.

Related Documentation

- Understanding Self-Signed Certificates on page 441
- Digital Certificates Configuration Overview on page 361
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 443

Using Automatically Generated Self-Signed Certificates (CLI Procedure)

Supported Platforms SRX Series

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.

You can add the following statement to your configuration if you want to use the automatically generated self-signed certificate to provide access to HTTPS services:

```
system {
 services {
  web-management {
   http {
    interface [ ... ];
   } https {
    system-generated-certificate;
    interface [ ... ];
   }
  }
 }
}
```

The device uses the following distinguished name for the automatically generated certificate:

" CN=<device serial number>, CN=system generated, CN=self-signed"

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

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# **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

- Understanding Self-Signed Certificates on page 441
- Digital Certificates Configuration Overview on page 361
- Example: Manually Generating Self-Signed Certificates on page 442

CHAPTER 21

Configuring a Device for Certificate Chains

Understanding Certificate Chains

Supported Platforms SRX Series, vSRX

Multilevel Hierarchy for Certificate Authentication

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

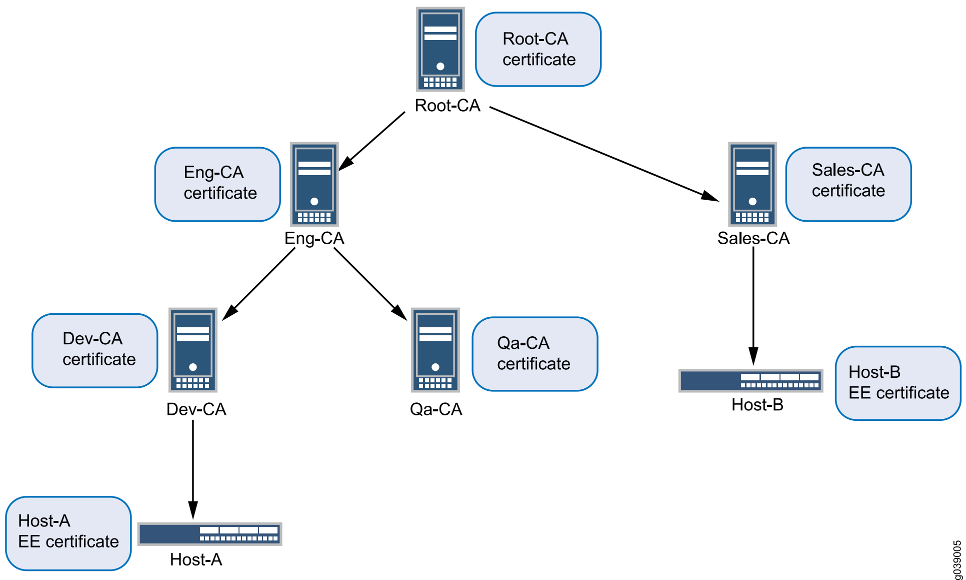
When a single-level hierarchy for certificate-based authentication is employed, all EE certificates in the network must be signed by the same CA. All firewall devices must have the same CA certificate enrolled for peer certificate validation. The certificate payload sent during IKE negotiation only contains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

In the example CA hierarchy shown in Figure 43 on page 446, 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 43: 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/";
        }
    }
}
```

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.

A VPN device must be able to check a peer's certificate for its revocation status. A device can use the CA certificate received from its peer to extract the URL to dynamically download the CA's CRL and check the revocation status of the peer's certificate. 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 CRL from the peer's CA and check the revocation status of the peer's certificate. In Figure 43 on page 446, 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.

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

The properties of a parent CA profile are inherited for dynamic CA profiles. In Figure 43 on page 446, 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 parent 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.

NOTE: 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. Because the CA certificate is not enrolled by an administrator, it is used only for validating the CRL received from the CA server and not for validating the peer certificate.

Related Documentation

- Example: Configuring a Device for Peer Certificate Chain Validation on page 448
- Understanding Certificates and PKI on page 353
- Understanding Certificate Authority Profiles on page 375
- Understanding Certificate Revocation Lists on page 418

Example: Configuring a Device for Peer Certificate Chain Validation

Supported Platforms SRX Series, vSRX

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

Requirements

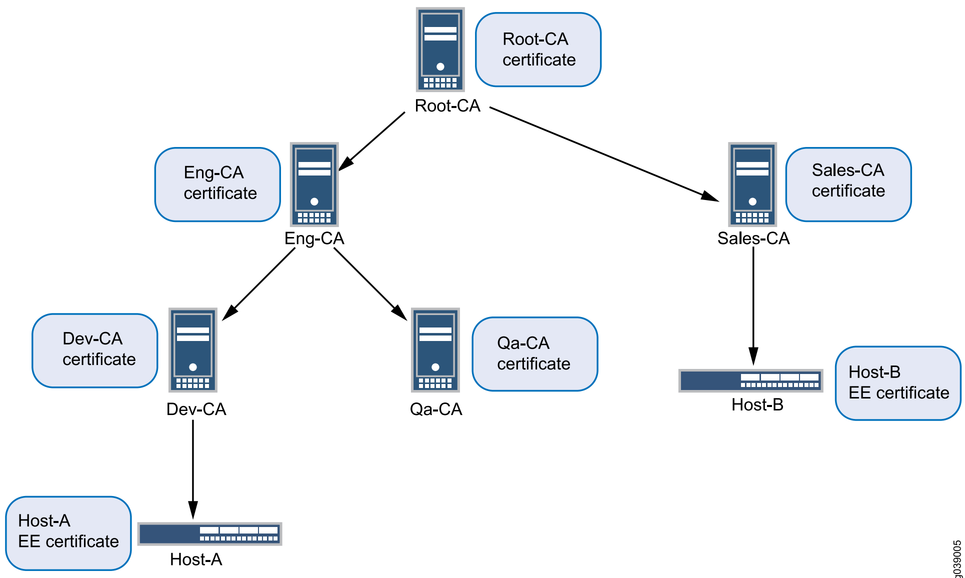
Before you begin, obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

Overview

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.

This example shows the configuration and operational commands on Host-A, as shown in Figure 44 on page 449. 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 44: Certificate Chain Example



NOTE: 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 27 for examples of configuring peer devices for VPNs.

Configuration

To configure a device for certificate chains:

- Configure CA Profiles on page 449
- Enroll Certificates on page 451
- Configure IPsec VPN Options on page 453

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. 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
```

5. 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)
```

6. 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
```

7. 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

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

- Verifying IKE Phase 1 Status on page 455
- Verifying IPsec Phase 2 Status on page 455

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 UP     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

- Checking for Revoked Certificates on page 456

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.

Related Documentation

- Understanding Certificate Chains on page 445
- Understanding Certificates and PKI on page 353
- Understanding Certificate Authority Profiles on page 375
- Understanding Certificate Revocation Lists on page 418

PART 8

Configuring AutoVPN

CHAPTER 22

Configuring AutoVPN on Hub-and-Spoke Devices

Understanding AutoVPN

Supported Platforms SRX Series, vSRX

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, an st0 interface configured at the [**edit interfaces st0 unit** *x*] hierarchy level is in point-to-point mode.
- Point-to-multipoint mode—In this mode, the **multipoint** option is configured at the [**edit interfaces st0 unit** *x*] hierarchy level on both AutoVPN hub and spokes. st0

interfaces on the hub and spokes must be numbered and the IP address configured on a spoke must exist in the hub's st0 interface subnetwork.

Table 63 on page 462 compares AutoVPN point-to-point and point-to-multipoint secure tunnel interface modes.

Table 63: Comparison Between AutoVPN Point-to-Point and Point-to-Multipoint Secure Tunnel Modes

Point-to-Point Mode	Point-to-Multipoint Mode
Uses traffic selectors to forward packets through VPN tunnels. Traffic selectors must be configured on each spoke. Administrator needs to be aware of the types of traffic that need to be permitted through the VPN tunnel.	Uses dynamic routing protocol to forward packets through VPN tunnels. The dynamic routing protocol must run in point-to-multipoint mode.
Does not support dynamic routing protocols on the st0 interface when traffic selectors are configured.	Cannot configure an st0 interface in point-to-multipoint mode with traffic selectors.
Supports IPv4 traffic only.	Supports IPv4 traffic only.
Allows spoke devices to be non-SRX Series devices.	Requires that hub and spoke devices are SRX Series devices.
Supports IKEv1 or IKEv2.	Supports IKEv1 only.
Supports dead peer detection only.	Supports dead peer detection and VPN monitoring.
Supports larger numbers of tunnels and spokes.	–

Authentication

The supported authentication for AutoVPN hubs and spokes is 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 463.

Configuration and Management

AutoVPN is configured and managed on SRX Series devices using the CLI. Multiple AutoVPN hubs can be configured on a single SRX Series device. The maximum number of spokes supported by a configured hub is specific to the model of the SRX Series device.

Related Documentation

- Understanding AutoVPN Limitations on page 462
- Understanding Spoke Authentication in AutoVPN Deployments on page 463
- AutoVPN Configuration Overview on page 465

Understanding AutoVPN Limitations

Supported Platforms SRX Series, vSRX

The following features are not supported for AutoVPN:

- AutoVPN does not support IPv6 traffic.
- 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.
- AutoVPN does not support multicast traffic.
- When IKE main mode is used with PKI authentication, all gateway configurations that use the same external interface on a device must use the same IKE policy.
- The group IKE ID user type is not supported with an IP address as the IKE ID.
- The IKE ID should not overlap with other IKE gateways when the group IKE ID user type is used.

Related Documentation

- Understanding AutoVPN on page 461

Understanding Spoke Authentication in AutoVPN Deployments

Supported Platforms SRX Series, vSRX

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:

- Group IKE ID Configuration on the Hub on page 463
- Excluding a Spoke Connection on page 465

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 Certificate Revocation Lists” on page 418 and “Understanding Certificate Authority Profiles” on page 375.

Related Documentation

- Understanding AutoVPN on page 461
- AutoVPN Configuration Overview on page 465

AutoVPN Configuration Overview

Supported Platforms SRX Series, vSRX

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.
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.
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.

   NOTE: **Only IKEv1 is supported on an SRX Series spoke with st0 interfaces in point-to-point mode.**

5. Configure an IPsec policy to match the IPsec policy configured on the hub.
6. Configure a dynamic routing protocol.

Related Documentation

- Understanding Traffic Selectors in Route-Based VPNs on page 179
- Understanding AutoVPN on page 461

Example: Configuring Basic AutoVPN with iBGP

Supported Platforms SRX Series, vSRX

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.

- Requirements on page 467
- Overview on page 467
- Configuration on page 470
- Verification on page 491

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series devices 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.

NOTE: 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

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 64 on page 467 shows the options used in this example.

Table 64: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates

Table 64: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations *(continued)*

Option	Value
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.

NOTE: **Junos OS only supports a single level of certificate hierarchy.**

Table 65 on page 468 shows the options configured on the hub and on all spokes.

Table 65: AutoVPN Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	1.1.1.1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string **SLT** in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/1.0	Spoke 1: fe-0/0/1.0 Spoke 2: ge-0/0/1.0

Table 65: AutoVPN Configuration for Hub and All Spokes *(continued)*

Option	Hub	All Spokes
VPN:		
Bind interface	st0.0	st0.0
Establish tunnels	(not configured)	Immediately on configuration commit

Table 66 on page 469 shows the configuration options that are different on each spoke.

Table 66: 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) 60.60.60.1/24	(fe-0.0/4.0) 70.70.70.1/24
Interface to Internet	(fe-0/0/1.0) 2.2.2.1/30	(ge-0/0/1.0) 3.3.3.1/30

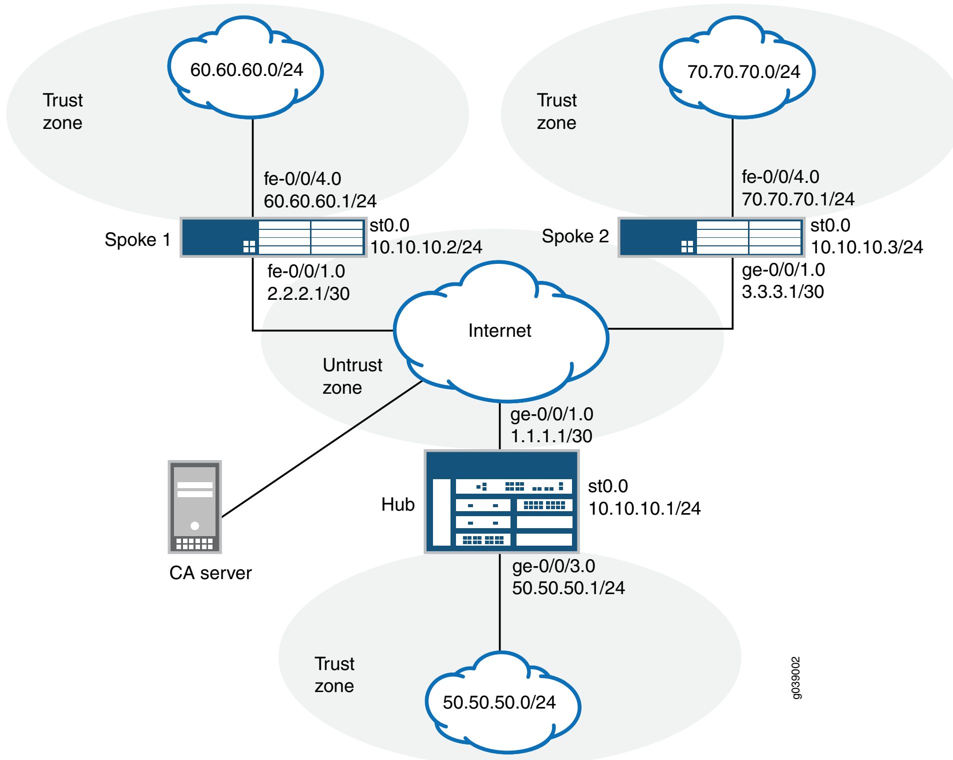
Routing information for all devices is exchanged through the VPN tunnels.

NOTE: 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 45 on page 470 shows the SRX Series devices to be configured for AutoVPN in this example.

Figure 45: Basic AutoVPN Deployment with iBGP



Configuration

To configure AutoVPN, perform these tasks:

NOTE: 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 on page 470
- Configuring the Hub on page 474
- Configuring Spoke 1 on page 480
- Configuring Spoke 2 on page 485

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 1.1.1.1 subject DC=example.net,CN=hub,OU=SLT,O=example,L=Bangalore,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: Bangalore, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bangalore, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 1.1.1.1
  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 2.2.2.1 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, 2.2.2.1
  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
```

NOTE: 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 3.3.3.1 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, 3.3.3.1
  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
```

NOTE: 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 1.1.1.1/30
set interfaces ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 1.2.3.4
set protocols bgp group ibgp peer-as 10
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 2.2.2.0/30 next-hop 1.1.1.2
set routing-options static route 3.3.3.0/30 next-hop 1.1.1.2
set routing-options autonomous-system 10
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 1.1.1.1/30
user@host# set ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 1.2.3.4
user@host# set group ibgp peer-as 10
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 2.2.2.0/30 next-hop 1.1.1.2
user@host# set static route 3.3.3.0/30 next-hop 1.1.1.2
user@host# set autonomous-system 10
```

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 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 1.1.1.1/30;
    }
  }
}
```

```
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 50.50.50.1/24;
            }
        }
    }
    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 1.2.3.4;
        peer-as 10;
        allow 10.10.10.0/24;
        export bgp_nh_self;
    }
}
[edit]
user@host# show routing-options
static {
    route 2.2.2.0/30 next-hop 1.1.1.2;
    route 3.3.3.0/30 next-hop 1.1.1.2;
    }
autonomous-system 10;
[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 2.2.2.1/30
set interfaces fe-0/0/4 unit 0 family inet address 60.60.60.1/24
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 1.1.1.0/30 next-hop 2.2.2.2
set routing-options autonomous-system 10
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 1.1.1.1
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 2.2.2.1/30
user@host# set fe-0/0/4 unit 0 family inet address 60.60.60.1/24
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 1.1.1.0/30 next-hop 2.2.2.2
user@host# set autonomous-system 10
```

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 1.1.1.1
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 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 2.2.2.1/30;
    }
  }
}
  fe-0/0/4 {
    unit 0 {
      family inet {
        address 60.60.60.1/24;
      }
    }
  }
  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 1.1.1.0/30 next-hop 2.2.2.2;
    }
autonomous-system 10;
[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 1.1.1.1;
        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 3.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 70.70.70.1/24
```

```
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 1.1.1.0/30 next-hop 3.3.3.2
set routing-options autonomous-system 10
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 1.1.1.1
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 3.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 70.70.70.1/24
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 1.1.1.0/30 next-hop 3.3.3.2**
   user@host# **set autonomous-system 10**

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 1.1.1.1**
   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 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 3.3.3.1/30;
    }
  }
}
  fe-0/0/4 {
    unit 0 {
      family inet {
        address 70.70.70.1/24;
      }
    }
  }
  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 1.1.1.0/30 next-hop 3.3.3.2;
    }
autonomous-system 10;
[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 1.1.1.1;
        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

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           3.3.3.1

5480162 UP     7a63d16a5a723df1  c471f7ae166d3a34  Main           2.2.2.1
```

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   2.2.2.1
  >268173400 ESP:des/ md5 aae5196b 3567/ unlim -   root 500   2.2.2.1
  <268173401 ESP:des/ md5 69c24d81 622/ unlim  -   root 500   3.3.3.1
  >268173401 ESP:des/ md5 e3fe0231 622/ unlim  -   root 500   3.3.3.1
```

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 60.60.60.0** command.

```
user@host> show route 60.60.60.0
inet.0: 45 destinations, 45 routes (44 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

60.60.60.0/24      *[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 70.70.70.0** command.

```
user@host> show route 70.70.70.0
inet.0: 45 destinations, 45 routes (44 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

70.70.70.0/24      *[BGP/170] 00:50:42, localpref 100
                      AS path: I
                    > to 10.10.10.3 via st0.0
```

Related Documentation

- Example: Configuring a Route-Based VPN on page 50
- *Routing Protocols Overview*

Example: Configuring Basic AutoVPN with OSPF

Supported Platforms SRX Series, vSRX

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.

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series devices 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.

NOTE: **You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.**

Overview

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 67 on page 494 shows the options used in this example.

Table 67: 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

Table 67: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPF Configurations *(continued)*

Option	Value
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.

NOTE: Junos OS only supports a single level of certificate hierarchy.

Table 68 on page 495 shows the options configured on the hub and on all spokes.

Table 68: AutoVPN Basic OSPF Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	1.1.1.1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string **SLT** in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/1.0	Spoke 1: fe-0/0/1.0 Spoke 2: ge-0/0/1.0
VPN:		

Table 68: AutoVPN Basic OSPF Configuration for Hub and All Spokes *(continued)*

Option	Hub	All Spokes
Bind interface	st0.0	st0.0
Establish tunnels	(not configured)	Immediately on configuration commit

Table 69 on page 496 shows the configuration options that are different on each spoke.

Table 69: 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: 60.60.60.1/24	fe-0.0/4.0: 70.70.70.1/24
Interface to Internet	fe-0/0/1.0: 2.2.2.1/30	ge-0/0/1.0: 3.3.3.1/30

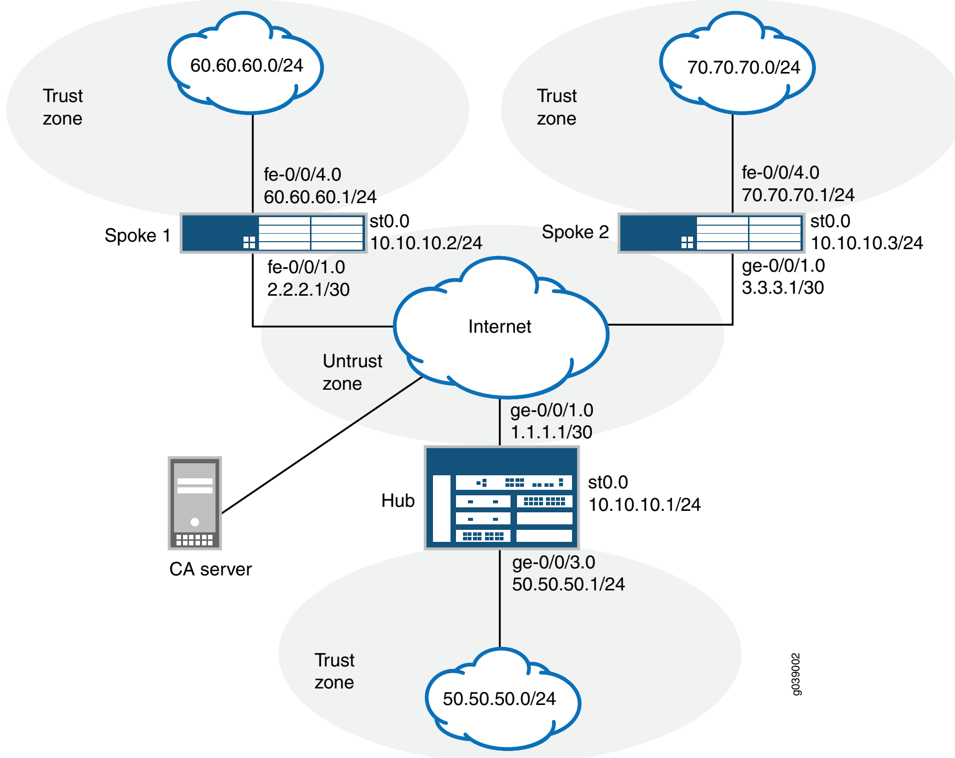
Routing information for all devices is exchanged through the VPN tunnels.

NOTE: 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 46 on page 497 shows the SRX Series devices to be configured for AutoVPN in this example.

Figure 46: Basic AutoVPN Deployment with OSPF



Configuration

To configure AutoVPN, perform these tasks:

NOTE: 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 1.1.1.1 subject DC=example.net,CN=hub,OU=SLT,O=example,L=Bangalore,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: Bangalore, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bangalore, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 1.1.1.1
  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 2.2.2.1 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, 2.2.2.1
  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
```

NOTE: 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 3.3.3.1 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, 3.3.3.1
  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
```

NOTE: 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 1.1.1.1/30
set interfaces ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 2.2.2.0/30 next-hop 1.1.1.2
set routing-options static route 3.3.3.0/30 next-hop 1.1.1.2
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 1.1.1.1/30
user@host# set ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 1.1.1.2
user@host# set static route 3.3.3.0/30 next-hop 1.1.1.2
```

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 1.1.1.1/30;
    }
  }
}
  ge-0/0/3 {
    unit 0 {
      family inet {
        address 50.50.50.1/24;
      }
    }
  }
  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 2.2.2.0/30 next-hop 1.1.1.2;
   route 3.3.3.0/30 next-hop 1.1.1.2;
}
[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 2.2.2.1/30
```

```
set interfaces fe-0/0/4 unit 0 family inet address 60.60.60.1/24
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 1.1.1.0/30 next-hop 2.2.2.2
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 1.1.1.1
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 2.2.2.1/30
user@host# set fe-0/0/4 unit 0 family inet address 60.60.60.1/24
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 1.1.1.0/30 next-hop 2.2.2.2
```

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 1.1.1.1
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 2.2.2.1/30;
    }
  }
}
  fe-0/0/4 {
    unit 0 {
      family inet {
        address 60.60.60.1/24;
      }
    }
  }
  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 1.1.1.0/30 next-hop 2.2.2.2;
  }
  [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 1.1.1.1;
      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 3.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 70.70.70.1/24
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 1.1.1.1/32 next-hop 3.3.3.2
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 1.1.1.1
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 3.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 70.70.70.1/24
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 1.1.1.1/32 next-hop 3.3.3.2
```

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 1.1.1.1
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 3.3.3.1/30;
    }
  }
}
  fe-0/0/4 {
    unit 0 {
      family inet {
        address 70.70.70.1/24;
      }
    }
  }
  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 1.1.1.1/32 next-hop 3.3.3.2;
}
[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 1.1.1.1;
    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

Confirm that the configuration is working properly.

- Verifying IKE Phase 1 Status on page 517
- Verifying IPsec Phase 2 Status on page 517

- Verifying IPsec Next-Hop Tunnels on page 517
- Verifying OSPF on page 518
- Verifying Learned Routes on page 518

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           2.2.2.1

5480161 UP     d455050707bc3eaf  b3dde111232270d2  Main           3.3.3.1
```

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   2.2.2.1
  >268173400 ESP:des/ md5 bb48d228 2954/ unlim -   root 500   2.2.2.1
  <268173401 ESP:des/ md5 bcd1390b 3530/ unlim -   root 500   3.3.3.1
  >268173401 ESP:des/ md5 77fcf6e2 3530/ unlim -   root 500   3.3.3.1
```

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 60.60.60.0
 inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

60.60.60.0/24      *[OSPF/10] 00:51:13, metric 2
                    > to 10.10.10.2 via st0.0
```

From operational mode, enter the **show route 70.70.70.0** command.

```
user@host> show route 70.70.70.0
inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

70.70.70.0/24      *[OSPF/10] 00:51:48, metric 2
                    > to 10.10.10.3 via st0.0
```

Related Documentation

- Example: Configuring a Route-Based VPN on page 50

Example: Configuring AutoVPN with iBGP and ECMP

Supported Platforms SRX Series, vSRX

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.

Requirements

This example uses the following hardware and software components:

- Two supported SRX Series devices 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.

NOTE: You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

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 70 on page 520 shows the options used in this example.

Table 70: 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
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.

NOTE: Junos OS only supports a single level of certificate hierarchy.

Table 71 on page 521 shows the options configured on the hub and on the spoke.

Table 71: AutoVPN iBGP ECMP Configuration for Hub and Spoke 1

Option	Hub	Spoke 1
IKE gateway:		
Remote IP address	hub-to-spoke-gw-1: Dynamic hub-to-spoke-gw-2: Dynamic	spoke-to-hub-gw-1: 1.1.1.1 spoke-to-hub-gw-2: 1.1.2.1
Remote IKE ID	hub-to-spoke-gw-1: DN on the spoke's certificate with the string **SLT** in the OU field hub-to-spoke-gw-2: DN on the spoke's certificate with the string **SBU** in the OU field	spoke-to-hub-gw-1: DN on the hub's certificate spoke-to-hub-gw-2: DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	hub-to-spoke-gw-1: ge-0/0/1.0 hub-to-spoke-gw-2: ge-0/0/2.0	spoke-to-hub-gw-1: fe-0/0/1.0 spoke-to-hub-gw-2: fe-0/0/2.0
VPN:		
Bind interface	hub-to-spoke-vpn-1: st0.0 hub-to-spoke-vpn-2: st0.1	spoke-to-hub-1: st0.0 spoke-to-hub-2: st0.1
Establish tunnels	(not configured)	Immediately on configuration commit

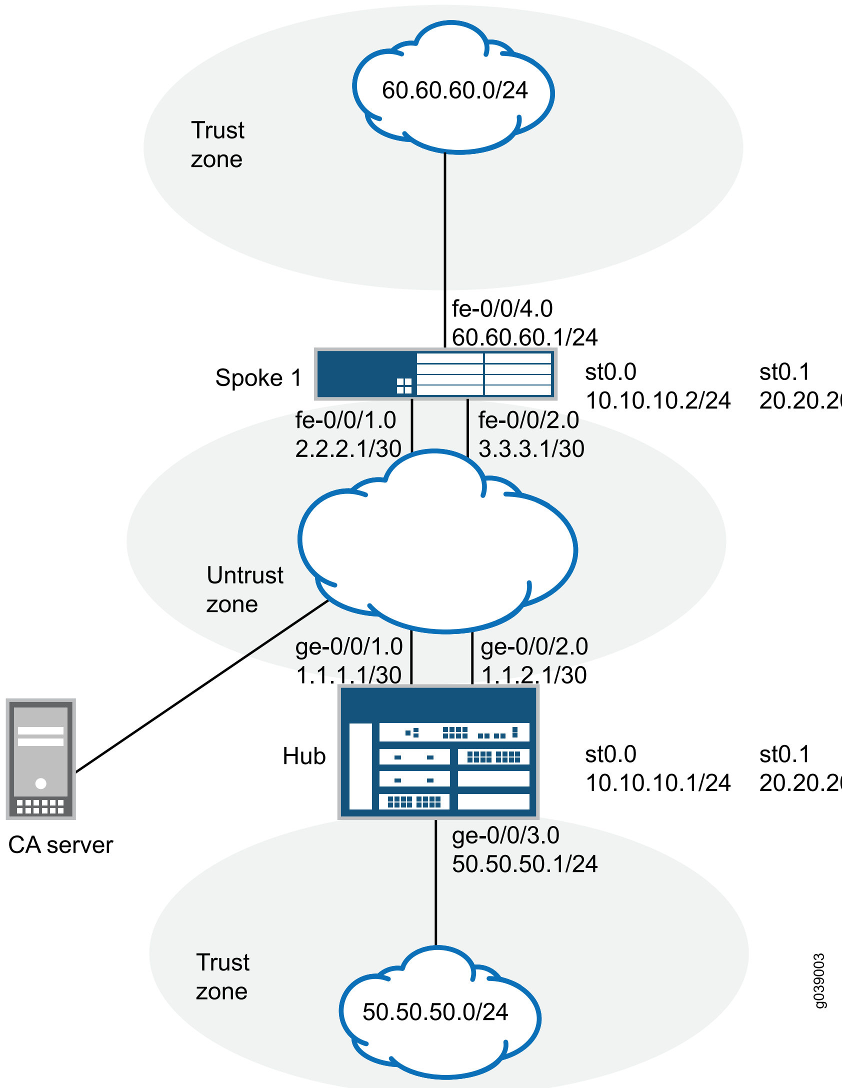
Routing information for all devices is exchanged through the VPN tunnels.

NOTE: 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 47 on page 522 shows the SRX Series devices to be configured for AutoVPN in this example.

Figure 47: AutoVPN Deployment with iBGP and ECMP



Configuration

To configure AutoVPN, perform these tasks:

NOTE: 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 on page 523
- Configuring the Hub on page 527
- Configuring Spoke 1 on page 534

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 1.1.1.1 subject DC=example.net,CN=hub,OU=SLT,O=example,L=Bangalore,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 1.1.2.1 subject DC=example.net,CN=hub_backup,OU=SBU,O=example,L=Bangalore,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: Bangalore, Common name: hub, Domain component: example.net
  Subject string:
     C=IN, DC=example.net, ST=KA, L=Bangalore, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 1.1.1.1
  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: Bangalore, Common name: hub_backup, Domain component:
example.net
  Subject string:
   C=IN, DC=example.net, ST=KA, L=Bangalore, O=example, OU=SBU, CN=hub_backup

  Alternate subject: "hub_backup@example.net", example.net, 1.1.2.1
  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 2.2.2.1 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 3.3.3.1 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, 2.2.2.1
  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, 3.3.3.1
  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
```

NOTE: 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 1.1.1.1/30
set interfaces ge-0/0/2 unit 0 family inet address 1.1.2.1/30
set interfaces ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 20.20.20.1/24
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 1.2.3.4
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 20.20.20.1
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 cluster 1.2.3.5
set protocols bgp group ibgp-2 multipath
set protocols bgp group ibgp-2 allow 20.20.20.0/24
set routing-options static route 2.2.2.0/30 next-hop 1.1.1.2
set routing-options static route 3.3.3.0/30 next-hop 1.1.2.2
set routing-options autonomous-system 10
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 1.1.1.1/30
user@host# set ge-0/0/2 unit 0 family inet address 1.1.2.1/30
user@host# set ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 20.20.20.1/24
```

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 1.2.3.4
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 20.20.20.1
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 cluster 1.2.3.5
user@host# set group ibgp-2 multipath
user@host# set group ibgp-2 allow 20.20.20.0/24

[edit routing-options]
user@host# set static route 2.2.2.0/30 next-hop 1.1.1.2
user@host# set static route 3.3.3.0/30 next-hop 1.1.2.2
user@host# set autonomous-system 10
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 1.1.1.1/30;
      }
    }
}
  ge-0/0/2 {
    unit 0 {
      family inet {
        address 1.1.2.1/30;
      }
    }
  }
  ge-0/0/3 {
    unit 0 {
      family inet {
        address 50.50.50.1/24;
      }
    }
  }
  st0 {
    unit 0 {
      multipoint;
      family inet {
        address 10.10.10.1/24;
      }
    }
    unit 1 {
      multipoint;
      family inet {
        address 20.20.20.1/24;
      }
    }
  }
[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 1.2.3.4;
    multipath;
    allow 10.10.10.0/24;
  }
  group ibgp-2 {
```

```
        type internal;
        local-address 20.20.20.1;
        export lan_nw;
        cluster 1.2.3.5;
        multipath;
        allow 20.20.20.0/24;
    }
}
[edit]
user@host# show routing-options
static {
    route 2.2.2.0/30 next-hop 1.1.1.2;
    route 3.3.3.0/30 next-hop 1.1.2.2;
    }
autonomous-system 10;
    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 2.2.2.1/30
set interfaces fe-0/0/2 unit 0 family inet address 3.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 60.60.60.1/24
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set interfaces st0 unit 1 family inet address 20.20.20.2/24
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 20.20.20.2
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 neighbor 20.20.20.1
set routing-options static route 1.1.1.0/30 next-hop 2.2.2.2
set routing-options static route 1.1.2.0/30 next-hop 3.3.3.2
set routing-options autonomous-system 10
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 1.1.1.1
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 1.1.2.1
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 2.2.2.1/30
   user@host# set fe-0/0/2 unit 0 family inet address 3.3.3.1/30
   ```

```
user@host# set fe-0/0/4 unit 0 family inet address 60.60.60.1/24
user@host# set st0 unit 0 family inet address 10.10.10.2/24
user@host# set st0 unit 1 family inet address 20.20.20.2/24
```

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 20.20.20.2
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 neighbor 20.20.20.1

[edit routing-options]
user@host# set static route 1.1.1.0/30 next-hop 2.2.2.2
user@host# set static route 1.1.2.0/30 next-hop 3.3.3.2
user@host# set autonomous-system 10
```

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 1.1.1.1
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 1.1.2.1
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 2.2.2.1/30;
    }
  }
}
  fe-0/0/2 {
    unit 0 {
      family inet {
        address 3.3.3.1/30;
      }
    }
  }
  fe-0/0/4 {
    unit 0 {
      family inet {
        address 60.60.60.1/24;
      }
    }
  }
  st0 {
    unit 0 {
      family inet {
        address 10.10.10.2/24;
      }
    }
    unit 1 {
      family inet {
        address 20.20.20.2/24;
      }
    }
  }
[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 20.20.20.2;
    export lan_nw;
    neighbor 20.20.20.1;
  }
}
[edit]
user@host# show routing-options
static {
  route 1.1.1.0/30 next-hop 2.2.2.2;
  route 1.1.2.0/30 next-hop 3.3.3.2;
  }
autonomous-system 10;
[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 1.1.1.1;
    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 1.1.2.1;
    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

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           2.2.2.1

3733048 UP     a88db7ed23ec5f6b  c88b81dff52617a5  Main           3.3.3.1
```

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   2.2.2.1
  >268173315 ESP:des/ md5 101de6f7 1152/ unlim -   root 500   2.2.2.1
  <268173313 ESP:des/ md5 272e29c0 1320/ unlim -   root 500   3.3.3.1
  >268173313 ESP:des/ md5 a3bf8fad 1320/ unlim -   root 500   3.3.3.1
```

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
20.20.20.2        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               10       4819       4820       0       2 1d 12:15:14
1/1/1/0              0/0/0/0
20.20.20.2               10       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 60.60.60.0 detail** command.

```
user@host> show route 60.60.60.0 detail
inet.0: 47 destinations, 48 routes (46 active, 0 holddown, 1 hidden)
60.60.60.0/24 (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: 20.20.20.2 via st0.1, selected
                Protocol next hop: 10.10.10.2
                Indirect next hop: 15c8000 262142
                Protocol next hop: 20.20.20.2
                Indirect next hop: 15c80e8 262143
                State: <Act Int Ext>
                Local AS:    10 Peer AS:    10
                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: 20.20.20.2
                Next hop type: Router
                Next hop: 20.20.20.2 via st0.1, selected
                Protocol next hop: 20.20.20.2
                Indirect next hop: 15c80e8 262143
                State: <NotBest Int Ext>
                Inactive reason: Not Best in its group - Update source
                Local AS:    10 Peer AS:    10
                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 60.60.60.0** 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
60.60.60.0/24      user     0                    ulst 262144     1
                                                 indr 262142     2
                              10.10.10.2         ucst   572      3 st0.0
                                                 indr 262143     2
                              20.20.20.2         ucst   573      3 st0.1
```

Related Documentation

- Example: Configuring a Route-Based VPN on page 50

Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels

Supported Platforms SRX Series, vSRX

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.

Requirements

This example uses the following hardware and software components:

- Two supported SRX Series devices 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.

NOTE: You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

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 72 on page 546 shows the options used in this example.

Table 72: 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.

NOTE: Junos OS only supports a single level of certificate hierarchy.

Table 73 on page 547 shows the options configured on the hub and on the spoke.

Table 73: AutoVPN IBGP Active-Backup Tunnel Configuration for Hub and Spoke 1

Option	Hub	Spoke 1
IKE gateway:		
Remote IP address	hub-to-spoke-gw-1: Dynamic hub-to-spoke-gw-2: Dynamic	spoke-to-hub-gw-1: 1.1.1.1 spoke-to-hub-gw-2: 1.1.2.1
Remote IKE ID	hub-to-spoke-gw-1: DN on the spoke's certificate with the string **SLT** in the OU field hub-to-spoke-gw-2: DN on the spoke's certificate with the string **SBU** in the OU field	spoke-to-hub-gw-1: DN on the hub's certificate spoke-to-hub-gw-2: DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	hub-to-spoke-gw-1: ge-0/0/1.0 hub-to-spoke-gw-2: ge-0/0/2.0	spoke-to-hub-gw-1: fe-0/0/1.0 spoke-to-hub-gw-2: fe-0/0/2.0
VPN:		
Bind interface	hub-to-spoke-vpn-1: st0.0 hub-to-spoke-vpn-2: st0.1	spoke-to-hub-1: st0.0 spoke-to-hub-2: st0.1
VPN monitor	hub-to-spoke-vpn-1: ge-0/0/1.0 (source interface) hub-to-spoke-vpn-2: ge-0/0/2.0 (source interface)	spoke-to-hub-1: 1.1.1.1 (destination IP) spoke-to-hub-2: 1.1.2.1 (destination IP)
Establish tunnels	(not configured)	Immediately on configuration commit

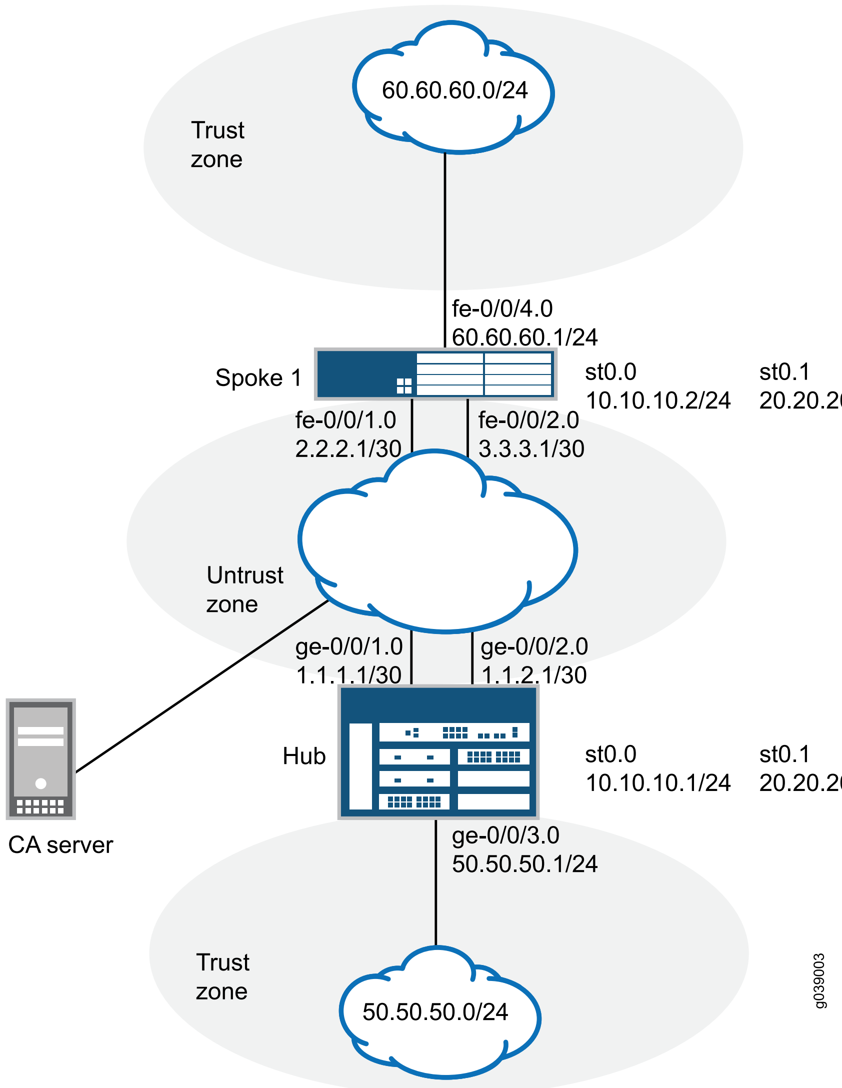
Routing information for all devices is exchanged through the VPN tunnels.

NOTE: 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 48 on page 548 shows the SRX Series devices to be configured for AutoVPN in this example.

Figure 48: 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 60.60.60.0/24 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 60.60.60.0/24 is rerouted through the backup tunnel.

Configuration

To configure AutoVPN, perform these tasks:

NOTE: 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 1.1.1.1 subject DC=example.net,CN=hub,OU=SLT,O=example,L=Bangalore,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 1.1.2.1 subject DC=example.net,CN=hub_backup,OU=SBU,O=example,L=Bangalore,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: Bangalore, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bangalore, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 1.1.1.1
  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: Bangalore, Common name: hub_backup, Domain component:
example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bangalore, O=example, OU=SBU, CN=hub_backup

  Alternate subject: "hub_backup@example.net", example.net, 1.1.2.1
  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 2.2.2.1 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 3.3.3.1 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, 2.2.2.1
  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, 3.3.3.1
  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
```

NOTE: 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 1.1.1.1/30
set interfaces ge-0/0/2 unit 0 family inet address 1.1.2.1/30
set interfaces ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 20.20.20.1/24
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 1.2.3.4
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 20.20.20.1
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 cluster 1.2.3.5
set protocols bgp group ibgp-2 allow 20.20.20.0/24
set routing-options static route 2.2.2.0/30 next-hop 1.1.1.2
set routing-options static route 3.3.3.0/30 next-hop 1.1.2.2
set routing-options autonomous-system 10
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 1.1.1.1/30
user@host# set ge-0/0/2 unit 0 family inet address 1.1.2.1/30
user@host# set ge-0/0/3 unit 0 family inet address 50.50.50.1/24
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 20.20.20.1/24
```


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 1.2.3.4
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 20.20.20.1
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 cluster 1.2.3.5
user@host# set group ibgp-2 allow 20.20.20.0/24

[edit routing-options]
user@host# set static route 2.2.2.0/30 next-hop 1.1.1.2
user@host# set static route 3.3.3.0/30 next-hop 1.1.2.2
user@host# set autonomous-system 10
```

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 1.1.1.1/30;
    }
  }
}
  ge-0/0/2 {
    unit 0 {
      family inet {
        address 1.1.2.1/30;
      }
    }
  }
  ge-0/0/3 {
    unit 0 {
      family inet {
        address 50.50.50.1/24;
      }
    }
  }
  st0 {
    unit 0 {
      multipoint;
      family inet {
        address 10.10.10.1/24;
      }
    }
    unit 1 {
      multipoint;
      family inet {
        address 20.20.20.1/24;
      }
    }
  }
[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 1.2.3.4;
      allow 10.10.10.0/24;
   }
   group ibgp-2 {
      type internal;
      local-address 20.20.20.1;
      export lan_nw;
      cluster 1.2.3.5;
      allow 20.20.20.0/24;
   }
}
[edit]
user@host# show routing-options
static {
   route 2.2.2.0/30 next-hop 1.1.1.2;
   route 3.3.3.0/30 next-hop 1.1.2.2;
   }
autonomous-system 10;
[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 2.2.2.1/30
set interfaces fe-0/0/2 unit 0 family inet address 3.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 60.60.60.1/24
set interfaces st0 unit 0 family inet address 10.10.10.2/24
```



```
set interfaces st0 unit 1 family inet address 20.20.20.2/24
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 20.20.20.2
set protocols bgp group ibgp-2 export default_route
set protocols bgp group ibgp-2 neighbor 20.20.20.1
set routing-options static route 1.1.1.0/30 next-hop 2.2.2.2
set routing-options static route 1.1.2.0/30 next-hop 3.3.3.2
set routing-options static route 0.0.0.0/0 next-hop st0.1
set routing-options autonomous-system 10
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 1.1.1.1
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 1.1.2.1
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 1.1.1.1
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 1.1.2.1
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 2.2.2.1/30
user@host# set fe-0/0/2 unit 0 family inet address 3.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 60.60.60.1/24
user@host# set st0 unit 0 family inet address 10.10.10.2/24
user@host# set st0 unit 1 family inet address 20.20.20.2/24
```

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 20.20.20.2
user@host# set group ibgp-2 export default_route
user@host# set group ibgp-2 neighbor 20.20.20.1

[edit routing-options]
user@host# set static route 1.1.1.0/30 next-hop 2.2.2.2
user@host# set static route 1.1.2.0/30 next-hop 3.3.3.2
user@host# set static route 0.0.0.0/0 next-hop st0.1
user@host# set autonomous-system 10
```


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 1.1.1.1
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 1.1.2.1
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 1.1.1.1
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 1.1.2.1
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 2.2.2.1/30;
    }
  }
}
  fe-0/0/2 {
    unit 0 {
```



```
            family inet {
                address 3.3.3.1/30;
            }
        }
    }
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 60.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            family inet {
                address 10.10.10.2/24;
            }
        }
        unit 1 {
            family inet {
                address 20.20.20.2/24;
            }
        }
    }
[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 20.20.20.2;
        export default_route;
        neighbor 20.20.20.1;
    }
}
[edit]
user@host# show routing-options
static {
```

```
    route 1.1.1.0/30 next-hop 2.2.2.2;
    route 1.1.2.0/30 next-hop 3.3.3.2;
    route 0.0.0.0/0 next-hop st0.1;
    }
autonomous-system 10;
[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 1.1.1.1;
        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 1.1.2.1;
        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 1.1.1.1;
    }
    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 1.1.2.1;
    }
    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

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           3.3.3.1

3733076 UP     d53c8a0b7d4c319b  c23c5f7a26388247  Main           2.2.2.1
```

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   2.2.2.1
  >268173316 ESP:des/ md5 1c09b9b 3555/ unlim  U   root 500   2.2.2.1
  <268173313 ESP:des/ md5 7c6ffca3 3340/ unlim U   root 500   3.3.3.1
  >268173313 ESP:des/ md5 33bf6f2f 3340/ unlim U   root 500   3.3.3.1
```

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
20.20.20.2        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               10          5          6       0       0          54
1/1/1/0              0/0/0/0
20.20.20.2               10         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 60.60.60.0/24 is through the st0.0 interface and the default route is through the st0.1 interface.

Action From operational mode, enter the **show route 60.60.60.0** command.

```
user@host> show route 60.60.60.0
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 20.20.20.2 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           3.3.3.1

3733076 UP     d53c8a0b7d4c319b  c23c5f7a26388247  Main           2.2.2.1
```

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   3.3.3.1
  >268173313 ESP:des/ md5 33bf6f2f 3156/ unlim U   root 500   3.3.3.1
```

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
20.20.20.2        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...
20.20.20.2               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 60.60.60.0/24 and the default route are through the st0.1 interface.

Action From operational mode, enter the **show route 60.60.60.0** 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 20.20.20.2 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 20.20.20.2 via st0.1
```

Related Documentation

- Example: Configuring a Route-Based VPN on page 50

CHAPTER 23

Configuring Auto Discovery VPNs

Understanding Auto Discovery VPN

Supported Platforms SRX Series, vSRX

AutoVPN deployments can use the Auto Discovery VPN (ADVPN) protocol to dynamically establish spoke-to-spoke VPN tunnels. When passing traffic from one spoke to another spoke, the hub can suggest that the spokes establish a direct security association (SA), called a shortcut, between each other. Shortcuts can be established and torn down dynamically between spokes, resulting in better network resource utilization and less reliance on a centrally located hub.

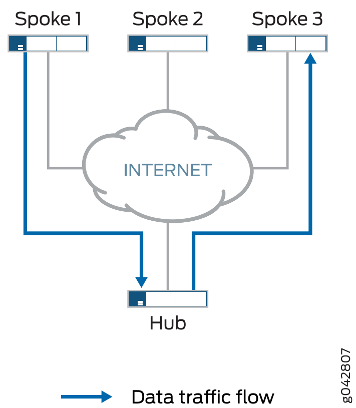
ADVPN Protocol

The ADVPN protocol is an extension of IKEv2 that allows a shortcut to be created between two VPN peers. Devices that support the ADVPN protocol send an ADVPN_SUPPORTED notification in the IKEv2 Notify payload during the initial IKE exchange. A device that supports ADVPN can act as either a shortcut suggester or a shortcut partner, but not both. This shortcut capability information, along with the ADVPN version number, is also exchanged.

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 49 on page 576 shows traffic from Spoke 1 to Spoke 3 passing through the hub.

Figure 49: Spoke-to-Spoke Traffic Passing Through Hub

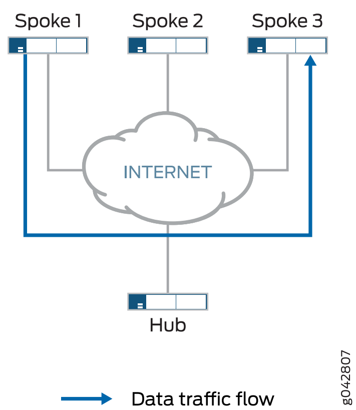


When ADVPN is configured on the devices, ADVPN shortcut capability information is exchanged between the hub and spokes. As long as Spokes 1 and 3 have previously advertised ADVPN shortcut partner capability to the hub, the hub can suggest that Spokes 1 and 3 establish a shortcut between each other.

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 50 on page 577 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 50: 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 74 on page 577 shows the parameters of the shortcut.

Table 74: Shortcut Parameters

Attributes	Received/Inherited From
ADVPN	Configuration

Table 74: Shortcut Parameters *(continued)*

Attributes	Received/Inherited From
Antireplay	Configuration
Authentication algorithm	Configuration
Dead peer detection	Configuration
DF bit	Configuration
Encryption algorithm	Configuration
Establish tunnels	Suggester
External interface	Configuration
Gateway policy	Configuration
General IKE ID	Configuration
IKE version	Configuration
Install interval	Configuration
Local address	Configuration
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 900 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.
- IPv6 and multicast traffic are not supported.

The following configurations are not supported with ADVPN:

- IKEv1
- Policy-based VPN
- IKEv2 configuration payload
- Traffic selectors
- Preshared key
- Point-to-point secure tunnel interfaces

Related Documentation

- Understanding Traffic Routing with Shortcut Tunnels on page 580
- Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels on page 582
- Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established on page 621

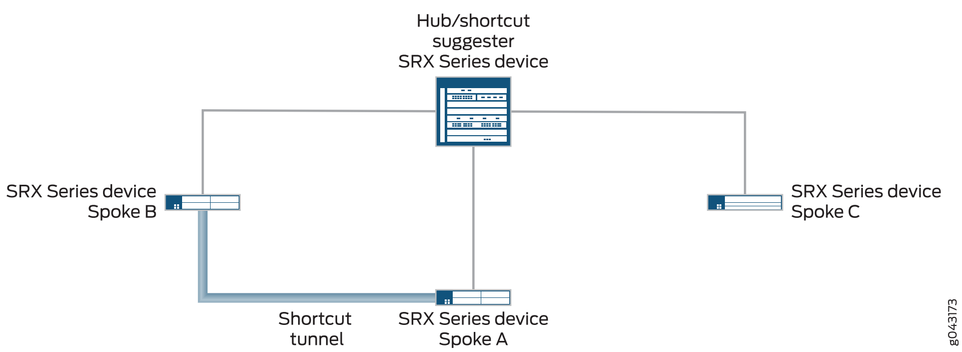
Understanding Traffic Routing with Shortcut Tunnels

Supported Platforms SRX Series, vSRX

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.

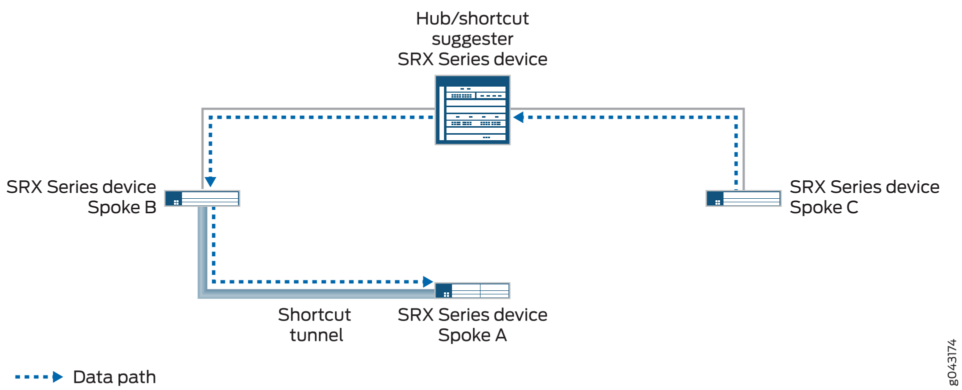
In Figure 51 on page 580, static tunnels exist between the hub and each of the spokes. OSPF adjacencies are established between the hub and spokes. Spoke A also has a shortcut tunnel with Spoke B and OSPF adjacencies are established between the spokes. The hub (the shortcut suggester) recognizes that 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 51: Static Tunnels and Shortcut Tunnel Established in Hub-and-Spoke Network



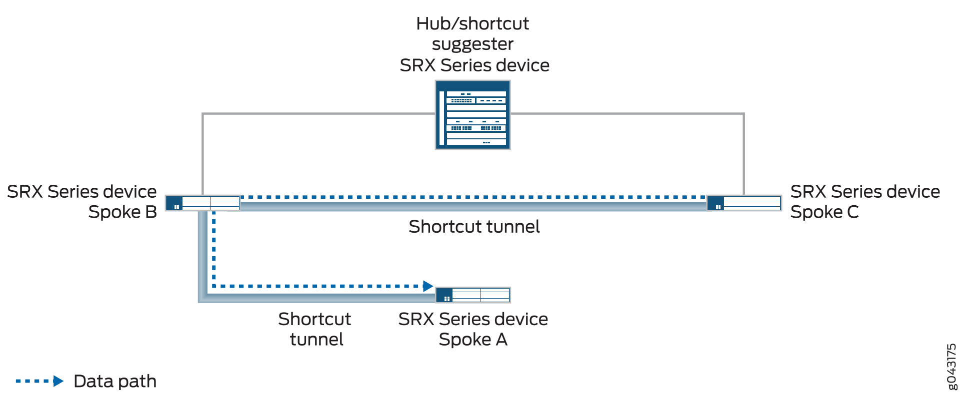
In Figure 52 on page 581, the static tunnel between the hub and Spoke A is down. If there is new traffic from Spoke C to Spoke A, Spoke C forwards the traffic to the hub because it does not have a shortcut tunnel with Spoke A. The hub does not have an active static tunnel with Spoke A but it recognizes that there is a shortcut tunnel between Spoke A and Spoke B, so it forwards the traffic from Spoke C to Spoke B.

Figure 52: Traffic Path from Spoke C to Spoke A



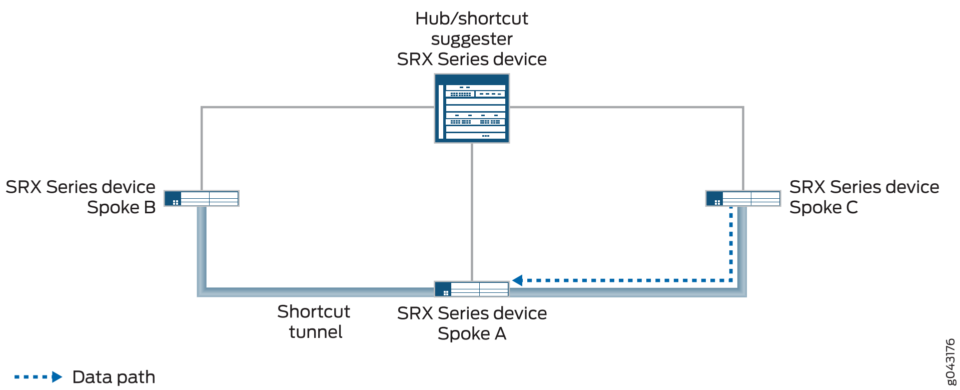
As long as both Spoke B and Spoke C support Auto Discovery VPN (ADVPN) partner capability, the hub can suggest that the spokes establish a direct shortcut between each other. This occurs even though there is no direct traffic between the two spokes. Traffic from Spoke C to Spoke A travels through the shortcut tunnel between Spoke C and Spoke B, and then through the shortcut tunnel between Spoke B and Spoke A (see Figure 53 on page 581).

Figure 53: 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 54 on page 582). 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 54: Traffic Path from Spoke C to Spoke A Through Shortcut Tunnel



NOTE: 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.

Related Documentation

- Understanding Auto Discovery VPN on page 575

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

Supported Platforms SRX Series, vSRX

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 devices 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 377.

2. Enroll the digital certificates in each device. See “Understanding Certificate Loading” on page 384.

NOTE: **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

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 75 on page 583 shows the values used in this example.

Table 75: 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

Table 75: Phase 1 and Phase 2 Options for AutoVPN Hub and Spokes for ADVPN Example *(continued)*

Option	Value
IKE gateway:	
Version	v2-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-256-cbc
IPsec policy:	
Perfect Forward Secrecy (PFS) group	group5

The IKE gateway configuration on the hub and spokes include remote and local values that identify VPN peers. Table 76 on page 584 shows the IKE gateway configuration for the hub and spokes in this example.

Table 76: IKE Gateway Configuration for ADVPN Example

Option	Hub	Spokes
Remote IP address	Dynamic	Spoke 1: 11.1.1.1 Spoke 2: 11.1.1.1
Local IP address	11.1.1.1	Spoke 1: 21.1.1.2 Spoke 2: 31.1.1.2
Remote IKE ID	Distinguished name (DN) with the string “XYZ” in the organization (O) field and “Sales” in the organization unit (OU) field in the spokes’ certificates	DN with the string “Sales” in the OU field in the hub’s certificate
Local IKE ID	DN on the hub’s certificate	DN on the spokes’ certificate

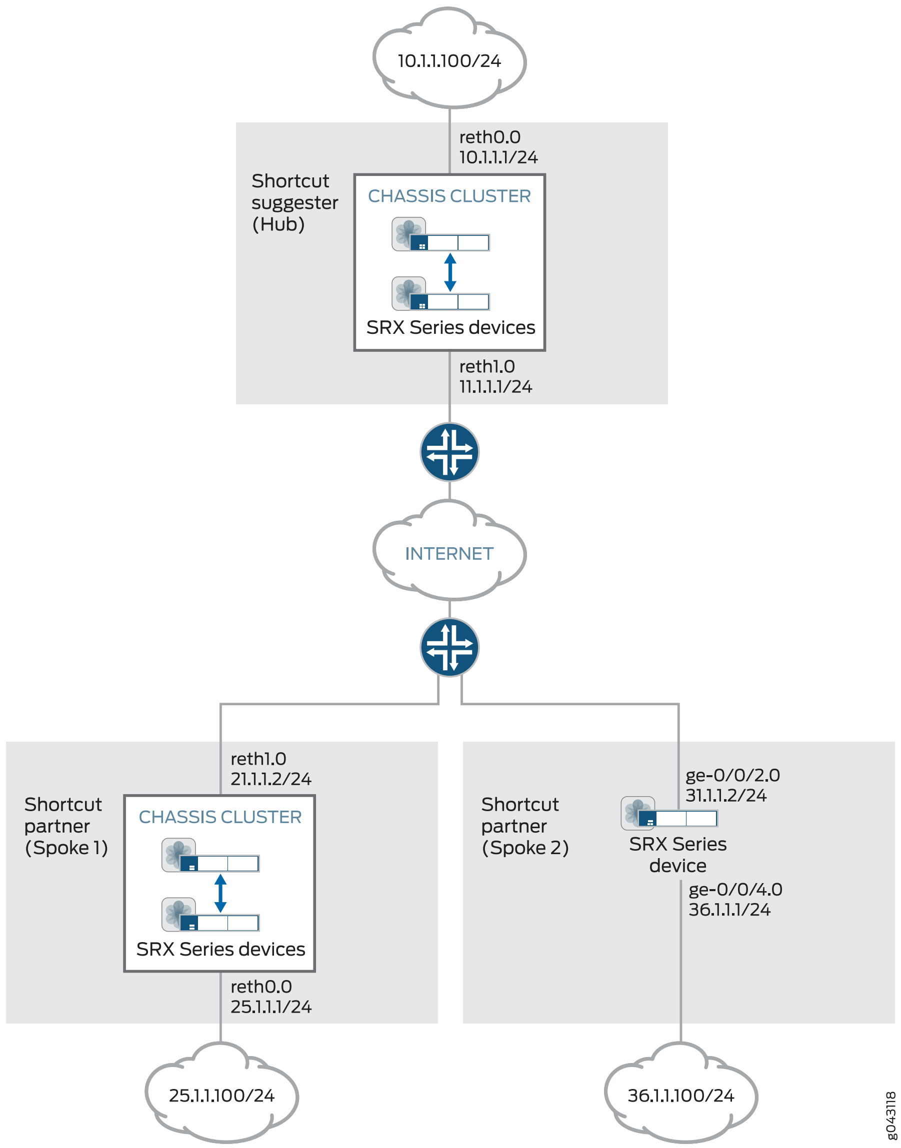
The hub authenticates the spokes’ IKE ID if the subject fields of the spokes’ certificates contain the string “XYZ” in the O field and “Sales” in the OU field.

NOTE: 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 55 on page 585 shows the SRX Series devices to be configured for this example.

Figure 55: AutoVPN Deployment with ADVPN



Configuration

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 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 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;
        }
    }
    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

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.

Related Documentation

- Understanding Auto Discovery VPN on page 575
- Understanding Traffic Routing with Shortcut Tunnels on page 580
- Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established on page 621

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

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- Understanding Auto Discovery VPN on page 575
- Understanding Traffic Routing with Shortcut Tunnels on page 580
- Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels on page 582

CHAPTER 24

Configuring AutoVPN and Traffic Selectors

Understanding AutoVPN with Traffic Selectors

Supported Platforms SRX Series, vSRX

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 devices.
- A single peer can establish multiple tunnels with the same VPN.
- A larger number of tunnels can be supported than with AutoVPN with dynamic routing protocols.

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.

NOTE: **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.
- IPv6 addresses cannot be configured for traffic selectors on AutoVPN hubs. Only IPv4-in-IPv4 tunnel encapsulation is supported for traffic selectors on AutoVPN hubs; IPv4-in-IPv6, IPv6-in-IPv4, and IPv6-in-IPv6 tunnels are not supported.
- Auto Discovery VPN and IKEv2 configuration payload cannot be configured with AutoVPN with traffic selectors.
- Spokes can be non-SRX Series devices; 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 devices 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 devices and the negotiation is rejected.

Related Documentation

- Understanding Traffic Selectors in Route-Based VPNs on page 179
- Understanding Auto Route Insertion on page 198
- Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors on page 640
- Example: Configuring Traffic Selectors in a Route-Based VPN on page 182

Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors

Supported Platforms SRX Series, vSRX

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.

- Requirements on page 624
- Overview on page 625
- Configuration on page 627
- Verification on page 637

Requirements

This example uses the following hardware and software components:

- Two SRX Series devices connected and configured in a chassis cluster. The chassis cluster is the AutoVPN hub.
- An SRX Series device 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 377.
- Enroll the digital certificates in each device. See "Understanding Certificate Loading" on page 384.

Overview

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.

NOTE: **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 77 on page 625 shows the values used in this example:

Table 77: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors

Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5
Authentication algorithm	sha-1
Encryption algorithm	aes-256-cbc
IKE policy:	
Mode	main
Certificate	local-certificate
IKE gateway:	

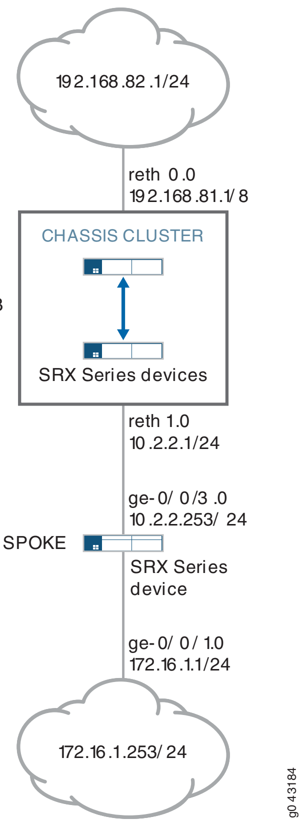
Table 77: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors *(continued)*

Option	Value
Dynamic	distinguished name wildcard DC=Common_component
IKE user type	group IKE id
Local identity	distinguished name
Version	v1-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-192-cbc
Lifetime	3600 seconds 150,000 kilobytes
IPsec policy:	
Perfect Forward Secrecy (PFS) group	group5

Topology

Figure 56 on page 627 shows the SRX Series devices to be configured for this example.

Figure 56: AutoVPN with Traffic Selectors



Configuration

- Configuring the Hub on page 627
- Configuring the Spoke on page 633

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
```

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

Confirm that the configuration is working properly.

- Verifying Tunnels on page 637
- Verifying Traffic Selectors on page 639

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 (also known as a proxy ID in IKEv1) 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).

Related Documentation

- Understanding AutoVPN with Traffic Selectors on page 623
- Understanding Traffic Selectors in Route-Based VPNs on page 179

Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors

Supported Platforms SRX Series, vSRX

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 devices. 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 devices 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.

Requirements

This example uses the following hardware and software components:

- Two SRX Series devices connected and configured in a chassis cluster. The chassis cluster is AutoVPN hub A.
- An SRX Series device 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 377.
- Enroll the digital certificates in each device. See "Understanding Certificate Loading" on page 384.

NOTE: **This example uses the BGP dynamic routing protocol to advertise routes toward the eNodeB devices to the core network.**

Overview

In this example, two AutoVPN hubs are configured with multiple traffic selectors on SRX Series devices 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 78 on page 642 shows the values used in this example:

Table 78: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs

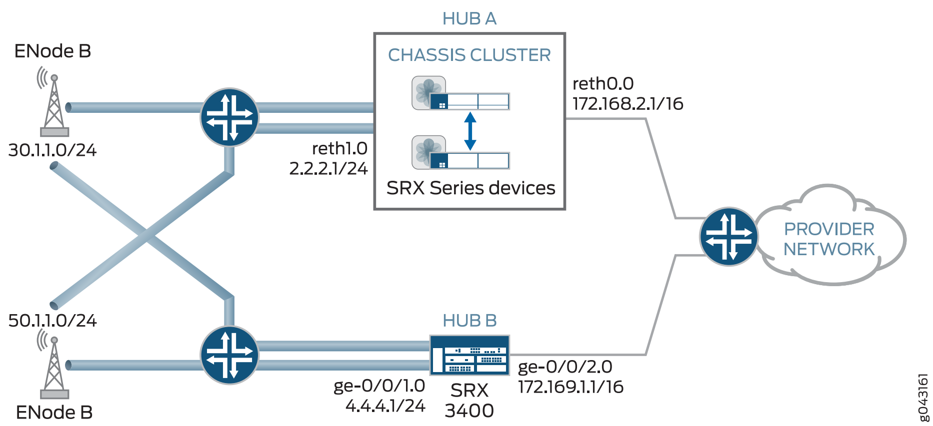
Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5
Authentication algorithm	sha-1
Encryption algorithm	aes-256-cbc
IKE policy:	
Certificate	local-certificate
IKE gateway:	
Dynamic	distinguished name wildcard DC=Common_component
IKE user type	group IKE id
Dead peer detection	probe-idle-tunnel
Local identity	distinguished name
Version	v2-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-256-cbc
IPsec policy:	
Perfect Forward Secrecy (PFS) group	group5

NOTE: 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 devices allows all types of inbound traffic; this configuration is not recommended for production deployments.

Topology

Figure 57 on page 643 shows the SRX Series devices to be configured for this example.

Figure 57: Georedundant IPsec VPN Gateways to eNodeB Devices



Configuration

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 100.100.1.100/24
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 172.168.2.1/16
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 2.2.2.1/24
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 172.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 50.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts2 local-ip 172.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts2 remote-ip 30.0.0.0/8
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 172.168.2.1
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 172.168.2.4
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
  30.1.2.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
  30.1.1.0/24 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
  50.1.1.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow from route-filter
  50.1.2.0/24 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
  172.168.1.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow from route-filter
  172.168.2.0/24 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 100.100.1.100/24
   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 172.168.2.1/16
   user@host# set reth1 redundant-ether-options redundancy-group 1
   user@host# set reth1 unit 0 family inet address 2.2.2.1/24
   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 172.0.0.0/8
user@host# set traffic-selector ts1 remote-ip 50.0.0.0/8
user@host# set traffic-selector ts2 local-ip 172.0.0.0/8
user@host# set traffic-selector ts2 remote-ip 30.0.0.0/8
```

4. Configure the BGP routing protocol.

```
[edit protocols bgp group internal-peers]
user@host# set type internal
user@host# set local-address 172.168.2.1
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 172.168.2.4
```

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 30.1.2.0/24 orlonger
user@host# set term cp_allow from route-filter 30.1.1.0/24 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 50.1.1.0/24 orlonger
user@host# set term mp_allow from route-filter 50.1.2.0/24 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 172.168.1.0/24 orlonger
user@host# set term up_allow from route-filter 172.168.2.0/24 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 100.100.1.100/24;
      }
    }
    redundant-pseudo-interface-options {
      redundancy-group 1;
    }
  }
  reth0 {
    redundant-ether-options {
```

```
        redundancy-group 1;
      }
      unit 0 {
        family inet {
          address 172.168.2.1/16;
        }
      }
    }
    reth1 {
      redundant-ether-options {
        redundancy-group 1;
      }
      unit 0 {
        family inet {
          address 2.2.2.1/24;
        }
      }
    }
    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 172.0.0.0/8;
            remote-ip 50.0.0.0/8;
        }
        traffic-selector ts2 {
            local-ip 172.0.0.0/8;
            remote-ip 30.0.0.0/8;
        }
    }
[edit]
user@host# show protocols bgp
    group internal-peers {
        type internal;
        local-address 172.168.2.1;
        export [ inject_ts1_routes inject_ts2_routes inject_up_routes ];
        neighbor 172.168.2.4;
    }
[edit]
user@host# show policy-options
policy-statement inject_ts1_routes {
    term cp_allow {
        from {
            protocol static;
            route-filter 30.1.2.0/24 orlonger;
            route-filter 30.1.1.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_ts2_routes {
    term mp_allow {
        from {
            protocol static;
            route-filter 50.1.1.0/24 orlonger;
            route-filter 50.1.2.0/24 orlonger;
        }
        then {
            next-hop self;
```

```
            accept;
        }
    }
}
policy-statement inject_up_routes {
    term up_allow {
        from {
            protocol static;
            route-filter 172.168.1.0/24 orlonger;
            route-filter 172.168.2.0/24 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 4.4.4.1/24
set interfaces ge-0/0/2 unit 0 family inet address 172.169.1.1/16
set interfaces lo0 unit 0 family inet address 100.100.1.101/24
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 172.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 50.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts2 local-ip 172.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts2 remote-ip 30.0.0.0/8
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 172.169.1.1
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
  30.1.2.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
  30.1.1.0/24 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
  50.1.1.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow from route-filter
  50.1.2.0/24 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
  172.169.1.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow from route-filter
  172.169.2.0/24 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 4.4.4.1/24
user@host# set ge-0/0/2 unit 0 family inet address 172.169.1.1/16
user@host# set lo0 unit 0 family inet address 100.100.1.101/24
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 172.0.0.0/8
user@host# set traffic-selector ts1 remote-ip 50.0.0.0/8
user@host# set traffic-selector ts2 local-ip 172.0.0.0/8
user@host# set traffic-selector ts2 remote-ip 30.0.0.0/8
```

4. Configure the BGP routing protocol.

```
[edit protocols bgp group internal-peers]
user@host# set type internal
user@host# set local-address 172.169.1.1
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 172.169.1.2
```

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 30.1.2.0/24 orlonger
user@host# set term cp_allow from route-filter 30.1.1.0/24 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 50.1.1.0/24 orlonger
user@host# set term mp_allow from route-filter 50.1.2.0/24 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 172.169.1.0/24 orlonger
user@host# set term up_allow from route-filter 172.169.2.0/24 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 4.4.4.1/24;
      }
    }
  }
  ge-0/0/2 {
    unit 0 {
      family inet {
        address 172.169.1.1/16;
      }
    }
  }
  lo0 {
```



```
      unit 0 {
        family inet {
          address 100.100.1.101/24;
        }
      }
    }
    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 172.0.0.0/8;
            remote-ip 50.0.0.0/8;
        }
        traffic-selector ts2 {
            local-ip 172.0.0.0/8;
            remote-ip 30.0.0.0/8;
        }
    }
[edit]
user@host# show protocols bgp
    group internal-peers {
        type internal;
        local-address 172.169.1.1;
        export [ inject_ts1_routes inject_ts2_routes inject_up_routes ];
        neighbor 172.169.1.2;
    }
user@host# show policy-options
policy-statement inject_ts1_routes {
    term cp_allow {
        from {
            protocol static;
            route-filter 30.1.2.0/24 orlonger;
            route-filter 30.1.1.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_ts2_routes {
    term mp_allow {
        from {
            protocol static;
            route-filter 50.1.1.0/24 orlonger;
            route-filter 50.1.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_up_routes {
    term up_allow {
        from {
            protocol static;
            route-filter 172.169.1.0/24 orlonger;
            route-filter 172.169.2.0/24 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

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=5.5.5.1 # self if
        leftsubnet=30.1.1.0/24 # left subnet
        leftid="CN=fight02, DC=Common_component, OU=Dept, O=Company, L=City,
 ST=CA, C=US " # self id
        right=2.2.2.1 # peer if
        rightsubnet=80.1.1.0/24 # 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=5.5.5.1 # self if
```

```
        leftsubnet=30.1.1.0/24 # self net for proxy id
        leftid="CN=fight02, DC=Common_component, OU=Dept, O=Company, L=City,
ST=CA, C=US " # self id
        right=4.4.4.1 # peer if
        rightsubnet=80.1.1.0/24 # 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

Confirm that the configuration is working properly.

- Verifying Tunnels on the AutoVPN Hubs on page 659
- Verifying Traffic Selectors on page 660
- Verifying ARI Routes on page 660

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          5.5.5.1

1350247532 UP   d5f0cb3a3b18cb92  91269f05527217a0  IKEv2          1.1.1.1

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  1.1.1.1
  >77594626 ESP:aes-cbc-192/sha1 c930a858 3600/  64 - root 500 1.1.1.1
  <69206018 ESP:aes-cbc-192/sha1 2b437fc 3600/  64 - root 500  5.5.5.1
  >69206018 ESP:aes-cbc-192/sha1 c6e02755 3600/  64 - root 500 5.5.5.1
```

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
 80.1.1.0-80.1.1.255              30.1.1.0-30.1.1.255              st0.1
69206018     DC=Common_component, CN=enodebA, OU=Dept, O=Company, L=City, ST=CA,
 C=US
 80.1.1.0-80.1.1.255              50.1.1.0-50.1.1.255              st0.1
77594626     DC=Common_component, CN=enodebB, OU=Dept, O=Company, L=City, ST=CA,
 C=US
```

Meaning A traffic selector (also known as a proxy ID in IKEv1) 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

1.1.0.0/16          *[Static/5] 02:57:57
                     > to 2.2.2.253 via reth1.0
2.2.2.0/24          *[Direct/0] 02:58:43
                     > via reth1.0
2.2.2.1/32          *[Local/0] 02:59:25
                       Local via reth1.0
5.5.0.0/16          *[Static/5] 02:57:57
                     > to 2.2.2.253 via reth1.0
```



```
10.0.0.0/8         *[Static/5] 21:54:52
                    > to 10.157.64.1 via fxp0.0
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
30.1.1.0/24        *[Static/5] 02:28:10  [ARI route added based on TSi]
                    > via st0.1
50.1.1.0/24        *[Static/5] 02:28:26
                    > via st0.1
66.129.230.0/24    *[Static/5] 21:54:52
                    > to 10.157.64.1 via fxp0.0
66.129.236.0/24    *[Static/5] 21:54:52
                    > to 10.157.64.1 via fxp0.0
80.0.0.0/8         *[Direct/0] 02:57:57
                    > via reth0.0
80.1.1.1/32        *[Local/0] 02:57:57
                      Local via reth0.0
100.100.1.0/24     *[Direct/0] 02:57:57
                    > via lo0.0
100.100.1.100/32   *[Local/0] 02:57:57
                      Local via lo0.0
102.100.1.0/24     *[Static/5] 02:57:57
                    > to 2.2.2.253 via reth1.0
104.100.1.0/24     *[Static/5] 02:57:57
                    > to 2.2.2.253 via reth1.0
172.16.0.0/12      *[Static/5] 21:54:52
                    > to 10.157.64.1 via fxp0.0
192.168.0.0/16     *[Static/5] 21:54:52
                    > to 10.157.64.1 via fxp0.0
207.17.136.0/24    *[Static/5] 21:54:52
                    > to 10.157.64.1 via fxp0.0
207.17.137.227/32  *[Static/5] 21:54:52
                    > to 10.157.64.1 via fxp0.0
```

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 30.1.1.0/24 and 50.1.1.0/24 are added to the routing table on the SRX Series hub. These routes are reachable through the st0.1 interface.

Related Documentation

- Understanding AutoVPN with Traffic Selectors on page 623
- Understanding Traffic Selectors in Route-Based VPNs on page 179
- Understanding Auto Route Insertion on page 198

PART 9

Configuring Group VPNs

CHAPTER 25

Configuring Group VPNv2

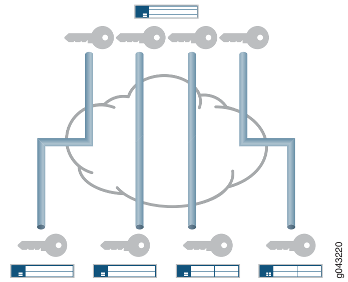
Managing IPsec SA By Configuring Group VPNv2

Group VPNv2 Overview

Supported Platforms SRX Series, vSRX

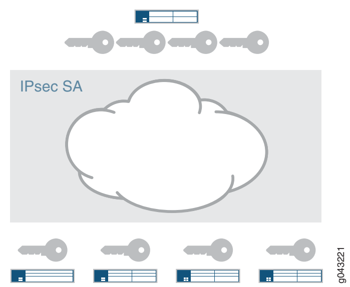
An IPsec security association (SA) is a unidirectional agreement between virtual private network (VPN) participants that defines the rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications. With many VPN implementations, the SA is a point-to-point tunnel between two security devices (see Figure 58 on page 666).

Figure 58: Point-to-Point SAs



Group VPNv2 extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 59 on page 666). 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, and SRX4200 devices and vSRX instances.

Figure 59: Shared SAs



NOTE: Group VPNv2 is an enhanced version of the group VPN feature introduced in an earlier Junos OS release for SRX Series devices. Group VPNv2 on Juniper devices support RFC 6407, *The Group Domain of Interpretation (GDOI)*, and interoperate with other devices that comply with RFC 6407.

Related Documentation

- Understanding the GDOI Protocol for Group VPNv2 on page 666
- Understanding Group VPNv2 Servers and Members on page 667
- Group VPNv2 Configuration Overview on page 669

Understanding the GDOI Protocol for Group VPNv2

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. Group VPNv2 is based on RFC 6407, *The Group Domain of Interpretation* (GDOI). This RFC describes the protocol between group members and group servers to establish SAs among group members. GDOI messages create, maintain, or delete SAs for a group of devices. The GDOI protocol runs on UDP port 848.

The Internet Security Association and Key Management Protocol (ISAKMP) defines two negotiation phases to establish SAs for an IKE IPsec tunnel. Phase 1 allows two devices to establish an ISAKMP SA for other security protocols, such as GDOI.

With Group VPNv2, Phase 1 ISAKMP SA negotiation is performed between a group server and a group member. The server and member must use the same ISAKMP policy. GDOI exchanges between the server and member establish the SAs that are shared with other group members. A group member does not need to negotiate IPsec with other group members. GDOI exchanges must be protected by ISAKMP Phase 1 SAs.

There are two types of GDOI exchanges:

- The **groupkey-pull** exchange allows a member to request SAs and keys shared by the group from the server. Group members must register with a group server through a **groupkey-pull** exchange.
- The **groupkey-push** exchange is a single rekey message that allows the server to send group SAs and keys to members before existing group SAs expire. Rekey messages are unsolicited messages sent from the server to members.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Servers and Members on page 667
- Understanding Group VPNv2 Key Operations on page 710

Understanding Group VPNv2 Servers and Members

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Server Clusters on page 714
- Understanding the GDOI Protocol for Group VPNv2 on page 666
- Understanding Group VPNv2 Limitations on page 668
- Group VPNv2 Configuration Overview on page 669

Understanding Group VPNv2 Limitations

Supported Platforms SRX Series, vSRX

NOTE: 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, and SRX4200 devices and vSRX 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.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding the GDOI Protocol for Group VPNv2 on page 666
- Understanding Group VPNv2 Servers and Members on page 667

Group VPNv2 Configuration Overview

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. This topic describes the main tasks for configuring Group VPNv2.

NOTE: 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 714.

On the group server(s), configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2" on page 670.

2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 671.

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 "Understanding Group VPNv2 Configuration" on page 672 and "Understanding Group VPNv2 Traffic Steering" on page 672.

On the group member, configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2" on page 670.

2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 671.

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 672.

4. Security policy to allow group VPN traffic between the zones specified in the IPsec policy.

NOTE: Group VPNv2 operation requires a working routing topology that allows client devices to reach their intended sites throughout the network.

Related Documentation

- Understanding Group VPNv2 Servers and Members on page 667
- Understanding Group VPNv2 Server-Member Communication on page 709

Understanding IKE Phase 1 Configuration for Group VPNv2

Supported Platforms SRX Series, vSRX

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, and SRX4200 devices and vSRX 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.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Servers and Members on page 667
- Group VPNv2 Configuration Overview on page 669
- Understanding IPsec SA Configuration for Group VPNv2 on page 671

Understanding IPsec SA Configuration for Group VPNv2

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Servers and Members on page 667
- Group VPNv2 Configuration Overview on page 669
- Understanding IKE Phase 1 Configuration for Group VPNv2 on page 670

Understanding Group VPNv2 Configuration

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. 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 672.
- 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 714.
- Server-member communication—Optional configuration that allows the server to send **groupkey-push** rekey messages to members. See “Understanding Group VPNv2 Server-Member Communication” on page 709.
- Server cluster—Optional configuration that supports group controller/key server (GCKS) redundancy. See “Understanding Group VPNv2 Server Clusters” on page 714.
- Antireplay—Optional configuration that detects packet interception and replay. See “Understanding Group VPNv2 Antireplay” on page 713.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Servers and Members on page 667
- Group VPNv2 Configuration Overview on page 669

Understanding Group VPNv2 Traffic Steering

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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.

NOTE: 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.

NOTE: 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 raffic 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.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Servers and Members on page 667
- Group VPNv2 Configuration Overview on page 669

Example: Configuring a Group VPNv2 Server and Members

Supported Platforms SRX Series, vSRX

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, and SRX4200 devices and vSRX instances.

- Requirements on page 675
- Overview on page 675
- Configuration on page 676
- Verification on page 703

Requirements

The example uses the following hardware and software components:

- A supported SRX Series device or vSRX instance running Junos OS Release 15.1X49-D30 or later that supports Group VPNv2. This SRX Series device or vSRX instance operates as a Group VPNv2 server.
- Two supported SRX Series devices or vSRX 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.

NOTE: **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 example, the Group VPNv2 network consists of a server and four members. Two of the members are SRX Series devices or vSRX 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.

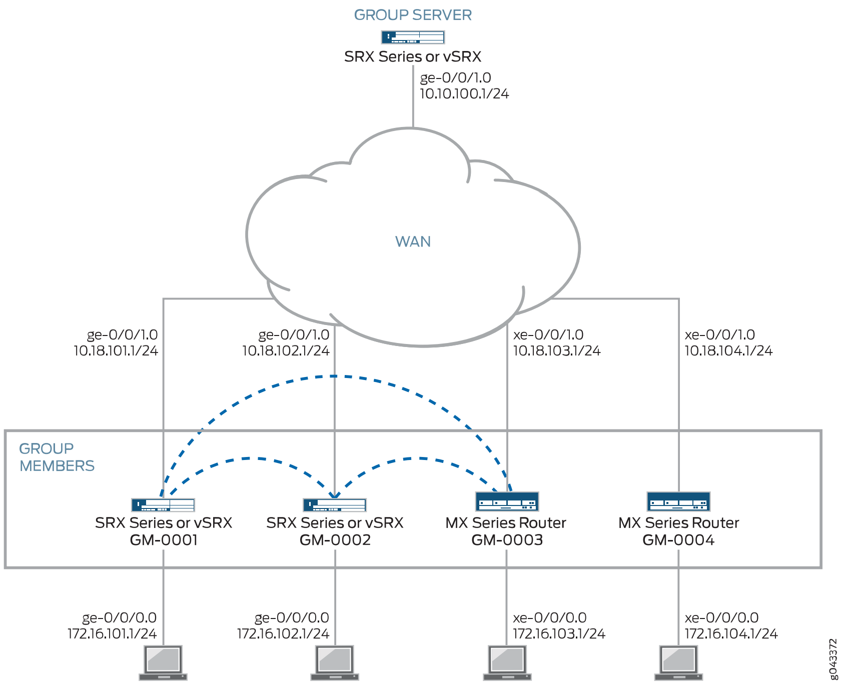
The same group identifier must be configured on both the group server and the group members. In this example, the group name is GROUP_ID-0001 and the group identifier is 1. The group policy configured on the server specifies that the SA and key are applied to traffic between subnetworks in the 172.16.0.0/12 range.

On SRX or vSRX 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 60 on page 676 shows the Juniper Networks devices to be configured for this example.

Figure 60: Group VPNv2 Server with SRX or vSRX and MX Series Members



Configuration

- Configuring the Group Server on page 676
- Configuring Group Member GM-0001 (SRX Series Device or vSRX Instance) on page 682
- Configuring Group Member GM-0002 (SRX Series Device or vSRX Instance) on page 688
- Configuring Group Member GM-0003 (MX Series Device) on page 694
- Configuring Group Member GM-0004 (MX Series Device) on page 698

Configuring the Group Server

CLI Quick Configuration

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

```
set interfaces ge-0/0/1 unit 0 family inet address 10.10.100.1/24
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then reject
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
```

```
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set routing-options static route 10.18.101.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.102.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.103.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.104.0/24 next-hop 10.10.100.254
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256
  authentication-method pre-shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256
  authentication-algorithm sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256
  encryption-algorithm aes-256-cbc
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.10.100.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600
  authentication-algorithm hmac-sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm
  aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds
  3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0005
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication
  communication-type unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication
  encryption-algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication
  lifetime-seconds 7200
set security group-vpn server group GROUP_ID-0001 server-member-communication
  sig-hash-algorithm sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal
  AES256-SHA256-L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001
  match-policy 1 source 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001
  match-policy 1 destination 172.16.0.0/12
```

```
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001
  match-policy 1 protocol 0
```

Step-by-Step Procedure

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

To configure the Group VPNv2 server:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.10.100.1/24

[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0

[edit security policies]
user@host# set global policy 1000 match source-address any
user@host# set global policy 1000 match destination-address any
user@host# set global policy 1000 match application any
user@host# set global policy 1000 match from-zone any
user@host# set global policy 1000 match to-zone any
user@host# set global policy 1000 then reject
user@host# set global policy 1000 then log session-init
user@host# set global policy 1000 then count
user@host# set default-policy deny-all
```

2. Configure the static routes.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.102.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.103.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.104.0/24 next-hop 10.10.100.254
```

3. Configure the IKE proposal, policy, and gateways.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm sha-256
user@host# set dh-group group14
user@host# set encryption-algorithm aes-256-cbc

[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"

[edit security group-vpn server ike gateway GM-0001]
```

```
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.10.100.1

[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.10.100.1

[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.10.100.1

[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.10.100.1
```

4. Configure the IPsec proposal.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600 VPN Group
```

5. Configure the group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
```

6. Configure server-to-member communications.

```
[edit security group-vpn server group GROUP_ID-0001
   server-member-communication]
user@host# set communication-type unicast
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 7200
user@host# set sig-hash-algorithm sha-256
```

7. Configure the group policy to be downloaded to the group members.

```
[edit security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001]
user@host# set proposal AES256-SHA256-L3600
user@host# set match-policy 1 source 172.16.0.0/12
user@host# set match-policy 1 destination 172.16.0.0/12
user@host# set match-policy 1 protocol 0
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, and **show security** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      address 10.10.100.1/24;
    }
  }
}
[edit]
user@host# show routing-options
static {
  route 10.18.101.0/24 next-hop 10.10.100.254;
  route 10.18.102.0/24 next-hop 10.10.100.254;
  route 10.18.103.0/24 next-hop 10.10.100.254;
  route 10.18.104.0/24 next-hop 10.10.100.254;
}
[edit]
user@host# show security
group-vpn {
  server {
    ike {
      proposal PSK-SHA256-DH14-AES256 {
        authentication-method pre-shared-keys;
        authentication-algorithm sha-256;
        dh-group group14;
        encryption-algorithm aes-256-cbc;
      }
      policy GMs {
        mode main;
        proposals PSK-SHA256-DH14-AES256;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
      }
      gateway GM-0001 {
        ike-policy GMs;
        address 10.18.101.1;
        local-address 10.10.100.1;
      }
      gateway GM-0002 {
        ike-policy GMs;
        address 10.18.102.1;
        local-address 10.10.100.1;
      }
      gateway GM-0003 {
        ike-policy GMs;
        address 10.18.103.1;
        local-address 10.10.100.1;
      }
      gateway GM-0004 {
        ike-policy GMs;
        address 10.18.104.1;
```



```
            local-address 10.10.100.1;
          }
        }
        ipsec {
          proposal AES256-SHA256-L3600 {
            authentication-algorithm hmac-sha-256-128;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 3600;
          }
        }
        group GROUP_ID-0001 {
          group-id 1;
          member-threshold 2000;
          ike-gateway GM-0001;
          ike-gateway GM-0002;
          ike-gateway GM-0003;
          ike-gateway GM-0004;
          anti-replay-time-window 1000;
          server-member-communication {
            communication-type unicast;
            lifetime-seconds 7200;
            encryption-algorithm aes-256-cbc;
            sig-hash-algorithm sha-256;
          }
          ipsec-sa GROUP_ID-0001 {
            proposal AES256-SHA256-L3600;
            match-policy 1 {
              source 172.16.0.0/12;
              destination 172.16.0.0/12;
              protocol 0;
            }
          }
        }
      }
    }
    policies {
      global {
        policy 1000 {
          match {
            source-address any;
            destination-address any;
            application any;
            from-zone any;
            to-zone any;
          }
          then {
            reject;
            log {
              session-init;
            }
            count;
          }
        }
      }
      default-policy {
        deny-all;
```

```
    }
  }
  zones {
    security-zone GROUPVPN {
      host-inbound-traffic {
        system-services {
          ike;
          ssh;
          ping;
        }
      }
      interfaces {
        ge-0/0/1.0;
      }
    }
  }
```

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

Configuring Group Member GM-0001 (SRX Series Device or vSRX 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 Device or vSRX 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

Confirm that the configuration is working properly.

- Verifying Group Member Registration on page 703
- Verifying That Group Keys Are Distributed on page 704
- Verifying Group VPN SAs on the Group Server on page 704
- Verifying Group VPN SAs on Group Members on page 705
- Verifying IPsec SAs on the Group Server on page 706
- Verifying IPsec SAs on the Group Members on page 707
- Verifying Group Policies (SRX or vSRX Group Members Only) on page 708

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 or vSRX 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 or vSRX 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 or vSRX Group Members Only)

Purpose Verify group policies on SRX or vSRX 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>
```

Related Documentation

- Group VPNv2 Configuration Overview on page 669
- Configuring Group VPNs in Group VPNv2 on Routing Device

Configuring Group VPNv2 Server-Member Communication

Understanding Group VPNv2 Server-Member Communication

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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. See “Understanding Group VPNv2 Key Operations” on page 710.
- 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.

Related Documentation

- Understanding Group VPNv2 Key Operations on page 710
- Understanding Group VPNv2 Configuration on page 672
- Example: Configuring Group VPNv2 Server-Member Communication for Unicast Rekey Messages on page 711

Understanding Group VPNv2 Key Operations

Supported Platforms SRX Series, vSRX

This topic contains the following sections:

Group Keys

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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 (see "Understanding Group VPNv2 Server-Member Communication" on page 709), 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.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding the GDOI Protocol for Group VPNv2 on page 666
- Understanding Group VPNv2 Servers and Members on page 667
- Group VPNv2 Configuration Overview on page 669

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

Supported Platforms SRX Series, vSRX

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, and SRX4200 devices and vSRX 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

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.

Related Documentation

- Group VPNv2 Configuration Overview on page 669
- Understanding Group VPNv2 Server-Member Communication on page 709

- Understanding Group VPNv2 Key Operations on page 710
- Understanding Group VPNv2 Configuration on page 672

Understanding the Group VPNv2 Recovery Probe Process

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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.

Related Documentation

- Understanding Group VPNv2 Servers and Members on page 667
- Understanding the GDOI Protocol for Group VPNv2 on page 666

Understanding Group VPNv2 Antireplay

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. 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.

Juniper recommends that NTP be configured on all Group VPNv2 devices.

NOTE: Group members that are running on vSRX 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.

Related Documentation

- Understanding Group VPNv2 Servers and Members on page 667
- Understanding Group VPNv2 Configuration on page 672

Configuring Group VPNv2 Server Clusters

Understanding Group VPNv2 Server Clusters

Supported Platforms SRX Series, vSRX

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.

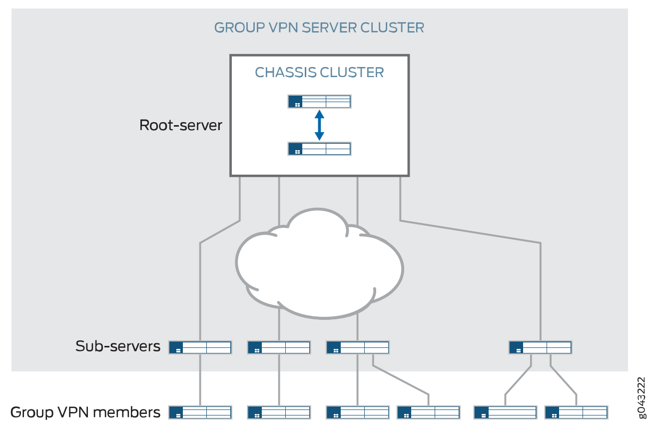
NOTE: Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. All servers in a Group VPNv2 server cluster must be supported SRX Series devices or vSRX instances. Group VPNv2 server clusters are a Juniper Networks proprietary solution and have no interoperability with other vendor's GCKS.

- Dead Peer Detection on page 717
- Load Balancing on page 717

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 61 on page 715.

Figure 61: 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.

NOTE: **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.

NOTE: 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.

NOTE: 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 717.

Group Member Registration with Server Clusters

Group members can register with any server in the Group VPNv2 server cluster for a given group, however we recommend that members only connect to sub-servers and not the root-server. Up to four server addresses can be configured on each group member. The server addresses configured on group members can be different. In the example shown below, group member A is configured for sub-servers 1 through 4, while member B is configured for sub-servers 4 and 3:

	Group member A:	Group member B:
Server addresses:	Sub-server 1	Sub-server 4
	Sub-server 2	Sub-server 3
	Sub-server 3	
	Sub-server 4	

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 717 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 is not 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.

NOTE: DPD is not 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.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Key Operations on page 710
- Understanding Group VPNv2 Server Cluster Messages on page 719
- Example: Configuring a Group VPNv2 Server Cluster and Members on page 725

Understanding Group VPNv2 Server Cluster Limitations

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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.

Related Documentation

- Understanding Group VPNv2 Server Clusters on page 714

Understanding Group VPNv2 Server Cluster Messages

Supported Platforms SRX Series, vSRX

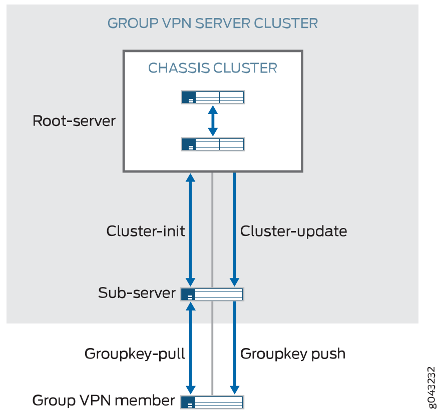
Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. All messages between servers in a Group VPNv2 server cluster are encrypted and authenticated by an IKE security association (SA). Each sub-server initiates an IKE SA with the 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 62 on page 720 shows the basic messages exchanged between the Group VPNv2 server cluster and Group VPNv2 members.

Figure 62: 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.

Related Documentation

- Understanding Group VPNv2 Server Clusters on page 714

Understanding Configuration Changes with Group VPNv2 Server Clusters

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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.

NOTE: 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 79 on page 722 describes the effects of various configuration changes on Group VPNv2 servers.

Table 79: Effects of Configuration Changes on Group VPNv2 Servers

Configuration Change	Standalone Group VPNv2 Server Action	Group VPNv2 Server Cluster Action	
		Root-server	Sub-server
Change IKE proposal, policy, or gateway	Delete the IKE SA for the affected gateway. For IKE proposal, policy, or gateway deletions, delete the registered members for the affected gateway.		
Change IPsec proposal	Changes take effect after the traffic encryption key (TEK) rekey.		
Group changes:			
Delete group name	Send “delete all” to group members. Delete all IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group.	Send “delete all” to sub-servers. Delete all keys in the group immediately. Mark all peers inactive. Delete sub-server IKE SAs. Delete all member IKE SAs.	Delete all member IKE SAs. Delete all keys in the group immediately. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs.
Change ID	Send “delete all” to all members. Delete all IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group. Generate new keys according to the configuration.	Send ”delete all” to sub-servers. Delete all member IKE SAs in the group. Delete all keys in the group immediately. Mark all peers inactive. Delete all peer server IKE SAs. Generate new keys according to the configuration.	Delete all member IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs. Initiate new **cluster-init** exchange.
Add or delete IKE gateway	No changes for additions. For deletions, delete the IKE SA and registered members for the affected gateway.		
Add or change anti-replay time window	New value takes effect after the TEK rekey.		
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 79: 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. 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 80 on page 723 describes the effects of changing Group VPNv2 server cluster configuration.

NOTE: You must ensure that there is only one root-server in a server cluster at any time.

Table 80: Effects of Group VPNv2 Server Cluster Configuration Changes

Server Cluster Configuration Change	Group VPNv2 Server Cluster	
	Root-server	Sub-server
IKE proposal, policy, or gateway (cluster peer)	For additions, there is no change. For changes or deletions, delete the IKE SA for the affected peer.	
Server cluster:		
Add	None.	Send “delete all” to group members. Delete all member IKE SAs in the group. Delete all TEKs and KEKs immediately in the group. Delete all registered members in the group. Send **cluster-init** to root-server.
Change role NOTE: 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.	

Table 80: Effects of Group VPNv2 Server Cluster Configuration Changes *(continued)*

Server Cluster Configuration Change	Group VPNv2 Server Cluster	
	Root-server	Sub-server
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.

Related Documentation

- Understanding Group VPNv2 Server Cluster Messages on page 719
- Understanding Group VPNv2 Key Operations on page 710

Migrating a Standalone Group VPNv2 Server to a Group VPNv2 Server Cluster

Supported Platforms SRX Series, vSRX

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX 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:

NOTE: **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 Feature Guide for SRX Series Devices* for more information

   NOTE: **A reboot is required during the upgrade of a standalone SRX Series device 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.

Related Documentation

- Understanding Group VPNv2 Server Clusters on page 714

Example: Configuring a Group VPNv2 Server Cluster and Members

Supported Platforms SRX Series, vSRX

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, and SRX4200 devices and vSRX instances.

Requirements

The example uses the following hardware and software components:

- Eight supported SRX Series devices or vSRX 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.

NOTE: 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 device or vSRX instance. We recommend that NTP also be configured on each device.

NOTE: The configurations in this example focus on what is needed for Group VPNv2 operation, based on the topology shown in Figure 63 on page 728. 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 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 devices or vSRX 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.

NOTE: 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.

  NOTE: 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.

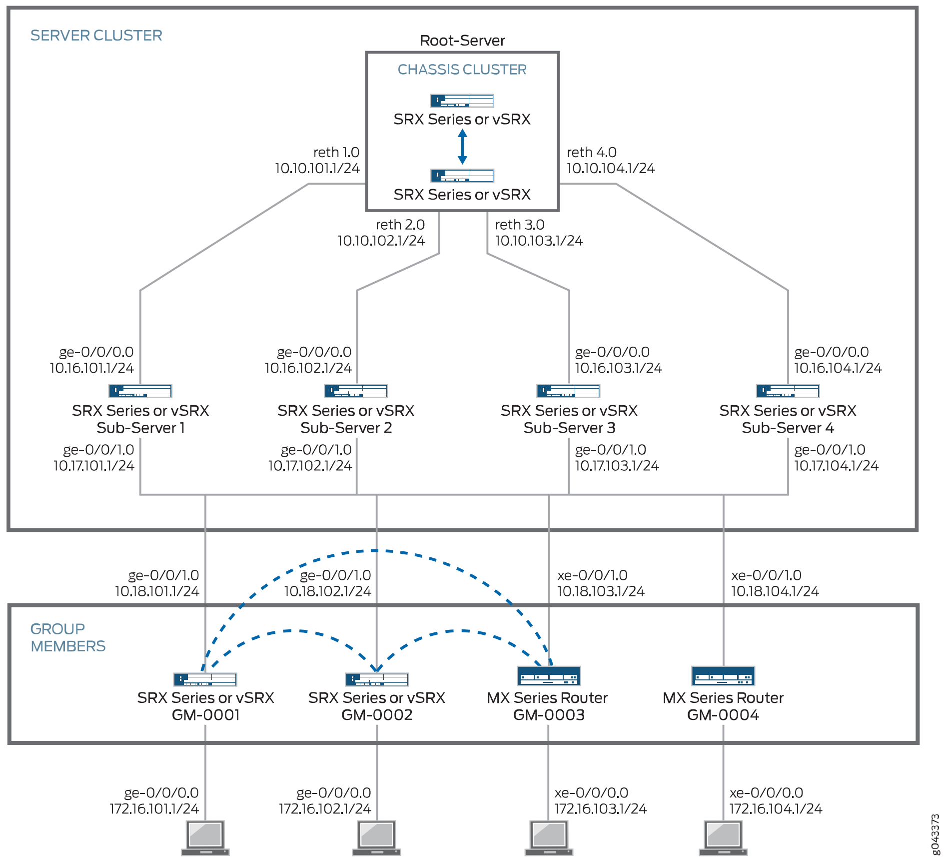
On SRX Series devices or vSRX group members, an IPsec policy is configured for the group with the LAN zone as the from-zone (incoming traffic) and the WAN zone as the to-zone (outgoing traffic). A security policy is also needed to allow traffic between the LAN and WAN zones.

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 63 on page 728 shows the Juniper Networks devices to be configured for this example.

Figure 63: Group VPNv2 Server Cluster with SRX Series or vSRX and MX Series Members



Configuration

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 Device or vSRX 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 Device or vSRX 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

Confirm that the configuration is working properly.

- Verifying Server Cluster Operation on page 782
- Verifying That SAs Are Distributed to Members on page 784
- Verifying IKE SAs on the Servers on page 786

- Verifying IPsec SAs on the Servers and Group Members on page 788
- Verifying IPsec Policies on Group Members on page 790

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 or vSRX 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 or vSRX 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 or vSRX group member.

NOTE: 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 or vSRX 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>
```

Related Documentation

- Group VPNv2 Configuration Overview on page 669
- Understanding Group VPNv2 Server Clusters on page 714
- Configuring Group VPNs in Group VPNv2 on Routing Device

CHAPTER 26

Configuring Group VPNv1

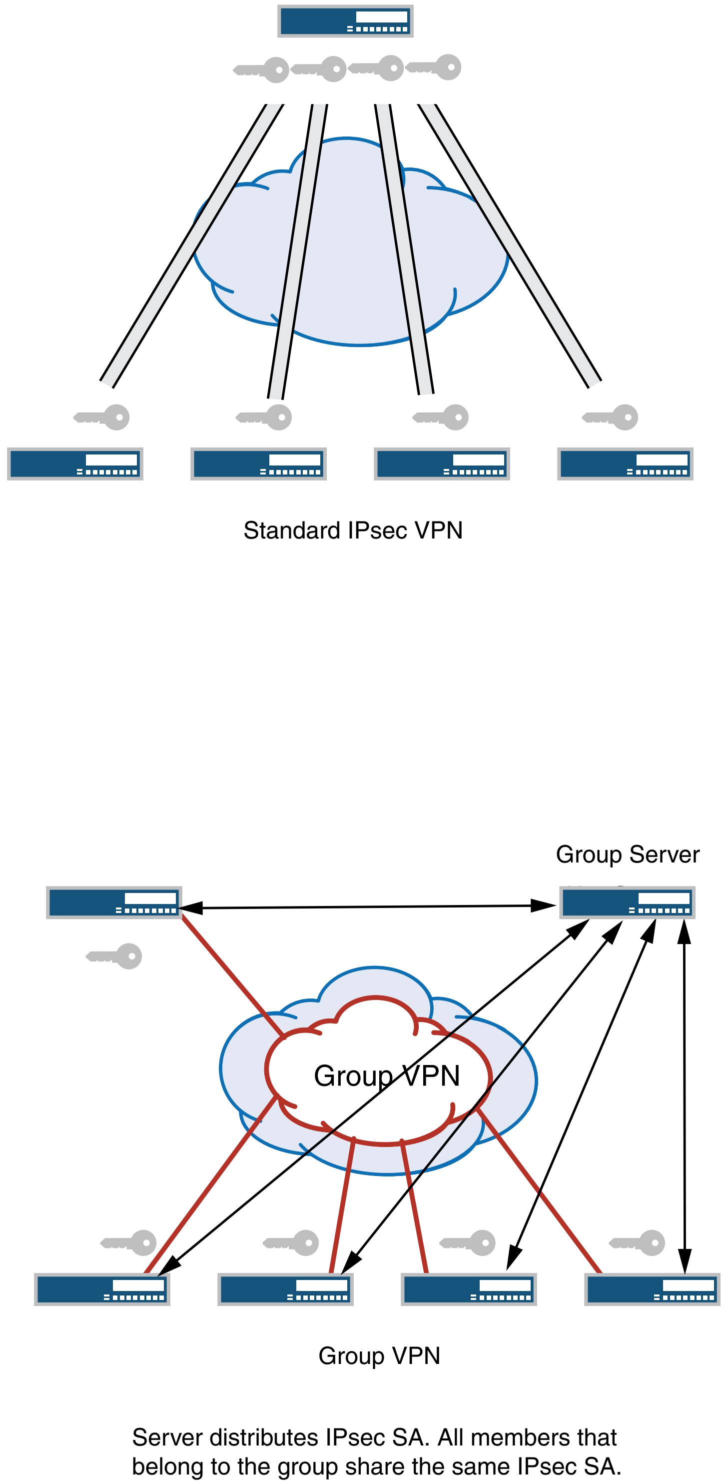
Managing IPsec SA By Configuring Group VPNv1

Group VPNv1 Overview

Supported Platforms SRX Series

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 64 on page 794).

Figure 64: 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.

NOTE: **Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members can interoperate with Group VPNv2 servers.**

NOTE: **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 instances starting with Junos OS Release 15.1X49-D30, SRX Series devices starting with Junos OS Release 15.1X49-D40, and MX Series devices starting with Junos OS Release 15.1r2.**

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.

Related Documentation

- IPsec VPN Overview on page 3
- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Understanding Group VPNv1 Servers and Members on page 820

Understanding the GDOI Protocol for Group VPNv1

Supported Platforms SRX Series

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.

Related Documentation

- Group VPNv1 Overview on page 793
- Understanding IKE and IPsec Packet Processing on page 10
- Understanding Group VPNv1 Servers and Members on page 820
- Understanding Group VPNv1 Group Key Operations on page 821

Understanding IKE Phase 1 Configuration for Group VPNv1

Supported Platforms SRX Series

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.

NOTE: 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.

Related Documentation

- Group VPNv1 Overview on page 793
- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Understanding Group VPNv1 Servers and Members on page 820
- Group VPNv1 Configuration Overview on page 800
- Understanding IPsec SA Configuration for Group VPNv1 on page 797

Understanding IPsec SA Configuration for Group VPNv1

Supported Platforms SRX Series

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.

Related Documentation

- Understanding Group VPNv1 Servers and Members on page 820
- Group VPNv1 Configuration Overview on page 800
- Understanding IKE Phase 1 Configuration for Group VPNv1 on page 796

Understanding Dynamic Policies for Group VPNv1

Supported Platforms SRX Series

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.

NOTE: **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.

  NOTE: **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.

Related Documentation

- *Security Policies Overview*
- *Understanding Security Policy Ordering*
- *Example: Configuring a Security Policy to Permit or Deny All Traffic*
- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Understanding Group VPNv1 Servers and Members on page 820
- Group VPNv1 Configuration Overview on page 800

Understanding Antireplay for Group VPNv1

Supported Platforms SRX Series

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.

Related Documentation

- IPsec VPN Overview on page 3
- Understanding IKE and IPsec Packet Processing on page 10
- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Understanding Group VPNv1 Servers and Members on page 820
- Understanding Group VPNv1 Configuration on page 799

Understanding Group VPNv1 Configuration

Supported Platforms SRX Series

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 797.
- Server-member communication—Optional configuration that allows the server to send rekey messages to members. See “Understanding Group VPNv1 Server-Member Communication” on page 819.
- Antireplay—Optional configuration that detects packet interception and replay. See “Understanding Antireplay for Group VPNv1” on page 799.

Related Documentation

- Understanding Group VPNv1 Servers and Members on page 820
- Group VPNv1 Configuration Overview on page 800

Group VPNv1 Configuration Overview

Supported Platforms SRX Series

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 670.
2. Phase 2 IPsec SA. See “Understanding IPsec SA Configuration for Group VPNv1” on page 797.
3. VPN group. See “Understanding Group VPNv1 Configuration” on page 799.

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 796.
2. Phase 2 IPsec SA. See “Understanding IPsec SA Configuration for Group VPNv1” on page 797.
3. Scope policy that determines which group policies are installed on the member. See “Understanding Dynamic Policies for Group VPNv1” on page 797.

NOTE: 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.

Related Documentation

- Understanding Group VPNv1 Servers and Members on page 820
- Understanding Group VPNv1 Server-Member Communication on page 819
- Example: Configuring Group VPNv1 Server and Members on page 801
- Example: Configuring Group VPNv1 with Server-Member Colocation on page 829

Example: Configuring Group VPNv1 Server and Members

Supported Platforms SRX Series

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

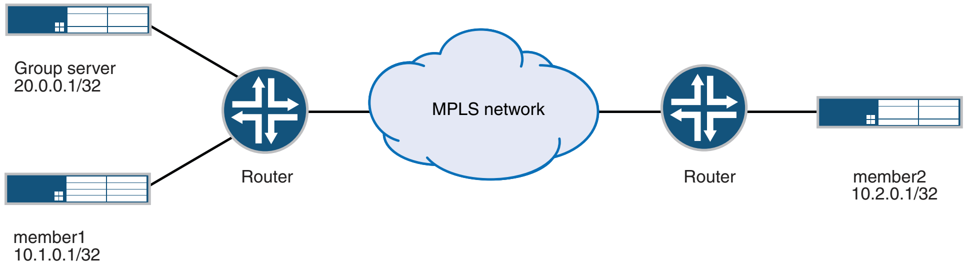
Before you begin:

- Configure the Juniper Networks security devices for network communication.
- Configure network interfaces on server and member devices. See *Interfaces Feature Guide for Security Devices*.

Overview

In Figure 65 on page 802, 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 65: 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

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
```

NOTE: 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
```

NOTE: 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

To confirm that the configuration is working properly, perform this task:

- Verifying Dynamic Policies for Member1 on page 815
- Verifying Dynamic Policies for Member2 on page 816

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.

Related Documentation

- Group VPNv1 Overview on page 793
- Group VPNv1 Configuration Overview on page 800

Understanding Group VPNv1 Limitations

Supported Platforms SRX Series

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.

Release History Table

Release	Description
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

- Group VPNv1 Overview on page 793
- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Understanding Group VPNv1 Servers and Members on page 820

Configuring Group VPNv1 Server-Group Communication

Understanding Group VPNv1 Server-Member Communication

Supported Platforms SRX Series

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. See “Understanding Group VPNv2 Key Operations” on page 710.
- 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. See “Understanding Group VPNv1 Heartbeat Messages” on page 824.
- Lifetime for the key encryption key (KEK). The default is 3600 seconds.

NOTE: **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.

Related Documentation

- Understanding Group VPNv1 Group Key Operations on page 821
- Understanding Group VPNv1 Configuration on page 799
- Example: Configuring Group VPNv1 Server-Member Communication for Unicast Rekey Messages on page 825
- Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages on page 826

Understanding Group VPNv1 Servers and Members

Supported Platforms SRX Series

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.

Related Documentation

- Group VPNv1 Overview on page 793
- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Group VPNv1 Configuration Overview on page 800
- Understanding Group VPNv1 Colocation Mode on page 829
- Understanding Dynamic Policies for Group VPNv1 on page 797

Understanding Group VPNv1 Group Key Operations

Supported Platforms SRX Series

This topic contains the following sections:

- Group Keys on page 821
- Rekey Messages on page 821
- Member Registration on page 823
- Key Activation on page 823

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 (see “Understanding Group VPNv2 Server-Member Communication” on page 709), 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:

- Types of Rekey Messages on page 822
- Rekey Intervals on page 822

Types of Rekey Messages

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.

  NOTE: 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 Feature Guide*.

Rekey Intervals

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.

Related Documentation

- Group VPNv1 Overview on page 793
- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Understanding Group VPNv1 Servers and Members on page 820
- Group VPNv1 Configuration Overview on page 800
- Understanding IPsec SA Configuration for Group VPNv1 on page 797
- Understanding Dynamic Policies for Group VPNv1 on page 797

Understanding Group VPNv1 Heartbeat Messages

Supported Platforms SRX Series

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.

NOTE: Heartbeat messages can increase network congestion and cause unnecessary member reregistrations. Thus, heartbeat detection can be disabled on the member if necessary.

Related Documentation

- Group VPNv1 Overview on page 793

- Understanding the GDOI Protocol for Group VPNv1 on page 795
- Understanding Group VPNv1 Servers and Members on page 820
- Understanding Group VPNv1 Server-Member Communication on page 819

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

Supported Platforms SRX Series

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

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.

Related Documentation

- Group VPNv1 Overview on page 793
- Group VPNv1 Configuration Overview on page 800

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

Supported Platforms SRX Series

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 VPNs* or “Example: Configuring Group VPNv1 with Server-Member Colocation” on page 829.
- 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*.

NOTE: 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

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

To confirm that the configuration is working properly, perform these tasks:

- Verifying Server-Member Communication for Multicast Rekey Messages on page 828

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.

Related Documentation

- Group VPNv1 Overview on page 793
- Group VPNv1 Configuration Overview on page 800

Configuring Group VPNv1 with Server-Member Colocation

Understanding Group VPNv1 Colocation Mode

Supported Platforms SRX Series

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.

Related Documentation

- Example: Configuring Group VPNv1 with Server-Member Colocation on page 829

Example: Configuring Group VPNv1 with Server-Member Colocation

Supported Platforms SRX Series

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:

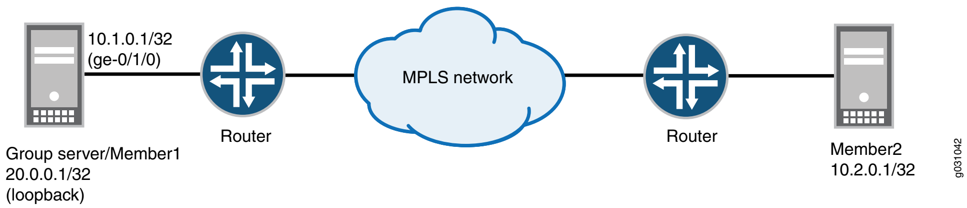
- Configure the Juniper Networks security devices for network communication.
- Configure network interfaces on server and member devices. See *Interfaces Feature Guide for Security Devices*.

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 66 on page 830, 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 66: Server-Member Colocation Example



NOTE: 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 VPNs*.

NOTE: 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

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 "9c1grK8-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.

NOTE: 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

To confirm that the configuration is working properly, perform these tasks:

- Verifying Group VPN Member Registration on page 837
- Verifying Group VPN Server Security Associations for IKE on page 838
- Verifying Group VPN Server Security Associations for IPsec on page 838
- Verifying Group VPN Member Security Associations for IKE on page 838
- Verifying Group VPN Member Security Associations for IPsec on page 838

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.

Related Documentation

- Understanding Group VPNv1 Colocation Mode on page 829

PART 10

Configuring Remote Access VPNs

CHAPTER 27

Configuring Remote Access VPNs with NCP Exclusive Remote Access Client

Understanding IPsec VPNs with NCP Exclusive Remote Access Client

Supported Platforms SRX Series, vSRX

This section describes IPsec VPN support on SRX Series devices for NCP Exclusive Remote Access Client software.

NCP Exclusive Remote Access Client

Users running NCP Exclusive Remote Access Client software on Windows and MAC OS devices can establish IKEv1 or IKEv2 IPsec VPN connections with SRX Series devices. NCP Exclusive Remote Access Client software is available for download at https://www.ncp-e.com/ncp-exclusive-remote-access-client/.

Licensing

A two-user license is supplied by default on an SRX Series device. A license is required for additional users. Contact your Juniper Networks representative for license information.

Licensing is based on the number of users. For example, if the number of licenses installed is for 100 users, then 100 different users can establish VPN connections. Because of traffic selectors, each user can establish multiple tunnels. When a user disconnects, their license is released one minute after the IKE and IPsec security associations (SAs) expire.

License enforcement is verified only after Phase 2 negotiation is completed. This means that a remote access user can connect to the SRX Series device and IKE and IPsec SAs can be established, but if the user exceeds the licensed user limit, the user is disconnected.

Licensing for vSRX instances is subscription-based: connected remote access users are not disconnected immediately when an installed license expires. When a remote access user disconnects and the corresponding IKE and IPsec SAs expire, subsequent reconnection of the user depends on whether the currently installed license is expired or not.

AutoVPN

The NCP Exclusive Remote Access Client is supported with AutoVPN in point-to-point secure tunnel interface mode. AutoVPN is only supported on route-based IPsec VPNs on the SRX Series device.

Traffic Selectors

Traffic selectors configured on the SRX Series device and the NCP client determine the client traffic that is sent through the IPsec VPN tunnel. Traffic in and out of the tunnel is allowed only for the negotiated traffic selectors. If the route lookup for a packet's destination address points to an st0 interface (on which traffic selectors are configured) and the packet's traffic selector does not match the negotiated traffic selector, the packet is dropped. Multiple Phase 2 IPsec SAs and auto route insertion (ARI) are supported with the NCP Exclusive Remote Access Client. Traffic selector flexible match with port and protocols is not supported. For this feature, the remote address of the traffic selector must be 0.0.0.0/0.

In many cases, all traffic from remote access clients is sent through VPN tunnels. The local address configured in the traffic selector can be 0.0.0.0/0 or a specific address, as explained in the next sections.

NOTE: Configuring a traffic selector on the SRX Series device with the remote address 0.0.0.0/0 is only supported for NCP Exclusive Remote Access Client connections. After VPN negotiation is completed, the remote address for the traffic selector is expected to be a single IP address (the address of the remote access client assigned by either a RADIUS server or the local address pool).

Split Tunneling

Split tunneling uses a shorter prefix than 0.0.0.0/0 as the protected resource's address for the local address in a traffic selector configured on the SRX Series device. A corresponding traffic selector can be configured on the remote access client. The SRX Series device allows traffic on the VPN tunnel that matches the results of the flexible match from both traffic selectors. If the traffic selector configured on the remote access client cannot be matched with the traffic selector configured on the SRX Series device, tunnel negotiation fails. For IKEv1, the local and remote addresses in the client's traffic selector configuration must be the same addresses or a subset of the addresses in the corresponding traffic selector configured on the SRX Series device.

Multiple Subnetworks

On the SRX Series device, one traffic selector can be configured for each protected subnetwork. Subnetworks cannot overlap. On the NCP Exclusive Remote Access Client, one traffic selector must be configured for each traffic selector configured on the SRX Series device. Addresses that are configured in the split tunnel window of the NCP Exclusive Remote Access Client are used as the client's remote traffic selector; these addresses must be the same addresses or a subset of the addresses in the corresponding traffic selector configured on the SRX Series device. One IPsec SA pair is created for each traffic selector.

NCP Exclusive Remote Access Client Authentication

There are two forms of extended authentication of the NCP Exclusive Remote Access Client, depending on the IKE version of the client:

- IKEv1 NCP Exclusive Remote Access Client authentication is supported with XAuth using either a RADIUS server or a local access profile. For IKEv1 remote access connections, preshared keys are used for IKE Phase 1 authentication. Extended Authentication (XAuth) is used to authenticate the remote access user. The SRX Series device must be configured for IKE aggressive mode.

  NOTE: **For the IKEv1 NCP Exclusive Remote Access Client, preshared key authentication is supported with AutoVPN. For AutoVPN deployments that do not use user-based authentication, only certificate authentication is supported.**

- IKEv2 NCP Exclusive Remote Access Client authentication requires a RADIUS server that supports EAP. The SRX Series device acts as a pass-through authenticator to relay EAP messages between the NCP Exclusive Remote Access Client and the RADIUS server. The following EAP authentication types are supported:

  - EAP-MSCHAPv2

    NOTE: **A master session key must be generated by the RADIUS server for EAP-MSCHAPv2.**

  - EAP-MD5
  - EAP-TLS

  For the IKEv2 NCP Exclusive Remote Access Client, a digital certificate is used to authenticate the SRX Series device. Extensible Authentication Protocol (EAP) is used to authenticate the remote access client.

Remote Access Client Attribute and IP Address Assignment

Attribute Assignment

For IKEv1 or IKEv2 remote access clients, attributes can be assigned through a RADIUS server or through local network attributes configuration. If a RADIUS server is used for authentication but no network attributes are assigned, network attributes (including IP addresses) can be configured locally if needed.

The following client attributes are based on RFC 2865, *Virtual Private Networks Identifier*, and are supported with IKEv1 and IKEv2 NCP Exclusive Remote Access Client:

- Framed-IP-Address
- Framed-IP-Netmask

The following Juniper vendor-specific attributes (VSAs) are supported with IKEv1 and IKEv2 NCP Exclusive Remote Access Client:

- Juniper-Primary-DNS
- Juniper-Primary-Wins
- Juniper-Secondary-DNS (only available with IKEv2)
- Juniper-Secondary-Wins (only available with IKEv2)

NOTE: The VSA Juniper-Local-Group-Name is not supported.

IP Address Assignment

If an IP address is allocated from both a local address pool and by a RADIUS server, the IP address allocated by the RADIUS server takes precedence. If the RADIUS server does not return an IP address and there is a user-configured local address pool, an IP address is assigned to the remote client from the local pool.

NOTE: The number of addresses in the local address pool or RADIUS server address pool should be larger than the number of remote access client users. This is because when a user disconnects, it can take up to one minute for the user to be logged off.

When an IP address is assigned from an external RADIUS server or a local address pool, an IP address with a 32-bit mask is passed to the NCP Exclusive Remote Access Client. After the tunnel is established, auto route insertion (ARI) automatically inserts a static route to the remote client's IP address so that traffic from behind the SRX Series device can be sent into the VPN tunnel to the client's IP address.

The configured traffic selectors might not cover the IP addresses allocated by the RADIUS server or a local address pool. In this case, a remote client may not be able to reach an IP address for another remote client in the subnetwork through a VPN tunnel. A traffic selector must be explicitly configured that matches the IP address allocated to the other remote client by the RADIUS server or local address pool.

Supported Features

The following features are supported on the SRX Series device with the NCP Exclusive Remote Access Client:

- Traffic initiation from the SRX Series device as well as the NCP Exclusive Remote Access Client
- Remote access clients behind a NAT device (NAT-T)
- Dead peer detection
- Chassis cluster configuration of the SRX Series device

Caveats

The following features are not supported on the SRX Series device with the NCP Exclusive Remote Access Client:

- Routing protocols
- AutoVPN with the st0 interface in point-to-multipoint mode
- Auto Discovery VPN (ADVPN)
- IKEv2 EAP with preshared keys

NOTE: **The IKEv2 NCP Exclusive Remote Access Client must use certificates for authenticating the SRX Series device.**

- Policy-based VPN
- IPv6 traffic
- VPN monitoring
- Next-hop tunnel binding (NHTB), both auto and manual
- Multiple traffic selectors in negotiation
- Traffic selectors received from the NCP Exclusive Remote Access Client in the same virtual router must not contain overlapping IP addresses

Related Documentation

- Understanding Traffic Selectors in Route-Based VPNs on page 179

Understanding SSL Remote Access VPNs with NCP Exclusive Remote Access Client

Supported Platforms SRX Series, vSRX

In many public hotspot environments, UDP traffic is blocked while TCP connections over port 443 are normally allowed. For these environments, SRX Series devices can support

IPsec messages encapsulated within a TCP connection. This implementation is compatible with the third-party NCP Exclusive Remote Access Client. This feature is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices. This section describes the support for NCP Exclusive Remote Access Client on SRX Series devices.

NCP Exclusive Remote Access Client

Users running NCP Exclusive Remote Access Client software on Windows and MAC OS devices can establish TCP connections with SRX Series devices to exchange encapsulated IPsec traffic. NCP Exclusive Remote Access Client software is available for download at https://www.ncp-e.com/ncp-exclusive-remote-access-client/.

Licensing

A two-user license is supplied by default on an SRX Series device. A license must be purchased and installed for additional users. Contact your Juniper Networks representative for license information.

Operation

On an SRX Series device, a *TCP encapsulation profile* defines the data encapsulation operation for remote access clients. Multiple TCP encapsulation profiles can be configured to handle different sets of clients. For each profile, the following information is configured:

- Name of the profile.
- Optional logging of remote access client connections.
- Tracing options.
- Optionally, tracking information about IPsec tunnels can be enabled for troubleshooting.

NOTE: TCP connections from NCP Exclusive Remote Access Client are accepted on port 443 on the SRX Series device.

The TCP encapsulation profile is configured with the **tcp-encap** statement at the [**edit security**] hierarchy level. The encapsulation profile is then specified with the **tcp-encap-profile** statement at the [**edit security ike gateway *gateway-name***] hierarchy level. You include the TCP encapsulation profile in the IKE gateway configuration. For example:

```
user@host# set security tcp-encap profile ncp
user@host# set security ike gateway RA tcp-encap-profile ncp
```


Supported Features

The following features are supported on an SRX Series device with NCP Exclusive Remote Access Client:

- AutoVPN in point-to-point mode with IPsec tunnels based on traffic selectors
- Traffic initiation from devices behind the gateway on an SRX Series device
- Dead peer detection
- Chassis cluster configuration of an SRX Series device

Caveats

TCP connections from NCP Exclusive Remote Access Clients use port 443 on SRX Series devices. Device management on TCP connections, such as J-Web, also use port 443 on SRX Series devices. If the J-Web connection uses a port other than 443, tcp-encap must be configured for host-inbound system services. Use the **set security zones security-zone *zone* host-inbound-traffic system-services tcp-encap** command. (IKE must also be configured for host-inbound system services using the **set security zones security-zone *zone* host-inbound-traffic system-services ike** command.)

To prevent NCP Exclusive Remote Access Client and J-Web connections on port 443, use the **except** option with https and tcp-encap system services. If the J-Web connection uses a port other than 443, use the **set security zones security-zone *zone* host-inbound-traffic system-services tcp-encap except** command to block NCP Exclusive Remote Access Client connections.

If the IKE external interface is disabled then enabled, tunnels that use TCP connections with NCP Exclusive Remote Access Clients may not come up. If this occurs, reduce the TCP timeout for the client connections with the **inactivity-timeout** option at the [**edit applications application *application-name***] hierarchy level. The **destination-port** configured at the [**edit applications application *application-name***] hierarchy level must match the **ports** option configured at the [**edit security tcp-encap profile *profile-name***] hierarchy level. The configuration application must then be specified in the **match application** configuration at the [**edit security policies from-zone *from-zone* to-zone *to-zone* policy *policy-name***] hierarchy level.

Tunnels that use TCP connections might not survive ISSU if the dead peer detection (DPD) timeout is not large enough. If you see this happening, increase the DPD timeout to a value greater than 120 seconds. The DPD timeout is a product of the configured DPD interval and threshold. For example, if the DPD interval is 32 and the threshold is 4, the timeout is 128.

The default DPD settings on the NCP Exclusive Remote Access Client specify sending messages at 20-second intervals for a maximum of eight times. When chassis cluster failover occurs, the SRX Series devices might not recover within the parameters specified by the DPD settings and the tunnel goes down. In this case, increase the DPD interval on the NCP Exclusive Remote Access Client to 60 seconds.

NAT-T is disabled during negotiation with clients, as NAT-T is not required with TCP encapsulation.

The following features are not supported on an SRX Series device with NCP Exclusive Remote Access Clients:

- Routing protocols
- AutoVPN with the st0 interface in point-to-multipoint mode
- Auto Discovery VPN (ADVPN)
- Policy-based VPN
- IPv6 traffic
- VPN monitoring
- Next-hop tunnel binding (NHTB), both auto and manual

Related Documentation

- tcp-encap on page 1098
- tcp-encap-profile on page 1099

CHAPTER 28

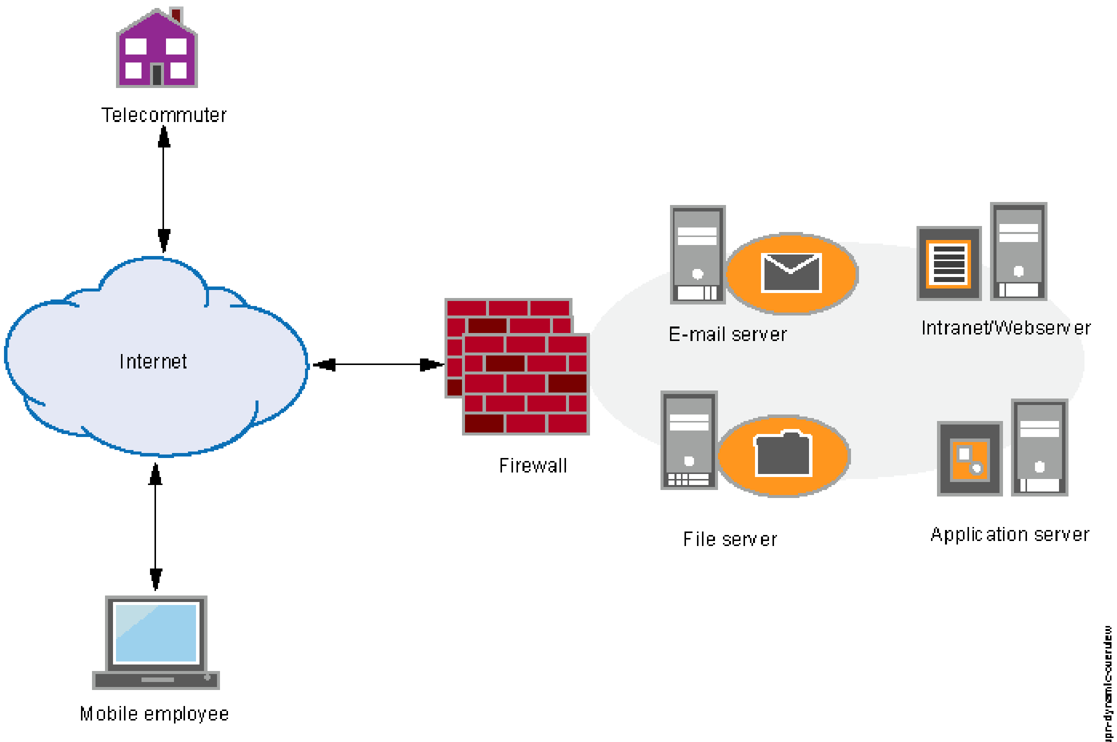
Configuring Dynamic VPNs with Pulse Clients

Dynamic VPN Overview

Supported Platforms SRX Series

Virtual private network (VPN) tunnels enable users to securely access assets such as e-mail servers and application servers that reside behind a firewall. End-to-site VPN tunnels are particularly helpful to remote users such as telecommuters because a single tunnel enables access to all of the resources on a network—the users do not need to configure individual access settings to each application and server. See Figure 67 on page 850.

Figure 67: Using a VPN Tunnel to Enable Remote Access to a Corporate Network



The dynamic VPN feature (also known as remote access VPN or IPsec VPN client) further simplifies remote access by enabling users to establish Internet Protocol Security (IPsec) VPN tunnels without having to manually configure VPN settings on their PCs or laptops. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices. Pulse Secure client software is used for VPN access. User authentication is supported through an external RADIUS server or a local IP address pool configured on the SRX gateway. The Layer 3 remote access client uses client-side configuration settings that it receives from the SRX gateway to create and manage a secure end-to-site VPN tunnel to the gateway.

NOTE: If more than two simultaneous user connections are required, a dynamic VPN license must be installed on the SRX gateway. The dynamic VPN feature is disabled by default on the device. To enable dynamic VPN, you must configure the feature using the **dynamic-vpn** configuration statement at the **[edit security]** hierarchy level. See the *Installation and Upgrade Guide* for information about installing and managing licenses.

Related Documentation

- Dynamic VPN Configuration Overview on page 855
- Understanding Dynamic VPN Tunnel Support on page 851
- Understanding Remote Client Access to the VPN on page 852

Understanding Dynamic VPN Tunnel Support

Supported Platforms SRX Series

Dynamic VPN tunnels are configured in the same way as traditional IPsec VPN tunnels. However, not all IPsec VPN options are supported. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

The following list describes the requirements and supported options when configuring dynamic VPN tunnels:

- Only policy-based VPNs are supported. Route-based VPNs are not supported with dynamic VPN tunnels. Routing protocols are not supported.
- Only IKEv1 is supported. IKEv2 is not supported.
- Only IPv4 traffic and IPv4-in-IPv4 tunnels are supported. IPv6 traffic and tunnels are not supported.
- Only preshared keys are supported for authentication. PKI is not supported.
- Aggressive mode is supported for IKE phase 1 exchanges. Main mode is not supported.
- VPN traffic can only be initiated from the remote client. VPN traffic initiated from the SRX gateway is not supported.
- Dead peer detection (DPD) is supported. VPN monitoring is not supported.
- Extended authentication (XAuth) with mode configuration is supported.
- Authentication is supported from a local profile. Attributes can be provided from a local address pool. Authentication and attributes can be provided from a RADIUS server.
- Chassis clusters are supported.
- NAT-T is supported.
- IKE in virtual routers or in virtual routing and forwarding instances is supported.
- AutoVPN is not supported.
- Auto route insertion (ARI) is not supported.
- Administrator rights are required to install Pulse client software, administrator rights are required.
- Users need to reauthenticate during IKE phase 1 rekeys. The rekey time is configurable.

Shared or group IKE IDs can be used to configure a single VPN that is shared by all remote clients. When a single VPN is shared, the total number of simultaneous connections to the gateway cannot be greater than the number of dynamic VPN licenses installed. When configuring a shared or group IKE ID gateway, you can configure the maximum number of connections to be greater than the number of installed dynamic VPN licenses. However, if a new connection exceeds the number of licensed connections, the connection will be denied. You can view dynamic VPN license information with the **show system license usage** command.

Related Documentation

- Dynamic VPN Overview on page 849
- Dynamic VPN Configuration Overview on page 855
- Example: Configuring Dynamic VPN on page 857
- Understanding IKE and IPsec Packet Processing on page 10

Understanding Remote Client Access to the VPN

Supported Platforms SRX Series

A common dynamic VPN deployment is to provide VPN access to remote clients connected through a public network such as the Internet. IPsec access is provided through a gateway on the Juniper Networks device. Pulse Secure client software is used for VPN access. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

NOTE: Pulse Secure client software can be obtained from the Juniper Networks Download Software site at http://www.juniper.net/support/downloads/?p=pulse#sw.

The following describes the process for a Pulse Secure remote client to access the VPN:

NOTE: For detailed instructions about connecting the remote client program to the SRX Series device, see KB17641. Also see the Pulse Secure documentation for current client information.

1. The user downloads and installs the Pulse Secure client software onto their device.
2. The user starts the Pulse Secure remote client program.

   In the Pulse Secure remote client program, the user does the following:

   a. Click **Add connection**.

   b. For Type, select **Firewall (SRX)**.

   c. For Name, enter the hostname of the SRX gateway.

NOTE: On the SRX Series device, this hostname is configured with the **set security ike gateway *gateway-name* dynamic hostname *hostname*** command. The SRX administrator must provide the hostname to remote users.

d. For Server URL Name, enter the IP address of the SRX gateway.

NOTE: On the SRX Series device, this IP address is the IP address of the external-interface configured with the **set security ike gateway *gateway-name*** command. The SRX administrator must provide the IP address to remote users.

3. Click **Add**, then click **Connect**. The Pulse Secure remote client program connects to the SRX Series using HTTPS.

4. Enter your username and password when prompted. Configuration information is downloaded from the SRX Series device to the remote client to enable the client to establish an IKE SA with the SRX Series device.

5. If you are accessing dynamic VPN for the first time, enter your user credentials again to establish an IPsec SA. An IP address is assigned to the remote client from a local address pool or from an external RADIUS server.

   NOTE: The user credentials you enter in step 4 are used to download the configuration to the remote client and establish an IKE SA between the client and the SRX Series device. The user credentials entered in this step are used to establish an IPsec SA. The user credentials can be the same or different, based on the configuration on the SRX Series device.

6. Upon successful authentication and address assignment, a tunnel is established.

Related Documentation

- Dynamic VPN Overview on page 849
- Dynamic VPN Configuration Overview on page 855
- Understanding Dynamic VPN Tunnel Support on page 851
- Example: Configuring Dynamic VPN on page 857

Dynamic VPN Proposal Sets

Supported Platforms SRX Series

This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices. Configuring custom Internet Key Exchange (IKE) and IP Security (IPsec) proposals for IKE and IPsec policies can be tedious and time-consuming when there are many dynamic VPN clients. The administrator can select basic, compatible, or standard proposal sets for dynamic VPN clients. Each proposal set consists of two or more predefined proposals. The server selects one predefined proposal from the set and pushes it to the client in the client configuration. The client uses this proposal in negotiations with the server to establish the connection.

The default values for IKE and IPsec security association (SA) rekey timeout are as follows:

- For IKE SAs, the rekey timeout is 28,800 seconds.
- For IPsec SAs, the rekey timeout is 3600 seconds.

NOTE: **Because proposal set configuration does not allow for configuration of rekey timeout, these values are included in the client configuration that is sent to the client at client download time.**

The basic use cases for proposals are as follows:

- IKE and IPsec both use proposal sets.

  The server selects a predefined proposal from the proposal set and sends it to the client, along with the default rekey timeout value.

- IKE uses a proposal set, and IPsec uses a custom proposal.

  The server sends a predefined IKE proposal from the configured IKE proposal set to the client, along with the default rekey timeout value. For IPsec, the server sends the setting that is configured in the IPsec proposal.

- IKE uses a custom proposal, and IPsec uses a proposal set.

  The server sends a predefined IPsec proposal from the configured IPsec proposal set to the client, along with the default rekey timeout value. For IKE, the server sends the setting that is configured in the IKE proposal.

NOTE: **If IPsec uses a standard proposal set and perfect forward secrecy (PFS) is not configured, then the default Perfect Forward Secrecy (PFS) is group2. For other proposal sets, PFS will not be set, because it is not configured. Also, for the IPsec proposal set, the group configuration in ipsec policy perfect-forward-secrecy keys overrides the Diffie-Hellman (DH) group setting in the proposal sets.**

Because the client accepts only one proposal for negotiating tunnel establishment with the server, the server internally selects one proposal from the proposal set to send to the client. The selected proposal for each set is listed as follows:

For IKE

- Sec-level basic: preshared key, g1, des, sha1
- Sec-level compatible: preshared key, g2, 3des, sha1
- Sec-level standard: preshared key, g2, aes128, sha1

For IPsec

- Sec-level basic: esp, no pfs (if not configured) or group*x* (if configured), des, sha1
- Sec-level compatible: esp, no pfs (if not configured) or group*x* (if configured), 3des, sha1
- Sec-level standard: esp, g2 (if not configured) or group*x* (if configured), aes128, sha1

Related Documentation

- Dynamic VPN Overview on page 849

Dynamic VPN Configuration Overview

Supported Platforms SRX Series

Dynamic VPN allows you to provide IPsec access for remote users to a gateway on a Juniper Networks device. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

There are two cases to consider when configuring dynamic VPN:

- When users are configured locally, they are configured at the [**edit access profile *profile-name* client *client-name***] hierarchy level and arranged into user groups using the **client-group** configuration option.
- Users can be configured on an external authentication server, such as a RADIUS server. Users configured on an external authentication server do not need to be configured at the [**edit access profile *profile-name***] hierarchy level.

For locally-configured users, the user group needs to be specified in the dynamic VPN configuration so that a user can be associated with a client configuration. You specify a user group with the **user-groups** option at the [**edit security dynamic-vpn clients *configuration-name***] hierarchy level.

When a user is authenticated, the user group is included in the authentication reply. This information is extracted and user groups configured at the [**edit security dynamic-vpn clients *configuration-name***] hierarchy level are searched to determine which client configuration to retrieve and return to the client for tunnel establishment.

If a user is associated with more than one user group, the first matching user group configuration is used. If a user creates a second connection, then the next matching user group configuration is used. Subsequent user connections use the next matching user group configuration until there are no more matching configurations.

The following procedure lists the tasks for configuring dynamic VPN.

1. Configure authentication and address assignment for the remote clients:

   a. Configure an XAuth profile to authenticate users and assign addresses. Either local authentication or an external RADIUS server can be used. Use the **profile** configuration statement at the [**edit access**] hierarchy level to configure the XAuth profile.

   b. Assign IP addresses from a local address pool if local authentication is used. Use the **address-assignment pool** configuration statement at the [**edit access**] hierarchy level. A subnet or a range of IP addresses can be specified. IP addresses for DNS and WINS servers can also be specified.

2. Configure the VPN tunnel:

   a. Configure the IKE policy. The mode must be aggressive. Basic, compatible, or standard proposal sets can be used. Only preshared keys are supported for Phase 1 authentication. Use the **policy** configuration statement at the [**edit security ike**] hierarchy level.

   b. Configure the IKE gateway. Either shared or group IKE IDs can be used. You can configure the maximum number of simultaneous connections to the gateway. Use the **gateway** configuration statement at the [**edit security ike**] hierarchy level.

   c. Configure the IPsec VPN. Basic, compatible, or standard proposal sets can be specified with the **policy** configuration statement at the [**edit security ipsec**] hierarchy level. Use the **vpn** configuration statement at the [**edit security ipsec**] hierarchy level to configure the IPsec gateway and policy.

   NOTE: A configuration check can be performed to verify that all IKE and IPsec parameters needed for dynamic VPN are correctly configured. If the configuration is invalid for IKE or IPsec, an error message is displayed. You enable the configuration check with the **set security dynamic-vpn config-check** command.

   d. Configure a security policy to allow traffic from the remote clients to the IKE gateway. Use the **policy** configuration statement at the [**edit security policies from-zone *zone* to-zone *zone***] hierarchy level.

   NOTE: Configure the security policy with the match criteria **source-address any**, **destination-address any**, and **application any** and the action **permit tunnel ipsec-vpn** with the name of the dynamic VPN tunnel. Place this policy at the end of the policy list.

   e. Configure host inbound traffic to allow specific traffic to reach the device from systems that are connected to its interfaces. For example, IKE and HTTPS traffic must be allowed. See *Understanding How to Control Inbound Traffic Based on Traffic Types*.

f. (Optional) If the client address pool belongs to a subnet that is directly connected to the device, the device would need to respond to ARP requests to addresses in the pool from other devices in the same zone. Use the **proxy-arp** configuration statement at the [**edit security nat**] hierarchy level. Specify the interface that directly connects the subnet to the device and the addresses in the pool.

3. Associate the dynamic VPN with remote clients:

   a. Specify the access profile for use with dynamic VPN. Use the **access-profile** configuration statement at the [**edit security dynamic-vpn**] hierarchy level.

   b. Configure the clients who can use the dynamic VPN. Specify protected resources (traffic to the protected resource travels through the specified dynamic VPN tunnel and is therefore protected by the firewall's security policies) or exceptions to the protected resources list (traffic that does not travel through the dynamic VPN tunnel and is sent in cleartext). These options control the routes that are pushed to the client when the tunnel is up, therefore controlling the traffic that is send through the tunnel. Use the **clients** configuration statement at the [**edit security dynamic-vpn**] hierarchy level.

4. To log dynamic VPN messages, configure the **traceoptions** statement at the [**edit security dynamic-vpn**] hierarchy level.

Related Documentation

- Dynamic VPN Overview on page 849
- Understanding Dynamic VPN Tunnel Support on page 851
- *Understanding Security Basics*
- Example: Configuring Dynamic VPN on page 857

Example: Configuring Dynamic VPN

Supported Platforms SRX Series

This example shows how to configure a dynamic VPN on a Juniper Networks device to provide VPN access to remote clients. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

- Requirements on page 858
- Overview on page 858
- Configuration on page 861
- Verification on page 866

Requirements

Before you begin:

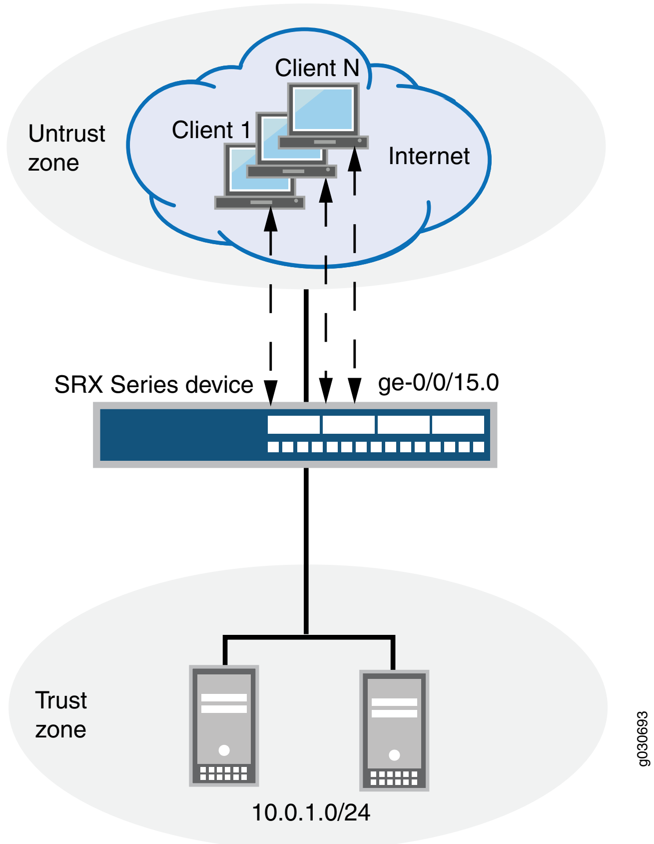
1. Configure network interfaces on the device. See *Interfaces Feature Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 111.
3. If there will be more than two simultaneous user connections, install a Dynamic VPN license in the device. See *Installation and Upgrade Guide*.
4. Read “Dynamic VPN Configuration Overview” on page 855.

Overview

A common deployment scenario for dynamic VPN is to provide VPN access to remote clients that are connected through a public network such as the Internet. A public IP address is assigned to one of the gateway’s interfaces; this interface is normally part of the untrust zone. After the client software is installed, the remote user can access the VPN by either logging in to the Web portal or by launching the client directly. In either case, the remote client authenticates with the SRX Series device and downloads the latest configuration available.

Figure 68 on page 859 illustrates this deployment topology. The ge-0/0/15.0 interface on the SRX Series device is the termination point for the dynamic VPN tunnel. Remote clients in the untrust zone access the ge-0/0/15.0 interface through a Pulse Secure client.

Figure 68: Dynamic VPN Deployment Topology



In this example, XAuth client authentication is performed locally and client IP addresses are assigned from an address pool configured on the SRX Series device. See Table 81 on page 859.

Then, standard proposal sets are used for both IKE and IPsec negotiations. For dynamic VPN tunnels, aggressive mode must be configured and only preshared keys are supported for Phase 1 authentication. A group IKE ID is used and the maximum number of connections is set to 10. Because dynamic VPNs must be policy-based VPNs, a security policy must be configured to forward traffic to the tunnel. IKE and HTTPS traffic must be allowed for host inbound traffic.See Table 82 on page 860.

Finally, the XAuth profile configured for remote clients is specified for the dynamic VPN. Remote users are associated with the configured IPsec VPN. Also configured are remote protected resources (the destination addresses of traffic that is always sent through the tunnel) and remote exceptions (the destination addresses of traffic that is sent in cleartext instead of through the tunnel). See Table 83 on page 860.

Table 81: Remote Client Authentication and Address Assignment Configuration

Feature	Name	Configuration Parameters
IP address pool	dyn-vpn-address-pool	• Addresses: 10.10.10.0/24 • DNS server address: 192.0.2.1/32.

Table 81: Remote Client Authentication and Address Assignment Configuration *(continued)*

Feature	Name	Configuration Parameters
XAuth profile	dyn-vpn-access-profile	• Remote client username: 'client1' with password $ABC123 • Remote client username: 'client2' with password $ABC456 • IP address pool reference: dyn-vpn-address-pool • This profile is the default profile for web authentication.

Table 82: VPN Tunnel Configuration Parameters

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	ike-dyn-vpn-policy	• Mode: aggressive • Proposal set: standard • Preshared key: (ASCII) $ABC789
IKE gateway (Phase 1)	dyn-vpn-local-gw	• IKE policy reference: ike-dyn-vpn-policy • Dynamic hostname: dynvpn • IKE user type: group IKE ID • Maximum number of concurrent connections: 10 • External interface: ge-0/0/15.0 • Access profile reference: dyn-vpn-access-profile
IPsec policy (Phase 2)	ipsec-dyn-vpn-policy	Proposal set: standard
IPsec VPN (Phase 2)	dyn-vpn	• IKE gateway reference: dyn-vpn-local-gw • IPsec policy reference: ipsec-dyn-vpn-policy

Security policy (permits traffic from the untrust zone to the trust zone)	dyn-vpn-policy	• Match criteria:
		  ◦ source address any
		  ◦ destination address any
		  ◦ application any
		• Permit action: tunnel ipsec-vpn dyn-vpn

Host inbound traffic		Allow the following types of traffic to the ge-0/0/15.0 interface in the untrust zone: • IKE • HTTPS • ping

Table 83: Dynamic VPN Configuration for Remote Clients

Feature	Name	Configuration Parameters
Access profile for remote clients		Access profile reference: dyn-vpn-access-profile

Table 83: Dynamic VPN Configuration for Remote Clients *(continued)*

Feature	Name	Configuration Parameters
Remote clients	all	• IPsec VPN reference: dyn-vpn • User name reference: client1 and client2 • Remote protected resources: 10.0.0.0/8 • Remote exceptions: 0.0.0.0/0

Configuration

- Configuring the Remote User Authentication and Address Assignment on page 861
- Configuring the VPN Tunnel on page 862
- Associate the Dynamic VPN with Remote Clients on page 865

Configuring the Remote User Authentication and Address Assignment

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 access profile dyn-vpn-access-profile client client1 firewall-user password "$ABC123"
set access profile dyn-vpn-access-profile client client2 firewall-user password "$ABC456"
set access profile dyn-vpn-access-profile address-assignment pool dyn-vpn-address-pool
set access address-assignment pool dyn-vpn-address-pool family inet network
  10.10.10.0/24
set access address-assignment pool dyn-vpn-address-pool family inet xauth-attributes
  primary-dns 192.0.2.1/32
set access firewall-authentication web-authentication default-profile
  dyn-vpn-access-profile
```

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 remote user authentication and address assignment:

1. Create the address assignment pool.

```
[edit access address-assignment]
user@host# set pool dyn-vpn-address-pool family inet network 10.10.10.0/24
user@host# set pool dyn-vpn-address-pool family inet xauth-attributes primary-dns
  192.0.2.1/32
```

2. Configure the XAuth profile.

```
[edit access]
user@host# set profile dyn-vpn-access-profile client client1 firewall-user password
  "$ABC123"
user@host# set profile dyn-vpn-access-profile client client2 firewall-user password
  "$ABC456"
```



```
user@host# set profile dyn-vpn-access-profile address-assignment pool
  dyn-vpn-address-pool
```

3. Configure Web authentication using the XAuth profile.

```
[edit access firewall-authentication]
user@host# set web-authentication default-profile dyn-vpn-access-profile
```

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

```
[edit]
user@host# show access
profile dyn-vpn-access-profile {
  client client1 {
    firewall-user {
      password "$ABC123"; ## SECRET-DATA
    }
  }
  client client2 {
    firewall-user {
      password "$ABC456"; ## SECRET-DATA
    }
  }
  address-assignment {
    pool dyn-vpn-address-pool;
  }
}
address-assignment {
  pool dyn-vpn-address-pool {
    family inet {
      network 10.10.10.0/24;
      xauth-attributes {
        primary-dns 192.02.1/32;
      }
    }
  }
}
firewall-authentication {
  web-authentication {
    default-profile dyn-vpn-access-profile;
  }
}
```

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

Configuring the VPN Tunnel

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 security ike policy ike-dyn-vpn-policy mode aggressive
set security ike policy ike-dyn-vpn-policy proposal-set standard
set security ike policy ike-dyn-vpn-policy pre-shared-key ascii-text "$ABC789"
set security ike gateway dyn-vpn-local-gw ike-policy ike-dyn-vpn-policy
set security ike gateway dyn-vpn-local-gw dynamic hostname dynvpn
set security ike gateway dyn-vpn-local-gw dynamic connections-limit 10
set security ike gateway dyn-vpn-local-gw dynamic ike-user-type group-ike-id
set security ike gateway dyn-vpn-local-gw external-interface ge-0/0/15.0
set security ike gateway dyn-vpn-local-gw aaa access-profile dyn-vpn-access-profile
set security ipsec policy ipsec-dyn-vpn-policy proposal-set standard
set security ipsec vpn dyn-vpn ike gateway dyn-vpn-local-gw
set security ipsec vpn dyn-vpn ike ipsec-policy ipsec-dyn-vpn-policy
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy match
  source-address any
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy match
  destination-address any
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy match
  application any
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy then permit
  tunnel ipsec-vpn dyn-vpn
set security zones security-zone untrust interfaces ge-0/0/15.0 host-inbound-traffic
  system-services ike
set security zones security-zone untrust interfaces ge-0/0/15.0 host-inbound-traffic
  system-services https
set security zones security-zone untrust interfaces ge-0/0/15.0 host-inbound-traffic
  system-services ping
```

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 VPN tunnel:

1. Configure the IKE policy.

   ```
   [edit security ike]
   user@host# set policy ike-dyn-vpn-policy mode aggressive
   user@host# set policy ike-dyn-vpn-policy proposal-set standard
   user@host# set policy ike-dyn-vpn-policy pre-shared-key ascii-text "$ABC789"
   ```

2. Configure the IKE gateway.

   ```
   [edit security ike]
   user@host# set gateway dyn-vpn-local-gw ike-policy ike-dyn-vpn-policy
   user@host# set gateway dyn-vpn-local-gw dynamic hostname dynvpn
   user@host# set gateway dyn-vpn-local-gw dynamic ike-user-type group-ike-id
   user@host# set gateway dyn-vpn-local-gw dynamic connections-limit 10
   user@host# set gateway dyn-vpn-local-gw external-interface ge-0/0/15.0
   user@host# set gateway dyn-vpn-local-gw aaa access-profile
     dyn-vpn-access-profile
   ```

3. Configure IPsec.

```
[edit security ipsec]
user@host# set policy ipsec-dyn-vpn-policy proposal-set standard
user@host# set vpn dyn-vpn ike gateway dyn-vpn-local-gw
user@host# set vpn dyn-vpn ike ipsec-policy ipsec-dyn-vpn-policy
```

4. Configure the security policy.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy dyn-vpn-policy match source-address any destination-address
  any application any
user@host# set policy dyn-vpn-policy then permit tunnel ipsec-vpn dyn-vpn
```

5. Configure host inbound traffic.

```
[edit security zones security-zone untrust interfaces ge-0/0/15.0]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services https
user@host# set host-inbound-traffic system-services ping
```

Results From configuration mode, confirm your configuration by entering the **show security ike**, **show security ipsec**, **show security policies**, 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 security ike
policy ike-dyn-vpn-policy {
  mode aggressive;
  proposal-set standard;
  pre-shared-key ascii-text "$ABC789"; ## SECRET-DATA
}
gateway dyn-vpn-local-gw {
  ike-policy ike-dyn-vpn-policy;
  dynamic {
    hostname dynvpn;
    connections-limit 10;
    ike-user-type group-ike-id;
  }
  external-interface ge-0/0/15.0;
  aaa access-profile dyn-vpn-access-profile;
}
```

```
[edit]
user@host# show security ipsec
policy ipsec-dyn-vpn-policy {
  proposal-set standard;
}
vpn dyn-vpn {
  ike {
    gateway dyn-vpn-local-gw;
    ipsec-policy ipsec-dyn-vpn-policy;
  }
```



```
}
[edit]
user@host# show security policies
from-zone untrust to-zone trust {
   policy dyn-vpn-policy {
      match {
         source-address any;
         destination-address any;
         application any;
      }
      then {
         permit {
            tunnel {
               ipsec-vpn dyn-vpn;
            }
         }
      }
   }
[edit]
user@host# show security zones
security-zone untrust {
   interfaces {
      ge-0/0/15.0 {
         host-inbound-traffic {
            system-services {
               ike;
               https;
               ping;
            }
         }
      }
   }
}
```

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

Associate the Dynamic VPN with Remote Clients

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 dynamic-vpn access-profile dyn-vpn-access-profile
set security dynamic-vpn clients all remote-protected-resources 10.0.0.0/8
set security dynamic-vpn clients all remote-exceptions 0.0.0.0/0
set security dynamic-vpn clients all ipsec-vpn dyn-vpn
set security dynamic-vpn clients all user client1
set security dynamic-vpn clients all user client2
```

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 associate the dynamic VPN with remote clients:

1. Specify the access profile to use with dynamic VPN.

   ```
   [edit security dynamic-vpn]
   user@host# set access-profile dyn-vpn-access-profile
   ```

2. Configure the clients who can use the dynamic VPN.

   ```
   [edit security dynamic-vpn]
   user@host# set clients all ipsec-vpn dyn-vpn
   user@host# set clients all user client1
   user@host# set clients all user client2
   user@host# set clients all remote-protected-resources 10.0.0.0/8
   user@host# set clients all remote-exceptions 0.0.0.0/0
   ```

Results

From configuration mode, confirm your configuration by entering the **show security dynamic-vpn** 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 dynamic-vpn
access-profile dyn-vpn-access-profile;
clients {
  all {
    remote-protected-resources {
      10.0.0.0/8;
    }
    remote-exceptions {
      0.0.0.0/0;
    }
    ipsec-vpn dyn-vpn;
    user {
      client1;
      client2;
    }
  }
}
```

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

Verification

Dynamic VPN tunnels can be monitored with the same commands used to monitor traditional IPsec VPN tunnels. To confirm that the configuration is working properly, perform these tasks:

- Verifying IKE Phase 1 Status on page 867
- Verifying Connected Clients and Assigned Addresses on page 867

Verifying IKE Phase 1 Status

Purpose Verify the IKE Phase 1 status of the security associations.

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
18      172.19.100.99   UP     37b45aa1469e488b  7d4454404002e2e6  Aggressive
```

Verifying Connected Clients and Assigned Addresses

Purpose Verify that the remote clients and the IP addresses assigned to them are using XAuth.

Action From operational mode, enter the **show security ike active-peer** command.

```
user@host> show security ike active-peer
Remote Address          Port     Peer IKE-ID      XAUTH username      Assigned
 IP
172.19.100.99           500     testdynvpn       test                10.10.10.2
```

Verifying IPsec Phase 2 Status

Purpose Verify the IPsec Phase 2 status of the security associations.

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

```
user@host> show security ipsec security-associations
  Total active tunnels: 1
  ID          Gateway       Port  Algorithm       SPI      Life:sec/kb  Mon vsys
  <133955586 172.19.100.99 500 ESP:aes-128/sha1 9c23b7a9 2862/  449996 -  root
  >133955586 172.19.100.99 500 ESP:aes-128/sha1 c72c8f88 2862/  449996 -  root
```

Verifying Concurrent Connections and Parameters for Each User

Purpose Verify the number of concurrent connections and the negotiated parameters for each user.

Action From operational mode, enter the **show security dynamic-vpn users** command.

```
user@host> show security dynamic-vpn users
```

```
User: test , User group: group-one, Number of connections: 1
    Remote IP: 172.19.100.99
    IPSEC VPN: dyn-vpn
    IKE gateway: dyn-vpn-local-gw
    IKE ID   : testdynvpn
    IKE Lifetime: 28800
    IPSEC Lifetime: 3600
    Status: CONNECTED
```

Related Documentation

- Dynamic VPN Overview on page 849
- Understanding Dynamic VPN Tunnel Support on page 851
- Dynamic VPN Configuration Overview on page 855

Understanding Local Authentication and Address Assignment

Supported Platforms SRX Series

This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices. A client application can request an IP address on behalf of a client. This request is made at the same time as the client authentication request. Upon successful authentication of the client, an IP address can be assigned to the client from a predefined address pool or a specific IP address can be assigned. Other attributes, such as WINS or DNS server IP addresses, can also be provided to the client.

Address pools are defined with the **pool** configuration statement at the [**edit access address-assignment**] hierarchy level. An address pool definition contains network information (IP address with optional netmask), optional range definitions, and DHCP or XAuth attributes that can be returned to the client. If all addresses in a pool are assigned, a new request for a client address will fail even if the client is successfully authenticated.

Access profiles are defined with the **profile** configuration statement at the [**edit access**] hierarchy. A defined address pool can be referenced in an access profile configuration.

You can also bind a specific IP address to a client in an access profile with the **xauth ip-address** ***address*** option. The IP address must be in the range of addresses specified in the address pool. It must also be different from the IP address specified with the **host** configuration statement at the [**edit access profile address-assignment pool** ***pool-name*** **family inet**] hierarchy level. For any application, if one IP address has been assigned, it will not be reassigned again until it is released.

Related Documentation

- Example: Configuring Local Authentication and Address Pool on page 869
- Dynamic VPN Overview on page 849
- Understanding Dynamic VPN Tunnel Support on page 851
- Dynamic VPN Configuration Overview on page 855
- Example: Configuring Dynamic VPN on page 857

Example: Configuring Local Authentication and Address Pool

Supported Platforms SRX Series

This example shows how to create an address pool and how to assign client IP addresses in an access profile. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Requirements

Before you begin, configure primary and secondary DNS and WINS servers and assign IP addresses to them.

Overview

This example creates an address pool **xauth1** that consists of the IP addresses in the 192.0.2.0/24 subnet. The **xauth1** pool also assigns IP addresses for primary and secondary DNS and WINS servers.

The access profile **dvpn-auth** references the xauth1 pool. The **dvpn-auth** access profile configures two clients:

- jason: The IP address 192.0.2.1 is bound to this client. Upon successful authentication, the client is assigned the IP address 192.0.2.1. If the client logs in again before logging out, the client is assigned an IP address from the xauth1 pool.
- jacky: Upon successful authentication, the client is assigned an IP address from the xauth1 pool.

In addition, the **dvpn-auth** access profile specifies that password authentication is used to verify clients at login. Additional authentication methods can be specified; the software tries the authentication methods in order, from first to last, for each client login attempt.

Configuration

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 access profile dvpn-auth authentication-order password
set access profile dvpn-auth client jacky firewall-user password "$ABC123"
set access profile dvpn-auth client jason xauth ip-address 192.0.2.1/32
set access profile dvpn-auth client jason firewall-user password "$ABC456"
set access profile dvpn-auth address-assignment pool xauth1
set access address-assignment pool xauth1 family inet network 192.0.2.0/24
set access address-assignment pool xauth1 family inet xauth-attributes primary-dns
  192.0.2.250/32
set access address-assignment pool xauth1 family inet xauth-attributes secondary-dns
  192.0.2.251/32
set access address-assignment pool xauth1 family inet xauth-attributes primary-wins
  192.0.2.253/32
```

set access address-assignment pool xauth1 family inet xauth-attributes secondary-wins 192.0.2.254/32

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 an address pool and an access profile that uses the address pool:

1. Create the address pool.

   ```
   [edit access address-assignment]
   user@host# set pool xauth1 family inet network 192.0.2.0/24 xauth-attributes
     primary-dns 192.0.2.250 secondary-dns 192.0.2.251 primary-wins 192.0.2.253
     secondary-wins 192.0.2.254
   ```

2. Configure the access profile.

   ```
   [edit access]
   user@host# set profile dvpn-auth address-assignment pool xauth1
   user@host# set profile dvpn-auth authentication-order password
   user@host# set profile dvpn-auth client jason xauth ip-address 192.0.2.1
   user@host# set profile dvpn-auth client jason firewall-user password jason
   user@host# set profile dvpn-auth client jacky firewall-user password jacky
   ```

Results

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

```
[edit]
user@host# show access
  profile dvpn-auth {
    authentication-order password;
    client jacky {
      firewall-user {
        password "$ABC123"; ## SECRET-DATA
      }
    }
    client jason {
      xauth {
        ip-address 192.0.2.1/32;
      }
      firewall-user {
        password "$ABC456"; ## SECRET-DATA
      }
    }
    address-assignment {
      pool xauth1;
    }
  }
    address-assignment {
      pool xauth1 {
        family inet {
```



```
            network 192.0.2.0/24;
            xauth-attributes {
              primary-dns 192.0.2.250/32;
              secondary-dns 192.0.2.251/32;
              primary-wins 192.0.2.253/32;
              secondary-wins 192.0.2.254/32;
            }
          }
        }
      }
```

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

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Address Assignment on page 871

Verifying Address Assignment

Purpose

Verify address assignment. For XAuth, the hardware address is always shown as NA. If a static IP address is assigned to a specific user, the user name and profile name (in the format user@profile) is displayed in the "Host/User" column. If a client is assigned an IP address from the pool, the username is displayed; if the username does not exist, NA is displayed. For other applications (for example, DHCP), the hostname is displayed if configured; if the hostname is not configured, NA is displayed.

Action

From operational mode, enter the **show network-access address-assignment pool** command.

user

```
user@host> show network-access address-assignment pool xauth1
IP address       Hardware address        Host/User                 Type
192.0.2.1        NA                      jason@dvpn-auth           XAUTH
192.0.2.2        NA                      jacky                     XAUTH
```

Related Documentation

- Understanding Local Authentication and Address Assignment on page 868
- Dynamic VPN Overview on page 849
- Understanding Dynamic VPN Tunnel Support on page 851
- Dynamic VPN Configuration Overview on page 855
- Example: Configuring Dynamic VPN on page 857

Understanding Group and Shared IKE IDs

Supported Platforms SRX Series

This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices. With dynamic VPN, a unique Internet Key Exchange (IKE) ID is used for each user connection. When there are a large number of users who need to access the VPN, configuring an individual IKE gateway, IPsec VPN, and a security policy for each user can be cumbersome. The group IKE ID and shared IKE ID features allow a number of users to share an IKE gateway configuration, thus reducing the number of VPN configurations required.

NOTE: **We recommend that you configure group IKE IDs for dynamic VPN deployments because group IKE IDs provide a unique preshared key and IKE ID for each user.**

This topic includes the following sections:

Group IKE IDs

When group IKE IDs are configured, the IKE ID of each user is a concatenation of a user-specific part and a part that is common to all group IKE ID users. For example, the user Bob might use "Bob.example.net" as his full IKE ID, where ".example.net" is common to all users. The full IKE ID is used to uniquely identify each user connection.

Although group IKE IDs do not require XAuth, XAuth is required by dynamic VPN to retrieve network attributes like client IP addresses. A warning is displayed if XAuth is not configured for a dynamic VPN that uses group IKE IDs.

NOTE: **We recommend that users use the same credentials for both WebAuth and XAuth authentication when group IKE IDs are configured.**

Multiple users can use the same group IKE ID, but a single user cannot use the same group IKE ID for different connections. If a user needs to have connections from different remote clients, they need to have different group IKE IDs configured, one for each connection. If a user only has one group IKE ID configured and attempts a second connection from another PC, the first connection will be terminated to allow the second connection to go through.

To configure a group IKE ID:

- Configure **ike-user-type group-ike-id** at the [**edit security ike gateway *gateway-name* dynamic**] hierarchy level.
- Configure the **hostname** configuration statement at the [**edit security ike gateway *gateway-name* dynamic**] hierarchy level. This configuration is the common part of the full IKE ID for all users.

- Configure the **pre-shared-key** configuration statement at the [**edit security ike policy *policy-name***] hierarchy level. The configured preshared key is used to generate the actual preshared key.

Shared IKE IDs

When a shared IKE ID is configured, all users share a single IKE ID and a single IKE preshared key. Each user is authenticated through the mandatory XAuth phase, where the credentials of individual users are verified either with an external RADIUS server or with a local access database. XAuth is required for shared IKE IDs.

The XAuth user name together with the configured shared IKE ID is used to distinguish between different user connections. Because the user name is used to identify each user connection, both the WebAuth user name and XAuth user name must be the same.

Multiple users can use the same shared IKE ID, but a single user cannot use the same shared IKE ID for different connections. If a user needs to have connections from different remote clients, they need to have different shared IKE IDs configured, one for each connection. If a user has only one shared IKE ID configured and attempts a second connection from another client, the first connection will be terminated to allow the second connection to go through. Also, because the user name is needed to identify each user connection along with the IKE ID, the user must use the same credentials for both WebAuth and XAuth authentication.

To configure a shared IKE ID:

- Configure **ike-user-type shared-ike-id** at the [**edit security ike gateway *gateway-name* dynamic**] hierarchy level.
- Configure the **hostname** configuration statement at the [**edit security ike gateway *gateway-name* dynamic**] hierarchy level. The configured hostname is shared by all users configured in the dynamic VPN access profile.
- Configure the **pre-shared-key** configuration statement at the [**edit security ike policy *policy-name***] hierarchy level. The configured preshared key is shared by all users configured in the dynamic VPN access profile.

Related Documentation

- Understanding Dynamic VPN Tunnel Support on page 851
- Dynamic VPN Configuration Overview on page 855
- Example: Configuring a Group IKE ID for Multiple Users on page 873
- Example: Configuring Individual IKE IDs for Multiple Users on page 881

Example: Configuring a Group IKE ID for Multiple Users

Supported Platforms SRX Series

This example shows how to configure a group IKE ID that is used by multiple users. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Requirements

Before you begin:

- Configure network interfaces on the device. See the *Interfaces Feature Guide for Security Devices*.
- Create security zones and assign interfaces to them. See *Understanding Security Zones*.
- If there will be more than two simultaneous user connections, install a Dynamic VPN license in the device. See *Installation and Upgrade Guide*.
- Read "Dynamic VPN Configuration Overview" on page 855.

Overview

In this example, you configure two remote dynamic VPN users who use a single IKE ID and a single IKE preshared key (see Table 84 on page 874 and Table 85 on page 875). An external RADIUS server is used to authenticate users and assign IP addresses to clients (see Table 86 on page 875).

Table 84: Group IKE ID VPN Tunnel Configuration Parameters

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	clientpol-group	• Mode: aggressive • Proposal set: compatible • Preshared key: (ASCII) for-everyone-in-access-profile
IKE gateway (Phase 1)	groupgw	• IKE policy reference: clientpol-group • Dynamic hostname: example.net • IKE user type: group IKE ID • Maximum number of concurrent connections: 50 • External interface: ge-0/0/0.0 • Access profile reference: radius-profile
IPsec policy (Phase 2)	client1vpnPol	Proposal set: compatible
IPsec VPN (Phase 2)	groupvpn	• IKE gateway reference: groupgw • IPsec policy reference: client1vpnPol

Table 84: Group IKE ID VPN Tunnel Configuration Parameters *(continued)*

Feature	Name	Configuration Parameters
Security policy (permits traffic from the untrust zone to the trust zone)	group-sec-policy	• Match criteria: ◦ source address any ◦ destination address any ◦ application any • Permit action: tunnel ipsec-vpn groupvpn
Host inbound traffic		Allow the following types of traffic to the ge-0/0/0.0 interface in the untrust zone: • IKE • HTTPS • ping • SSH

Table 85: Group IKE ID Dynamic VPN Configuration for Remote Clients

Feature	Name	Configuration Parameters
Access profile for remote clients		Access profile reference: radius-profile
Remote clients	groupcfg	• IPsec VPN reference: groupvpn • User name reference: derek and chris • Remote protected resources: 10.100.100.0/24 • Remote exceptions: 0.0.0.0/0, 192.0.2.1/24, 0.0.0.0/32

Table 86: RADIUS Server User Authentication (Group IKE ID)

Feature	Name	Configuration Parameters
XAuth profile	radius-profile	• RADIUS is the authentication method used to verify user credentials. • The RADIUS server IP address is 10.100.100.250 and the password is secret. • This profile is the default profile for Web authentication.

Configuration

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 access profile radius-profile authentication-order radius
set access profile radius-profile radius-server 10.100.100.250 secret "$ABC123"
set access firewall-authentication web-authentication default-profile radius-profile
set security ike policy clientpol-group mode aggressive
set security ike policy clientpol-group proposal-set compatible
set security ike policy clientpol-group pre-shared-key ascii-text "$ABC456"
set security ike gateway groupgw ike-policy clientpol-group
```

```
set security ike gateway groupgw dynamic hostname example.net
set security ike gateway groupgw dynamic connections-limit 50
set security ike gateway groupgw dynamic ike-user-type group-ike-id
set security ike gateway groupgw external-interface ge-0/0/0.0
set security ike gateway groupgw aaa access-profile radius-profile
set security ipsec policy client1vpnPol proposal-set compatible
set security ipsec vpn groupvpn ike gateway groupgw
set security ipsec vpn groupvpn ike ipsec-policy client1vpnPol
set security policies from-zone untrust to-zone trust policy group-sec-policy match
  source-address any
set security policies from-zone untrust to-zone trust policy group-sec-policy match
  destination-address any
set security policies from-zone untrust to-zone trust policy group-sec-policy match
  application any
set security policies from-zone untrust to-zone trust policy group-sec-policy then permit
  tunnel ipsec-vpn groupvpn
set security dynamic-vpn access-profile radius-profile
set security dynamic-vpn clients groupcfg remote-protected-resources 10.100.100.0/24
set security dynamic-vpn clients groupcfg remote-exceptions 0.0.0.0/0
set security dynamic-vpn clients groupcfg remote-exceptions 192.0.2.1/24
set security dynamic-vpn clients groupcfg remote-exceptions 0.0.0.0/32
set security dynamic-vpn clients groupcfg ipsec-vpn groupvpn
set security dynamic-vpn clients groupcfg user chris
set security dynamic-vpn clients groupcfg user derek
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services https
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ssh
```

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 group IKE ID for multiple users:

1. Configure the XAuth profile.

```
[edit access]
user@host# set profile radius-profile authentication-order radius
user@host# set profile radius-profile radius-server 10.100.100.250 secret secret
user@host# set firewall-authentication web-authentication default-profile
  radius-profile
```

2. Configure the IKE policy.

```
[edit security ike]
user@host# set policy clientpol-group mode aggressive
user@host# set policy clientpol-group proposal-set compatible
user@host# set policy clientpol-group pre-shared-key ascii-text
  for-everyone-in-access-profile
```


3. Configure the IKE gateway.

```
[edit security ike]
user@host# set gateway groupgw ike-policy clientpol-group
user@host# set gateway groupgw dynamic hostname example.net
user@host# set gateway groupgw dynamic ike-user-type group-ike-id
user@host# set gateway groupgw dynamic connections-limit 50
user@host# set gateway groupgw external-interface ge-0/0/0.0
user@host# set gateway groupgw aaa access-profile radius-profile
```

4. Configure IPsec.

```
[edit security ipsec]
user@host# set policy client1vpnPol proposal-set compatible
user@host# set vpn groupvpn ike gateway groupgw
user@host# set vpn groupvpn ike ipsec-policy client1vpnPol
```

5. Configure the security policy.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy group-sec-policy match source-address any
  destination-address any application any
user@host# set policy group-sec-policy then permit tunnel ipsec-vpn groupvpn
```

6. Configure host inbound traffic.

```
[edit security zones security-zone untrust interfaces ge-0/0/0.0]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services https
user@host# set host-inbound-traffic system-services ping
user@host# set host-inbound-traffic system-services ssh
```

7. Specify the access profile to use with dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set access-profile radius-profile
```

8. Configure the clients who can use the dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set clients groupcfg ipsec-vpn groupvpn
user@host# set clients groupcfg user derek
user@host# set clients groupcfg user chris
user@host# set clients groupcfg remote-protected-resources 10.100.100.0/24
user@host# set clients groupcfg remote-exceptions 0.0.0.0/0
user@host# set clients groupcfg remote-exceptions 192.0.2.1/24
user@host# set clients groupcfg remote-exceptions 0.0.0.0/32
```

Results From configuration mode, confirm your configuration by entering the **show security ike**, **show security ipsec**, **show security policies**, **show security zones**, and **show security dynamic-vpn** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show access
profile radius-profile {
  authentication-order radius;
  radius-server {
    10.100.100.250 secret "$ABC123"; ## SECRET-DATA
  }
}
  firewall-authentication {
    web-authentication {
      default-profile radius-profile;
    }
  }
[edit]
user@host# show security ike
ike {
  policy clientpol-group {
    mode aggressive;
    proposal-set compatible;
    pre-shared-key ascii-text
    "$ABC456"; ## SECRET-DATA
  }
  gateway groupgw {
    ike-policy clientpol-group;
    dynamic {
      hostname example.net;
      connections-limit 50;
      ike-user-type group-ike-id;
    }
    external-interface ge-0/0/0.0;
    aaa access-profile radius-profile;
  }
}
[edit]
user@host# show security ipsec
ipsec {
  policy client1vpnPol {
    proposal-set compatible;
  }
  vpn groupvpn {
    ike {
      gateway groupgw;
      ipsec-policy client1vpnPol;
    }
  }
}
[edit]
user@host# show security policies
from-zone untrust to-zone trust {
  policy group-sec-policy {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
```



```
          permit {
            tunnel {
              ipsec-vpn groupvpn;
            }
          }
        }
      }
    }
  }
[edit]
user@host# show security zones
security-zone untrust {
  interfaces {
    ge-0/0/0.0 {
      host-inbound-traffic {
        system-services {
          ike;
          https;
          ping;
          ssh;
        }
      }
    }
  }
}
[edit]
user@host# show security dynamic-vpn
dynamic-vpn {
  access-profile radius-profile;
  clients {
    groupcfg {
      remote-protected-resources {
        10.100.100.0/24;
      }
      remote-exceptions {
        0.0.0.0/0;
        192.0.2.1/24;
        0.0.0.0/32;
      }
      ipsec-vpn groupvpn;
      user {
        chris;
        derek;
      }
    }
  }
}
```

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

Verification

Dynamic VPN tunnels can be monitored with the same commands used to monitor traditional IPsec VPN tunnels. To confirm that the configuration is working properly, perform these tasks:

Verifying IKE Phase 1 Status

Purpose Verify the IKE Phase 1 status of the security associations.

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

Verifying Connected Clients and Assigned Addresses

Purpose Verify that the remote clients and the IP addresses assigned to them are using XAuth.

Action From operational mode, enter the **show security ike active-peer** command.

Verifying IPsec Phase 2 Status

Purpose Verify the IPsec Phase 2 status of the security associations.

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

Verifying Concurrent Connections and Parameters for Each User

Purpose Verify the number of concurrent connections and the negotiated parameters for each user.

Action From operational mode, enter the **show security dynamic-vpn users** command.

Related Documentation

- Understanding Dynamic VPN Tunnel Support on page 851
- Dynamic VPN Configuration Overview on page 855
- Understanding Group and Shared IKE IDs on page 871

Example: Configuring Individual IKE IDs for Multiple Users

Supported Platforms SRX Series

This example shows how to configure individual IKE IDs for multiple users. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

NOTE: When there are a large number of users who need to access the VPN, configuring an individual IKE gateway, IPsec VPN, and a security policy for each user can be cumbersome. The group IKE ID feature allows a number of users to share an IKE gateway configuration, thus reducing the number of VPN configurations required. See "Understanding Group and Shared IKE IDs" on page 871.

Requirements

Before you begin:

- Configure network interfaces on the device. See *Interfaces Feature Guide for Security Devices*.
- Create security zones and assign interfaces to them. See *Understanding Security Zones*.
- If there will be more than two simultaneous user connections, install a Dynamic VPN license in the device. See *Installation and Upgrade Guide*.
- Read "Dynamic VPN Configuration Overview" on page 855.

Overview

The following example shows the configuration for two remote dynamic VPN users. For each user, an IKE policy and gateway, IPsec policy and VPN, and a security policy must be configured (see Table 87 on page 881 and Table 88 on page 882). An external RADIUS server is used to authenticate users and assign IP addresses to clients (see Table 89 on page 883).

Table 87: Client 1 Configuration Parameters

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	client1pol	• Mode: aggressive • Proposal set: compatible • Preshared key: (ASCII) for-client1

Table 87: Client 1 Configuration Parameters *(continued)*

Feature	Name	Configuration Parameters
IKE gateway (Phase 1)	client1gw	• IKE policy reference: client1pol • Dynamic hostname: example.net • External interface: ge-0/0/0.0 • Access profile reference: radius-profile
IPsec policy (Phase 2)	client1vpnPol	Proposal set: compatible
IPsec VPN (Phase 2)	client1vpn	• IKE gateway reference: client1gw • IPsec policy reference: client1vpnPol

Security policy (permits traffic from the untrust zone to the trust zone)	client1-policy	• Match criteria:
		  • source address any
		  • destination address any
		  • application any
		• Permit action: tunnel ipsec-vpn client1vpn

Host inbound traffic		Allow the following types of traffic to the ge-0/0/0.0 interface in the untrust zone: • IKE • HTTPS • ping • SSH
Access profile for remote clients		Access profile reference: radius-profile
Remote clients	cfg1	• IPsec VPN reference: client1vpn • User name reference: derek • Remote protected resources: 10.100.100.0/24 • Remote exceptions: 0.0.0.0/0, 192.0.2.1/24, 0.0.0.0/32

Table 88: Client 2 Configuration Parameters

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	client2pol	• Mode: aggressive • Proposal set: compatible • Preshared key: (ASCII) for-client2
IKE gateway (Phase 1)	client2gw	• IKE policy reference: client2pol • Dynamic hostname: example.net • External interface: ge-0/0/0.0 • Access profile reference: radius-profile
IPsec policy (Phase 2)	client2vpnPol	Proposal set: compatible

Table 88: Client 2 Configuration Parameters *(continued)*

Feature	Name	Configuration Parameters
IPsec VPN (Phase 2)	client2vpn	• IKE gateway reference: client2gw • IPsec policy reference: client2vpnPol

Security policy (permits traffic from the untrust zone to the trust zone)	client2-policy	• Match criteria:
		  ◦ source address any
		  ◦ destination address any
		  ◦ application any
		• Permit action: tunnel ipsec-vpn client2vpn

Host inbound traffic		Allow the following types of traffic to the ge-0/0/0.0 interface in the untrust zone: • IKE • HTTPS • ping • SSH
Access profile for remote clients		Access profile reference: radius-profile
Remote clients	cfg2	• IPsec VPN reference: client2vpn • User name reference: chris • Remote protected resources: 10.100.100.0/24 • Remote exceptions: 0.0.0.0/0, 192.0.2.1/24

Table 89: RADIUS Server User Authentication (Individual IKE ID)

Feature	Name	Configuration Parameters
XAuth profile	radius-profile	• RADIUS is the authentication method used to verify user credentials. • RADIUS server IP address is 10.100.100.250 and the password is secret. • This profile is the default profile for Web authentication.

Configuration

Configuring the XAuth Profile

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 access profile radius-profile authentication-order radius
set access profile radius-profile radius-server 10.100.100.250 secret "$ABC123"
```

set access firewall-authentication web-authentication default-profile radius-profile

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 XAuth profile:

1. Configure the access profile.

```
[edit access]
user@host# set profile radius-profile authentication-order radius
user@host# set profile radius-profile radius-server 10.100.100.250 secret secret
```

2. Configure Web authentication using the XAuth profile.

```
[edit access]
user@host# set firewall-authentication web-authentication default-profile
  radius-profile
```

Results

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

```
[edit]
user@host# show access
  profile radius-profile {
    authentication-order radius;
    radius-server {
      10.100.100.250 secret "$ABC123"; ## SECRET-DATA
    }
  }
    firewall-authentication {
      web-authentication {
        default-profile radius-profile;
      }
    }
```

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

Configuring Client 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 ike policy client1pol mode aggressive
set security ike policy client1pol proposal-set compatible
set security ike policy client1pol pre-shared-key ascii-text "$ABC123"
set security ike gateway client1gw ike-policy client1pol
set security ike gateway client1gw dynamic hostname example.net
```



```
set security ike gateway client1gw external-interface ge-0/0/0.0
set security ike gateway client1gw aaa access-profile radius-profile
set security ipsec policy client1vpnPol proposal-set compatible
set security ipsec vpn client1vpn ike gateway client1gw
set security ipsec vpn client1vpn ike ipsec-policy client1vpnPol
set security policies from-zone untrust to-zone trust policy client1-sec-policy match
  source-address any
set security policies from-zone untrust to-zone trust policy client1-sec-policy match
  destination-address any
set security policies from-zone untrust to-zone trust policy client1-sec-policy match
  application any
set security policies from-zone untrust to-zone trust policy client1-sec-policy then permit
  tunnel ipsec-vpn client1vpn
set security dynamic-vpn access-profile radius-profile
set security dynamic-vpn clients cfg1 remote-protected-resources 10.100.100.0/24
set security dynamic-vpn clients cfg1 remote-exceptions 0.0.0.0/0
set security dynamic-vpn clients cfg1 remote-exceptions 192.0.2.1/24
set security dynamic-vpn clients cfg1 remote-exceptions 0.0.0.0/32
set security dynamic-vpn clients cfg1 ipsec-vpn client1vpn
set security dynamic-vpn clients cfg1 user derek
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services https
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ssh
```

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 dynamic VPN for a single user:

1. Configure the IKE policy.

   ```
   [edit security ike]
   user@host# set policy client1pol mode aggressive
   user@host# set policy client1pol proposal-set compatible
   user@host# set policy client1pol pre-shared-key ascii-text for-client1
   ```

2. Configure the IKE gateway.

   ```
   [edit security ike]
   user@host# set gateway client1gw ike-policy client1pol
   user@host# set gateway client1gw dynamic hostname example.net
   user@host# set gateway client1gw external-interface ge-0/0/0.0
   user@host# set gateway client1gw aaa access-profile radius-profile
   ```

3. Configure IPsec.

   ```
   [edit security ipsec]
   user@host# set policy client1vpnPol proposal-set compatible
   user@host# set vpn client1vpn ike gateway client1gw
   ```

```
user@host# set vpn client1vpn ike ipsec-policy client1vpnPol
```

4. Configure the security policy.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy client1-sec-policy match source-address any
  destination-address any application any
user@host# set policy client1-sec-policy then permit tunnel ipsec-vpn client1vpn
```

5. Configure host inbound traffic.

```
[edit security zones security-zone untrust interfaces ge-0/0/0.0]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services https
user@host# set host-inbound-traffic system-services ping
user@host# set host-inbound-traffic system-services ssh
```

6. Specify the access profile to use with dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set access-profile radius-profile
```

7. Configure the clients who can use the dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set clients cfg1 ipsec-vpn client1vpn
user@host# set clients cfg1 user derek
user@host# set clients cfg1 remote-protected-resources 10.100.100.0/24
user@host# set clients cfg1 remote-exceptions 0.0.0.0/0
user@host# set clients cfg1 remote-exceptions 192.0.2.1/24
user@host# set clients cfg1 remote-exceptions 0.0.0.0/32
```

Results From configuration mode, confirm your configuration by entering the **show security ike**, **show security ipsec**, **show security policies**, **show security zones**, and **show security dynamic-vpn** 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
policy client1pol {
  mode aggressive;
  proposal-set compatible;
  pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
  gateway client1gw {
    ike-policy client1pol;
    dynamic hostname example.net;
    external-interface ge-0/0/0.0;
    aaa access-profile radius-profile;
  }
{edit]
user@host# show security ipsec
policy client1vpnPol {
```



```
  proposal-set compatible;
}
  vpn client1vpn {
    ike {
      gateway client1gw;
      ipsec-policy client1vpnPol;
    }
  }
{edit]
user@host# show security policies
from-zone untrust to-zone trust {
  policy client1-sec-policy {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-vpn client1vpn;
        }
      }
    }
  }
}
{edit]
user@host# show security zones
security-zone untrust {
  interfaces {
    ge-0/0/0.0 {
      host-inbound-traffic {
        system-services {
          ike;
          https;
          ping;
          ssh;
        }
      }
    }
  }
}
{edit]
user@host# show security dynamic-vpn
access-profile radius-profile;
  clients {
    cfg1 {
      remote-protected-resources {
        10.100.100.0/24;
      }
      remote-exceptions {
        0.0.0.0/0;
        192.0.2.1/24;
        0.0.0.0/32;
      }
      ipsec-vpn client1vpn;
```

```
        user {
          derek;
        }
      }
    }
```

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

Configuring Client 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 ike policy client2pol mode aggressive
set security ike policy client2pol proposal-set compatible
set security ike policy client2pol pre-shared-key ascii-text "$ABC456"
set security ike gateway client2gw ike-policy client2pol
set security ike gateway client2gw dynamic hostname example.net
set security ike gateway client2gw external-interface ge-0/0/0.0
set security ike gateway client2gw aaa access-profile radius-profile
set security ipsec policy client2vpnPol proposal-set compatible
set security ipsec vpn client2vpn ike gateway client2gw
set security ipsec vpn client2vpn ike ipsec-policy client2vpnPol
set security policies from-zone untrust to-zone trust policy client2-sec-policy match
  source-address any
set security policies from-zone untrust to-zone trust policy client2-sec-policy match
  destination-address any
set security policies from-zone untrust to-zone trust policy client2-sec-policy match
  application any
set security policies from-zone untrust to-zone trust policy client2-sec-policy then permit
  tunnel ipsec-vpn client1vpn
set security dynamic-vpn access-profile radius-profile
set security dynamic-vpn clients cfg2 remote-protected-resources 10.100.100.0/24
set security dynamic-vpn clients cfg2 remote-exceptions 192.0.2.1/24
set security dynamic-vpn clients cfg2 remote-exceptions 0.0.0.0/32
set security dynamic-vpn clients cfg2 ipsec-vpn client2vpn
set security dynamic-vpn clients cfg2 user chris
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services https
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ssh
```

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 dynamic VPN for a single user:

1. Configure the IKE policy.

   ```
   [edit security ike]
   user@host# set policy client2pol mode aggressive
   user@host# set policy client2pol proposal-set compatible
   user@host# set policy client2pol pre-shared-key ascii-text for-client2
   ```

2. Configure the IKE gateway.

   ```
   [edit security ike]
   user@host# set gateway client2gw ike-policy client2pol
   user@host# set gateway client2gw dynamic hostname example.net
   user@host# set gateway client2gw external-interface ge-0/0/0.0
   user@host# set gateway client2gw aaa access-profile radius-profile
   ```

3. Configure IPsec.

   ```
   [edit security ipsec]
   user@host# set policy client2vpnPol proposal-set compatible
   user@host# set vpn client2vpn ike gateway client2gw
   user@host# set vpn client2vpn ike ipsec-policy client2vpnPol
   ```

4. Configure the security policy.

   ```
   [edit security policies from-zone untrust to-zone trust]
   user@host# set policy client2-sec-policy match source-address any
     destination-address any application any
   user@host# set policy client2-sec-policy then permit tunnel ipsec-vpn client2vpn
   ```

5. Configure host inbound traffic.

   ```
   [edit security zones security-zone untrust interfaces ge-0/0/0.0]
   user@host# set host-inbound-traffic system-services ike
   user@host# set host-inbound-traffic system-services https
   user@host# set host-inbound-traffic system-services ping
   user@host# set host-inbound-traffic system-services ssh
   ```

6. Specify the access profile to use with dynamic VPN.

   ```
   [edit security dynamic-vpn]
   user@host# set access-profile radius-profile
   ```

7. Configure the clients who can use the dynamic VPN.

   ```
   [edit security dynamic-vpn]
   user@host# set clients cfg2 ipsec-vpn client1vpn
   user@host# set clients cfg2 user chris
   user@host# set clients cfg2 remote-protected-resources 10.100.100.0/24
   ```

```
user@host# set clients cfg2 remote-exceptions 192.0.2.1/24
user@host# set clients cfg2 remote-exceptions 0.0.0.0/32
```

Results From configuration mode, confirm your configuration by entering the **show security ike**, **show security ipsec**, **show security policies**, **show security zones**, and **show security dynamic-vpn** 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
policy client2pol {
  mode aggressive;
  proposal-set compatible;
  pre-shared-key ascii-text "$ABC456"; ## SECRET-DATA
}
  gateway client2gw {
    ike-policy client2pol;
    dynamic hostname example.net;
    external-interface ge-0/0/0.0;
    aaa access-profile radius-profile;
  }
[edit]
user@host# show security ipsec
policy client2vpnPol {
  proposal-set compatible;
}
  vpn client2vpn {
    ike {
      gateway client2gw;
      ipsec-policy client2vpnPol;
    }
  }
[edit]
user@host# show security policies
from-zone untrust to-zone trust {
  policy client2-sec-policy {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-vpn client2vpn;
        }
      }
    }
  }
}
[edit]
user@host# show security zones
security-zone untrust {
  interfaces {
```



```
        ge-0/0/0.0 {
          host-inbound-traffic {
            system-services {
              ike;
              https;
              ping;
              ssh;
            }
          }
        }
      }
    }
    [edit]
    user@host# show security dynamic-vpn
    access-profile radius-profile;
      clients {
        cfg2 {
          remote-protected-resources {
            10.100.100.0/24;
          }
          remote-exceptions {
            192.0.2.1/24;
            0.0.0.0/32;
          }
          ipsec-vpn client2vpn;
          user {
            chris;
          }
        }
      }
```

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

Verification

Dynamic VPN tunnels can be monitored with the same commands used to monitor traditional IPsec VPN tunnels. To confirm that the configuration is working properly, perform these tasks:

- Verifying IKE Phase 1 Status on page 891
- Verifying Connected Clients and Assigned Addresses on page 892
- Verifying IPsec Phase 2 Status on page 892
- Verifying Concurrent Connections and Parameters for Each User on page 892

Verifying IKE Phase 1 Status

Purpose Verify the IKE Phase 1 status of the security associations.

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

Verifying Connected Clients and Assigned Addresses

Purpose Verify that the remote clients and the IP addresses assigned to them are using XAuth.

Action From operational mode, enter the **show security ike active-peer** command.

Verifying IPsec Phase 2 Status

Purpose Verify the IPsec Phase 2 status of the security associations.

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

Verifying Concurrent Connections and Parameters for Each User

Purpose Verify the number of concurrent connections and the negotiated parameters for each user.

Action From operational mode, enter the **show security dynamic-vpn users** command.

Related Documentation

- Understanding Dynamic VPN Tunnel Support on page 851
- Dynamic VPN Configuration Overview on page 855
- Understanding Group and Shared IKE IDs on page 871
- Example: Configuring a Group IKE ID for Multiple Users on page 873

PART 11

Monitoring and Improving VPN Traffic Performance

CHAPTER 29

Configuring VPN Monitoring Features

Understanding VPN Alarms and Auditing

Supported Platforms SRX Series, vSRX

Configure the following command to enable security event logging during the initial set up of the device. This feature is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX instances.

set security log cache

The administrators (audit, cryptographic, IDS and security) cannot modify the security event logging configuration if the above command is configured and each administrator role is configured to have a distinct, unique set of privileges apart from all other administrative roles.

Alarms are triggered by a VPN failure. A VPN alarm is generated when the system monitors any of the following audited events:

- **Authentication failures**—You can configure the device to generate a system alarm when the packet authentication failures reaches a specified number.
- **Encryption and decryption failures**—You can configure the device to generate a system alarm when encryption or decryption failures exceed a specified number.

- **IKE Phase 1 and IKE Phase 2 failures**—Internet Key Exchange (IKE) Phase 1 negotiations are used to establish IKE security associations (SAs). These SAs protect the IKE Phase 2 negotiations. You can configure the device to generate a system alarm when IKE Phase 1 or IKE Phase 2 failures exceed a specified number.
- **Self-test failures**—Self-tests are tests that a device runs upon power on or reboot to verify whether security software is implemented correctly on your device.

  Self-tests ensure the correctness of cryptographic algorithms. The Junos-FIPS image performs self-tests automatically upon power-on, and continuously for key-pair generation. In either domestic or FIPS images, self-tests can be configured to be performed according to a defined schedule, upon demand or immediately after key generation.

  You can configure the device to generate a system alarm when a self-test failure occurs.
- **IDP flow policy attacks**—An intrusion detection and prevention (IDP) policy allows you to enforce various attack detection and prevention techniques on network traffic. You can configure the device to generate a system alarm when IDP flow policy violations occur.
- **Replay attacks**—A replay attack is a network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. You can configure the device to generate a system alarm when a replay attack occurs.

The syslog messages are included 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.

Alarms are raised based on syslog messages. Every failure is logged, but an alarm is generated only when a 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.

After appropriate actions have been taken, you can clear the alarm. The alarm remains in the queue until you clear it (or until you reboot the device). To clear the alarm, run the **clear security alarms** command.

Related Documentation

- Example: Setting an Audible Alert as Notification of a Security Alarm on page 897
- Example: Generating Security Alarms in Response to Potential Violations on page 898

Example: Setting an Audible Alert as Notification of a Security Alarm

Supported Platforms SRX Series, vSRX

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 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

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.

Related Documentation

- IPsec VPN Overview on page 3

Example: Generating Security Alarms in Response to Potential Violations

Supported Platforms SRX Series, vSRX

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 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

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 *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.

Related Documentation

- Understanding VPN Alarms and Auditing on page 895
- Example: Setting an Audible Alert as Notification of a Security Alarm on page 897

Understanding VPN Monitoring and DPD

Supported Platforms SRX Series, vSRX

VPN monitoring and dead peer detection (DPD) are features available on SRX Series devices to verify the availability of VPN peer devices. This section compares the operation and configuration of these features.

NOTE: The SRX Series device responds to DPD messages sent by VPN peers even if DPD is not configured on the device. You can configure the SRX Series device to initiate DPD messages to VPN peers. You can also configure DPD and VPN monitoring to operate simultaneously on the same SRX Series device, although the number of peers that can be monitored with either method is reduced.

VPN monitoring is a Junos OS mechanism that monitors only Phase 2 security associations (SAs). VPN monitoring is enabled on a per-VPN basis with the **vpn-monitor** statement at the [**edit security ipsec vpn *vpn-name***] hierarchy level. The destination IP and source interface must be specified. The **optimized** option enables the device to use traffic patterns as evidence of peer liveliness; ICMP requests are suppressed.

VPN monitoring options are configured with the **vpn-monitor-options** statement at the [**edit security ipsec**] hierarchy level. These options apply to all VPNs for which VPN monitoring is enabled. Options you can configure include the interval at which ICMP requests are sent to the peer (the default is 10 seconds) and the number of consecutive ICMP requests sent without receiving a response before the peer is considered unreachable (the default is 10 consecutive requests).

DPD is an implementation of RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*. It operates at the IKE level and monitors the peer based on both IKE and IPsec traffic activity.

DPD is configured on an individual IKE gateway with the **dead-peer-detection** statement at the [**edit security ike gateway *gateway-name***] hierarchy level. You can configure DPD modes of operation. The default (optimized) mode sends DPD messages to the peer if there is no incoming IKE or IPsec traffic within a configured interval after the local device sends outgoing packets to the peer. Other configurable options include the interval at which DPD messages are sent to the peer (the default is 10 seconds) and the number of consecutive DPD messages sent without receiving a response before the peer is considered unavailable (the default is five consecutive requests).

Related Documentation

- Understanding Dead Peer Detection on page 902
- IPsec VPN Overview on page 3

Understanding Dead Peer Detection

Supported Platforms SRX Series, vSRX

Dead peer detection (DPD) is a method that network devices use to verify the current existence and availability of other peer devices.

You can use DPD as an alternative to VPN monitoring. VPN monitoring applies to an individual IPsec VPN, while DPD is configured only in an individual IKE gateway context.

A device performs DPD verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgements (R-U-THERE-ACK messages) from the peer. The device sends an R-U-THERE message only if it has not received any traffic from the peer during a specified DPD interval. If the device receives an R-U-THERE-ACK message from the peer during this interval, it considers the peer alive. If the device receives traffic on the tunnel from the peer, it resets its R-U-THERE message counter for that tunnel, thus starting a new interval. If the device does not receive an R-U-THERE-ACK message during the interval, it considers the peer dead. When the device changes the status of a peer device to be dead, the device removes the Phase 1 security association (SA) and all Phase 2 SAs for that peer.

The following DPD modes are supported on the SRX Series devices:

- Optimized—R-U-THERE messages are triggered if there is no incoming IKE or IPsec traffic within a configured interval after the device sends outgoing packets to the peer. This is the default mode.
- Probe idle tunnel—R-U-THERE messages are triggered if there is no incoming or outgoing IKE or IPsec traffic within a configured interval. R-U-THERE messages are sent periodically to the peer until there is traffic activity. This mode helps in early detection of a downed peer and makes the tunnel available for data traffic.
- Always send—R-U-THERE messages are sent at configured intervals regardless of traffic activity between the peers.

NOTE: **We recommend that the probe idle tunnel mode be used instead of the always-send mode.**

DPD timers are active as soon as the Phase 1 SA is established. The DPD behavior is the same for both IKEv1 and IKEv2 protocols.

You can configure the following DPD parameters:

- The interval parameter specifies the amount of time (expressed in seconds) the device waits for traffic from its peer before sending an R-U-THERE message. The default interval is 10 seconds, with a permissible range of 10 to 60 seconds.
- The threshold parameter specifies the maximum number of times to send the R-U-THERE message without a response from the peer before considering the peer

dead. The default number of transmissions is five times, with a permissible range of 1 to 5 retries.

Note the following considerations before configuring DPD:

- When a DPD configuration is added to an existing gateway with active tunnels, R-U-THERE messages are started without clearing Phase 1 or Phase 2 SAs.
- When a DPD configuration is deleted from an existing gateway with active tunnels, R-U-THERE messages are stopped for the tunnels. IKE and IPsec SAs are not affected.
- Modifying any DPD configuration option such as the mode, interval, or threshold values updates the DPD operation without clearing Phase 1 or Phase 2 SAs.
- If the IKE gateway is configured with DPD and VPN but the option to establish tunnels immediately is not configured, DPD does not initiate Phase 1 negotiation.
- If the IKE gateway is configured with multiple peer IP addresses and DPD but Phase 1 SA fails to be established to the first peer IP address, a Phase 1 SA is attempted with the next peer IP address. DPD is active only after a Phase 1 SA is established.
- If the IKE gateway is configured with multiple peer IP addresses and DPD but DPD fails with the current peer's IP address, the Phase 1 and Phase 2 SAs are cleared and a failover to the next peer IP address is triggered.
- More than one Phase 1 or Phase 2 SA can exist with the same peer because of simultaneous negotiations. In this case, R-U-THERE messages are sent on all Phase 1 SAs. Failure to receive DPD responses for the configured number of consecutive times clears the Phase 1 SA and the associated Phase 2 SA (for IKEv2 only).

Related Documentation

- Understanding VPN Monitoring and DPD on page 901
- Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device on page 257

Understanding VPN Monitoring

Supported Platforms SRX Series, vSRX

VPN monitoring uses ICMP echo requests (or pings) to determine if a VPN tunnel is up. When VPN monitoring is enabled, the security device sends pings through the VPN tunnel to the peer gateway or to a specified destination at the other end of the tunnel. Pings are sent by default at intervals of 10 seconds for up to 10 consecutive times. If no reply is received after 10 consecutive pings, the VPN is considered to be down and the IPsec security association (SA) is cleared.

VPN monitoring is enabled for a specified VPN by configuring the **vpn-monitor** option at the [**edit security ipsec vpn *vpn-name***] hierarchy level. The peer gateway's IP address is the default destination; however, you can specify a different destination IP address (such as a server) at the other end of the tunnel. The local tunnel endpoint is the default source interface, but you can specify a different interface name.

NOTE: VPN monitoring of an externally connected device (such as a PC) is not supported on SRX5400, SRX5600, and SRX5800 devices. The destination for VPN monitoring must be a local interface on the SRX5400, SRX5600, or SRX5800 device.

The VPN monitoring **optimized** option sends pings only when there is outgoing traffic and no incoming traffic through the VPN tunnel. If there is incoming traffic through the VPN tunnel, the security device considers the tunnel to be active and does not send pings to the peer. Configuring the **optimized** option can save resources on the security device because pings are only sent when peer liveliness needs to be determined. Sending pings can also activate costly backup links that would otherwise not be used.

You can configure the interval at which pings are sent and the number of consecutive pings that are sent without a reply before the VPN is considered to be down. These are configured with the **interval** and **threshold** options, respectively, at the [**edit security ipsec vpn-monitor-options**] hierarchy level.

NOTE: VPN monitoring can cause tunnel flapping in some VPN environments if ping packets are not accepted by the peer based on the packet's source or destination IP address.

Related Documentation

- Understanding VPN Monitoring and DPD on page 901

Understanding Global SPI and VPN Monitoring Features

Supported Platforms SRX Series, vSRX

You can monitor and maintain the efficient operation of your VPN using the following global VPN features:

- SPI—Peers in a security association (SA) can become unsynchronized when one of the peers fails. 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.
- VPN monitoring—You can use the global VPN monitoring feature to periodically send Internet Control Message Protocol (ICMP) requests to the peer to determine if the peer is reachable.

Related Documentation

- IPsec VPN Overview on page 3
- Example: Configuring Global SPI and VPN Monitoring Features on page 905

Example: Configuring Global SPI and VPN Monitoring Features

Supported Platforms SRX Series, vSRX

Requirements

Before you begin, understand global SPI and VPN monitoring features. See "Understanding Global SPI and VPN Monitoring Features" on page 904.

Overview

In this example, you configure the device to detect and respond five times to a bad IPsec SPI before deleting the SA and initiating a new one. You also configure the device to monitor the VPN by sending ICMP requests to the peer every 15 seconds, and to declare the peer unreachable after 15 unsuccessful pings.

Configuration

Step-by-Step Procedure

To configure global VPN settings in the CLI editor:

- Specify global VPN settings.

```
[edit]
user@host# set security ike respond-bad-spi 5
user@host# set security ipsec vpn-monitor-options interval 15 threshold 15
```

Related Documentation

- Example: Configuring a Policy-Based VPN on page 206
- Example: Configuring a Route-Based VPN on page 50

Understanding Tunnel Events

Supported Platforms SRX Series, vSRX

When there is a network problem related to a VPN, after the tunnel comes up only the tunnel status is tracked. Many issues can occur before the tunnel comes up. Hence, instead of tracking only the tunnel status, tunnel down issues, or negotiation failures, successful events such as successful IPsec SA negotiations, IPsec rekey, and IKE SA rekeys are now tracked. These events are called tunnel events.

For Phase 1 and Phase 2, negotiation events for a given tunnel are tracked along with the events that occur in external daemons like AUTHD or PKID. When a tunnel event occurs multiple times, only one entry is maintained with the updated time and the number of times that event occurred.

Overall, 16 events are tracked: 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, an event is not stored unless a tunnel is down.

The following special events fall into this category:

- Lifetime in kilobytes expired for IPsec SA
- Hard lifetime of IPsec SA expired
- IPsec SA delete payload received from peer, corresponding IPsec SAs cleared
- Cleared unused redundant backup IPsec SA pairs
- IPsec SAs cleared as corresponding IKE SA deleted

AutoVPN tunnels are created and removed dynamically and consequently tunnel events corresponding to these tunnels are short lived. Sometimes these tunnel events cannot be associated with any tunnel so system logging is used for debugging instead.

Related Documentation

- IPsec VPN Overview on page 3

Understanding IPsec DataPath Verification

Supported Platforms SRX Series, vSRX

Overview

By default, the state of the secure tunnel (st0) interfaces configured in point-to-point mode in route-based VPNs is based on the state of the VPN tunnel. Soon after the IPsec SA is established, routes associated with the st0 interface are installed in the Junos OS forwarding table. In certain network topologies, such as where a transit firewall is located between the VPN tunnel endpoints, IPsec data traffic that uses active routes for an established VPN tunnel on the st0 interface may be blocked by the transit firewall. This can result in traffic loss.

When you enable the IPsec datapath verification, the st0 interface is not brought up and activated until the datapath is verified. The verification is configured with the **set security ipsec vpn *vpn-name* vpn-monitor verify-path** statement for route-based site-to-site and dynamic endpoint VPN tunnels.

If there is a NAT device in front of the peer tunnel endpoint, the IP address of the peer tunnel endpoint is translated 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. This is configured with the **set security ipsec vpn *vpn-name* vpn-monitor verify-path destination-ip** configuration.

VPN Monitor Verify-Path Operation

When IPsec datapath verification is configured, the following actions occur:

1. Upon the establishment of the VPN tunnel, an ICMP request is sent 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, “**V**” is displayed in the VPN Monitoring field in the **show security ipsec security-association detail** command output.

2. The **st0** interface is activated only when a response is received 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 no ICMP response is received from the peer, another ICMP request is sent at the configured VPN monitor interval (the default is 10 seconds) until the VPN monitor threshold (the default is 10 times) is reached.

   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 error is logged 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.

   NOTE: VPN monitor interval and threshold values are configured with **vpn-monitor-options** at the **[edit security ipsec]** hierarchy level.

4. If no ICMP response is received from the peer after the VPN monitor threshold is reached, the established VPN tunnel is brought down and the VPN tunnel is renegotiated.

Caveats

The source interface and destination IP addresses that can be configured for VPN monitor 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, VPN monitoring is automatically activated and used after the st0 interface is brought up. We recommend that you configure the VPN monitor optimized option with the **set security ipsec vpn *vpn-name* vpn-monitor optimized** command whenever you enable IPsec datapath verification.

If a chassis cluster failover occurs during the IPsec datapath verification, the new active node starts the verification again. The st0 interface is not activated until the verification succeeds.

No IPsec datapath verification is performed for IPsec SA rekeys, because the st0 interface state does not change for rekeys.

IPsec datapath verification is not supported on st0 interfaces configured in point-to-multipoint mode that are used with AutoVPN, Auto Discovery VPN, and multiple traffic selectors. VPN monitoring and IPsec datapath verification do not support IPv6 addresses, so IPsec datapath verification cannot be used with IPv6 tunnels.

Related Documentation

- IPsec VPN Overview on page 3

CHAPTER 30

Improving IPsec VPN Traffic Performance

Understanding VPN Session Affinity

Supported Platforms SRX Series

VPN session affinity occurs when a clear-text 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 clear-text and IPsec tunnel session in the same SPU. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices.

Without VPN session affinity, a clear-text 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 clear-text packets to the IPsec tunnel.

By default, VPN session affinity is disabled on SRX Series devices. When VPN session affinity is enabled, a new clear-text session is placed on the same SPU as the IPsec tunnel session. Existing clear-text 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.

Enabling VPN session affinity can improve VPN throughput under the following traffic conditions:

- A number of IPsec tunnels are needed and they are distributed evenly among SPUs. If IPsec tunnels are already concentrated on several SPUs, then enabling VPN session affinity allows all clear-text SPUs to also use those SPUs. This can cause those SPUs to be overutilized while other SPUs might be underutilized.

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
```

- Clear-text sessions passing through the tunnels should be at the highest volume for the longest periods of time as possible. Applying VPN session affinity to clear-text sessions of small volumes and short periods (for example, DNS sessions) will decrease the effect of session affinity and might even have a negative impact on VPN throughput under certain conditions.

NOTE: 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 Releases 12.3X48-D50 and 15.1X49-D90, 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 clear-text 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.

Release History Table

Release	Description
12.3X48-D50	Starting with Junos OS Releases 12.3X48-D50 and 15.1X49-D90, 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).

Related Documentation

- Enabling VPN Session Affinity on page 911
- *SRX5000 Line Devices Processing Overview*
- *Understanding Session Cache*
- *Express Path Overview*
- *Example: Enabling Express Path in Security Policies*
- *Example: Configuring SRX5K-MPC3-100G10G (IOC3) and SRX5K-MPC3-40G10G (IOC3) on an SRX5000 Line Device to Support Express Path*

Enabling VPN Session Affinity

Supported Platforms SRX Series

By default, VPN session affinity is disabled on SRX Series devices. 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.

NOTE: Junos OS Release 15.1X49-D10 introduces session affinity support on the IOCs (SRX5K-MPC [IOC2], SRX5K-MPC3-100G10G [IOC3], and SRX5K-MPC3-40G10G [IOC3]). You can enable session affinity for the IPsec tunnel session on the IOC FPCs.

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.

Related Documentation

- Understanding VPN Session Affinity on page 909
- *Understanding Session Cache*
- *Express Path Overview*

Accelerating the IPsec VPN Traffic Performance

Supported Platforms SRX Series

You can accelerate IPsec VPN performance by configuring the performance acceleration parameter. By default, VPN performance acceleration is disabled on SRX Series devices. Enabling the VPN performance acceleration can improve the VPN throughput with VPN session affinity enabled. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices.

This topic describes how to use the CLI to enable VPN performance acceleration.

NOTE: To enable performance acceleration, you must ensure that cleartext sessions and IPsec tunnel sessions are established on the same Services Processing Unit (SPU). For more information on enabling the session affinity, see “Understanding VPN Session Affinity” on page 909.

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
```

Related Documentation

- Understanding VPN Session Affinity on page 909
- Enabling VPN Session Affinity on page 911
- **ipsec-performance-acceleration (Security Flow) on page 1020**

- *show security flow status*

PART 12

Troubleshooting

CHAPTER 31

Tunnel Events

Tunnel Events

Supported Platforms SRX Series, vSRX

Tunnel events can include successful IPsec SA negotiations, IPsec and IKE SA rekeys, SA negotiation failures, and reasons for a tunnel going down. Tunnel events appear in the output for the **show security ipsec inactive-tunnel**, **show security ipsec inactive-tunnel detail**, and **show security ipsec security-association detail** commands. Table 90 on page 917 lists the tunnel events in alphabetical order. Each event includes a description and the action you can take.

Table 90: IPsec VPN Tunnel Events

Tunnel Event	Description	Action
Bind-interface's address deleted. Existing IPSec SAs cleared	A configuration commit removed the IP address from the st0 interface, which resulted in the clearing of the IPsec SA for VPNs bound to the interface.	Review the VPN setup to determine the need for the IP address on the st0 tunnel interface. Review system logs for the commit change.
Bind-interface's address received. Information updated	A configuration commit changed or added an IP address to the st0 tunnel interface.	No action required.
Bind-interface's family deleted. Existing IPSec SAs cleared	A configuration commit removed the family inet or inet6 from the st0 interface, which resulted in the clearing of the IPsec SA for VPNs bound to the interface.	Verify in the configuration that st0.x has the family inet or inet6 associated with the interface. Review system logs for the commit changes.
Bind-interface's family received. Information updated	A configuration commit added the family inet or inet6 on the st0 interface.	No action required.
Bind-interface's zone received. Information updated	A configuration commit changed or added a security zone on the st0 tunnel interface.	No action required.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
Bind-interface's zone status changed. Existing IPSec SAs cleared	The st0.x interface status changed from Up, which cleared the IPsec SA for VPNs bound to the st0 interface where the status changed.	Review system logs for the the interface status change reason.
CA certificate for configured local certificate not found. Negotiation not initiated/successful	During VPN establishment using PKI certificates, the CA for the local certificate was not found on the device, which resulted in VPN establishment failure.	Verify the **ca-profile** configuration. Verify that the CA certificate is loaded on the device. Reload the CA certificate if necessary.
Certificate has expired. Refer to syslog for more information	An attempt to establish a VPN using PKI certificates failed because the CA or local certificate was expired.	Verify certificate validity dates. Verify the system date and time.
Cleared unused redundant backup IPSec SA pairs.	The IPsec SA count for a tunnel crossed two pairs.	No action required.
Configured local certificate has been revoked. Negotiation not initiated/successful	During a local certificate revocation check using the CRL, the local certificate was revoked or the CRL could not be downloaded to allow the revocation check, which resulted in VPN establishment failure or a failure to initiate the VPN tunnel.	Review system logs or PKI trace options for information about the CRL validation failure. Verify the downloaded CRL. Manually load an updated CRL. Consult the CA administrator about why the certificate is on the CRL. Disable the CRL revocation check.
CRL check failed as CA not reachable. Refer to syslog for more information	During a certificate revocation check using the CRL, the CA server could not be reached or did not respond, which resulted in VPN establishment failure.	Verify that the CA server and the CRL distribution point are reachable.
CRL check failed for a certificate. Refer to syslog for more information	During a certificate revocation check using the CRL, the received peer certificate was revoked or the CRL could not be downloaded to allow the revocation check, which resulted in VPN establishment failure.	Review system logs or PKI trace options for information about the CRL validation failure. Verify the downloaded CRL. Manually load an updated CRL. Consult the CA administrator about why the certificate is on the CRL. Disable the CRL revocation check.
Deactivated tunnel as interface information is not ready on new primary node	During a failover in an SRX300, SRX320, SRX340, SRX345, or SRX550HM chassis cluster, interface information was not available on the new primary node. This event is specific to SRX300, SRX320, SRX340, SRX345, and SRX550HM chassis clusters.	No action required.
DPD detected peer as down. Existing IKE/IPSec SAs cleared.	DPD is enabled and the peer was not reachable for the configured interval and threshold. When this happens, the corresponding IKE and IPsec SAs are cleared, causing the tunnel to flap.	Check peer connectivity. Verify peer gateway connectivity, increase DPD intervals or thresholds, or enable **probe-idle-tunnel**.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
Duplicate IKE/IPSec session detected. Old session cleared	An established peer connected again with different information, such as IP address, username, or IKE ID. This event occurs for AutoVPN, dynamic endpoint, and dialup tunnels only.	No action required.
External-interface's address deleted. Existing IPSec SAs cleared	A configuration commit removed or adjusted the IP address on the IKE external interface, which resulted in the clearing of the IPsec SA for IKE gateways bound to the interface.	Verify that an IP address is assigned to the IKE external interface. Review system logs for the commit change.
External interface's address received. Information updated	A configuration commit changed or added a security zone on the IKE external interface.	No action required.
External-interface's device status changed. Existing IPSec SAs cleared	The IKE gateway external interface status changed from Up, which resulted in clearing of the IPsec SA for all IKE gateways associated with the external interface where the status changed.	Review system logs for the interface status change reason.
External-interface's primary address change triggered clearing of IPSec SA	A configuration commit adjusted the IP address on the IKE external interface, which resulted in clearing of the IPsec SA for IKE gateways bound to the adjusted interface.	Verify the IP address assigned to the external interface. Verify use of the primary setting on the external interface. Review system logs for the commit change.
External-interface's sub-unit status changed. Existing IPSec SAs cleared	The IKE gateway external interface status changed from Up, which resulted in clearing of the IPsec SA for all IKE gateways associated with the external interface where the status changed.	Review system logs for the interface status change reason.
External interface's zone received. Information updated	A configuration commit changed or added a security zone on the IKE external interface.	No action required.
External interface's zone status changed. Existing IPSec SAs cleared	A configuration commit changed the security zone for the IKE external interface, which resulted in the clearing of the IPsec SA for all IKE gateways associated with the changed external interface.	Review system logs for commit changes.
Gateway configuration deletion triggered clearing of IPSec SA	A configuration commit deleted or deactivated the IKE gateway, which resulted in clearing of the IPsec SA.	Review system logs for commit changes.
Group VPN configuration change triggered clearing of IPSec SA	A configuration commit changed the group VPN configuration, which resulted in clearing of the IPsec SA.	Review system logs for commit changes.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
Hard lifetime of IPSec SA expired.	This event is tracked for a tunnel only if there are no more IPsec SAs. Otherwise, this event is tracked in statistics only to avoid multiple events being recorded during rekeys.	If the rekey fails or does not complete before the lifetime expires, this event is recorded and the statistics counter is incremented. If the hard lifetime expires before a rekey occurs, a higher lifetime value is recommended. If a rekey was triggered and failed, there might be some other issue noted in another tunnel event.
Idle timer triggered. Existing IPSec SAs cleared.	**idle-time** is configured at the [**edit security ipsec vpn** ***vpn-name*** **ike**] hierarchy level, and the tunnel was idle for the configured time.	Increase the idle tunnel interval.
IKE SA cleared as lifetime expired	The IKE configured lifetime seconds expired. The default setting is 28,800 seconds. This event does not impact current IPsec SAs.	No action required. You can use DPD to maintain IKE establishment.
IKE SA cleared on backup HA node as requested from primary HA node	The primary chassis cluster node requested that the IKE SA be cleared on the backup node.	Review system logs on the primary node for the IKE SA clear reason.
IKE SA negotiation successfully completed	IKE Phase 1 negotiations were successfully completed.	No action required.
IKE SA rekey successfully completed	When using IKEv2, the IKE SA expired with an established IPsec SA. IKEv2 requires an established IKE SA while an IPsec SA is active.	No action required.
IKE SA UDP port change detected with peer. Existing IPSec SAs cleared.	There was a NAT-T port change, possibly caused by changed ports on the NAT device after the tunnel was established. An IPsec layer UDP packet was received from the peer with a different port for the established tunnel. This event resulted in the clearing of the IPsec SA.	Verify the NAT device behavior that led to the port change.
IKE version mismatch detected	The SRX Series device and the VPN peer attempted to use different IKE versions, which resulted in tunnel establishment failure. The SRX Series device is configured for IKEv1 usage by default.	Adjust the VPN peer to use the same IKE version as the SRX Series; or configure the SRX Series to use the same IKE version in as the peer with **set security ike gateway** ***gateway-name*** **version v1-only** or **set security ike gateway** ***gateway-name*** **version v2-only**.
Initial-Contact received from peer. Stale IKE/IPSec SAs cleared.	Initial contact was received from the peer.	No action required.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
IPSec SA delete payload received from peer, corresponding IPSec SAs cleared.	A peer or remote device sent a delete notification for a given IPsec SA, resulting in the deletion of that particular SA pair. If that SA is the last IPsec SA for that tunnel, the tunnel goes down. This event can occur for various reasons: for example, after a rekey the peer might send a delete for an old SA, or a configuration change triggered on a peer resulted in the clearing of the IPsec SA.	Review peer logs to locate the event that caused the SA deletion request to be sent.
IPSec SA negotiation successfully completed	IPsec Phase 2 negotiations were successfully completed.	No action required.
IPSec SA rekey successfully completed	The IPsec rekey was successfully completed.	No action required.
IPSec SA UDP port change detected with peer. Existing IPSec SAs cleared.	There was a NAT-T port change, possibly caused by changed ports on the NAT device after the tunnel was established. An IPsec layer UDP packet was received from the peer with a different port for the established tunnel. This event resulted in the clearing of the IPsec SA.	Verify the NAT device behavior that led to the port change.
IPSec SAs cleared as corresponding IKE SA deleted.	The IPsec SA was deleted.	No action required.
Key pair not found for configured local certificate. Negotiation failed	During VPN establishment using PKI certificates, a corrupt or missing key-pair file from the local device was detected, which resulted in VPN establishment failure.	Verify configuration of **local-certificate** in the IKE policy. Verify that the key-pair is located in ***/var/db/certs/common/key-pair***. Generate a new key-pair and certificate-request, and load the new certificate.
Lifetime in kilobytes expired for IPSec SA.	The **lifetime-kilobytes** value has expired. Before this event, the soft lifetime triggered a rekey for the IPsec SA. The event is not captured and only a statistics counter is incremented.	If the rekey fails or does not complete before the lifetime expires, this event is recorded and the statistics counter is incremented. If the hard lifetime expires before a rekey occurs, a higher lifetime value is recommended. If a rekey was triggered and failed, there might be some other issue noted in another tunnel event.
Manual next-hop-tunnel configuration change triggered clearing of IPSec SA	A configuration commit changed the next-hop tunnel for the st0 interface, which resulted in the clearing of the IPsec SA for the VPN linked to the changed next-hop tunnel.	Review system logs for commit changes.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
Negotiation failed with error code INVALID_IKE_VERSION received from peer	The peer device rejected an incoming VPN tunnel setup request from the SRX Series device because of mismatched IKE versions, resulting in tunnel establishment failure.	Verify the VPN configuration and VPN peer configuration for IKE version usage. Configure the SRX Series device to use IKEv1 or IKEv2 based on the peer setup by entering **set security ike gateway *gateway-name* version v1-only** or **set security ike gateway *gateway-name* version v2-only**.
Negotiation failed with error code NO_PROPOSAL_CHOSEN received from peer	The VPN peer informed the SRX Series device of VPN failure based on a mismatch of proposals, IKE version, peer gateway match, proxy ID/traffic-selectors, DH groups, or PSK usage.	Review peer logs for the failure reason. Review configurations on the SRX Series device and the peer to ensure that expected VPN attributes match.
Negotiation failed with error code TS_UNACCEPTABLE received from peer	The VPN peer rejected the proxy ID/traffic selector requested by the SRX Series device, which resulted in tunnel establishment failure.	Review peer logs for the rejection reason. For route-based VPNs, verify the configured proxy ID/traffic selector. For policy-based VPNs, verify the source, policy, or application defined in the security policy bound to the VPN.
OCSP revocation check failed as server not reachable. Refer to syslog for more information	During a certificate revocation check using OCSP, the OCSP server could not be contacted, which resulted in VPN establishment failure.	Verify that the OCSP server is reachable. Verify the configured IP address of the OCSP server.
OCSP revocation check failed for a certificate. Refer to syslog for more information	During a certificate revocation check using OCSP, a revoke response was received, which resulted in VPN establishment failure.	Review OCSP server logs for the revocation reason.
Peer proposed phase1 negotiation mode (main/aggressive) does not match with configuration	The IKE negotiation mode configured on the SRX Series device for IKEv1 does not match the peer's proposed mode.	Revise the peer or SRX Series device configuration to match the other device.
Peer proposed phase1 proposal conflicts with local configuration. Negotiation failed	The Phase 1 proposal configured on the SRX Series device does not match the peer's proposal.	Revise the peer or SRX Series device configuration to match the other device.
Peer proposed traffic-selectors are not in configured range	The traffic selector configured on the SRX Series device does not match the peer's proposed traffic selectors.	Revise the peer or SRX Series device configuration to match the other device.
Peer proposed unsupported multiple traffic-selector attributes for a single IPSec SA. Negotiation failed.	During IKEv2 negotiations, the peer device sent a proposal containing multiple traffic selectors for a single VPN tunnel, which resulted in the failure of the VPN tunnel setup.	Review the peer configuration of ACLs or traffic selectors.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
Peer proposed unsupported port range in traffic-selector attribute. Phase 2 negotiation failed	During IPsec negotiation, the peer device sent a traffic selector that contained an unsupported port range, which resulted in the failure of the VPN tunnel setup.	Adjust the peer configuration for the port range setup for the ACLs or traffic selectors.
Peer proposed unsupported protocol in traffic-selector attribute. Phase 2 negotiation failed	During IPsec negotiation, the peer device sent a traffic selector that contained an unsupported protocol, which resulted in the failure of the VPN tunnel setup.	Adjust the peer configuration for the protocol setup for the ACLs or traffic selectors.
Peer's IKE-ID validation failed during negotiation	The received IKE ID did not match the expected IKE ID, which resulted in tunnel establishment failure. The default expected IKE ID is the IP address, peer, or **dynamic** setting configured for the IKE gateway.	Review the VPN peer configuration for the IKE ID the peer is sending. Configure the SRX Series device using **remote-identity** to adjust to the expected IKE ID of the peer.
Proposed peer's IKE-ID does not match with peer's certificate. Negotiation failed	When using PKI certificates, the peer IKE ID value was not in the SAN field of the received certificate, which resulted in VPN establishment failure.	Review the VPN peer and reissue a certificate with an updated SAN based on the IKE ID value. Adjust the VPN peer's IKE ID to match the SAN field of the certificate.
Received use IKEv1 message from peer	The peer device rejected an incoming VPN tunnel setup request from the SRX Series device to use IKEv2 when the peer is configured to use IKEv1, which resulted in tunnel establishment failure.	Adjust the VPN peer setup to use IKEv2, or adjust the SRX Series device's configuration to use IKEv1 by entering **set security ike gateway *gateway-name* version v1-only**.
Requested peer to use IKEv1 instead of IKEv2	The SRX Series device is configured to use IKEv1 by default, and the peer attempted to set up IKE using IKEv2, which resulted in tunnel establishment failure.	Adjust the VPN peer to use IKEv1, or configure the SRX Series device to use IKEv2 by entering **set security ike gateway *gateway-name* version v2-only**.
Security policy change triggered clearing of IPSec SA	A policy-based VPN configuration commit changed security policies bound to the IPsec VPN, which resulted in the clearing of the IPsec SA associated with the changed policies.	Review system logs for commit changes.
Shortcut Tunnel deleted because of inactivity	When using IKEv2 with ADVPN, the device received a shortcut suggestion. However, it did not receive a request from a partner to complete the setup of the shortcut tunnel.	Verify that shortcut tunnel peers can reach each other. Verify that the shortcut partners can exchange UDP500 IKEv2 traffic between them.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
Shortcut Tunnel deleted when idle-time is reached	When using IKEv2 with ADVPN, traffic flowing over the shortcut tunnel fell below the **idle-threshold** for longer than the **idle-time** (default is 5 packets per second for 900 seconds). Traffic continues to flow through the IPsec tunnel to the hub.	If traffic is sporadic, decrease **idle-threshold** and increase **idle-time**. The shortcut tunnel should remain established during times of low traffic throughput.
Tunnel configuration changed. Corresponding IKE/IPSec SAs are deleted	A configuration commit adjusted the IKE/IPsec configuration, which resulted in clearing of the IPsec SA.	Review system logs for commit changes.
Tunnel configuration is deleted. Corresponding IKE/IPSec SAs are deleted	A configuration commit deleted or deactivated the IKE/IPsec configuration, which resulted in clearing of the IPsec SA.	Review system logs for commit changes.
Tunnel deleted on backup HA node as requested from primary HA node	This event is generated on the backup chassis cluster node when the tunnel on the primary node is deleted.	No action required.
Tunnel ID reused for other tunnel on primary node. Cleared stale tunnel	On SRX5400, SRX5600, and SRX5800 chassis clusters, if the tunnel ID becomes out of sync for a given tunnel, the old tunnel is removed on the backup chassis cluster node.	No action required.
Tunnel is ready. Waiting for trigger event or peer to trigger negotiation.	The required configuration is available for peer negotiation. The device is awaiting traffic for tunnel establishment or a tunnel setup request from the peer.	No action required.
Unsupported AH and ESP bundle negotiation request denied	The peer proposed AH and ESP protocols on the same IPsec tunnel, but the SRX Series device does not support this configuration.	Reconfigure the peer for either AH or ESP protocol on the tunnel.
User cleared IKE SA from CLI, corresponding IPSec SAs cleared	The IKE SA was manually cleared using the CLI, which cleared the IKE SA but which does not affect current established IPsec SAs.	No action required.
User cleared IPSec SA from CLI.	A user or an administrator has cleared the IPsec SA manually in the CLI.	No action required.
VPN monitoring detected tunnel as down. Existing IPSec SAs cleared.	VPN monitor is configured for the tunnel and the peer did not respond to VPN monitor keepalive messages, or the peer was not reachable. The corresponding IPsec SAs were cleared.	Check peer connectivity and the VPN monitor destination address.

Table 90: IPsec VPN Tunnel Events *(continued)*

Tunnel Event	Description	Action
Zone change for all interface detected. Existing IPSec SAs cleared	A configuration commit changed the security zone for all interfaces, which resulted in clearing of all device IPsec SAs.	Review system logs for commit changes.

Related Documentation

- show security ipsec inactive-tunnels on page 1231
- show security ipsec security-associations on page 1236

PART 13

Configuration Statements and Operational Commands

CHAPTER 32

Configuration Statements

aaa

Supported Platforms SRX Series, vSRX

Syntax

```
aaa {
  access-profile profile-name;
}
```

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D80.

Description Specify that extended authentication is performed in addition to IKE Phase 1 authentication for remote users trying to access a VPN tunnel. This authentication can be through Extended Authentication (XAuth) or Extensible Authentication Protocol (EAP). Include a previously created access profile, configured with the **edit access profile** statement, to specify the access profile to be used for authentication information.

Options **access-profile *profile-name***—Name of the previously created access profile to reference for authentication information.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

access-profile (Security Dynamic VPN)

Supported Platforms SRX Series

Syntax access-profile *profile-name*;

Hierarchy Level [edit security dynamic-vpn]

Release Information Statement introduced in Junos OS Release 9.5.

Description Specify the access profile to use for Extended Authentication for remote users trying to download the Access Manager. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

access-profile (Security IKE Gateway)

Supported Platforms SRX Series, vSRX

Syntax access-profile *profile-name*;

Hierarchy Level [edit security ike gateway *gateway-name* aaa]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the access profile to use for extended authentication for remote users trying to access a VPN tunnel.

Options ***profile-name*** —Name of the access profile.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

address (Security Group VPN Server IKE Gateway)

Supported Platforms SRX Series, vSRX

Syntax address *ip-address-or-hostname* ;

Hierarchy Level [edit security group-vpn server ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 10.2 for **group-vpn** hierarchy.

Description Specify the IPv4 address of the primary Internet Key Exchange (IKE) gateway. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options *ip-address-or-hostname*—IPv4 address of an IKE gateway.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

address (Security IKE Gateway)

Supported Platforms SRX Series, vSRX

Syntax address [*ip-address-or-hostname*];

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1.

Description Specify the IPv4 or IPv6 address or the hostname of the primary Internet Key Exchange (IKE) gateway and up to four backup gateways.

Options *ip-address-or-hostname*—IPv4 or IPv6 address or hostname of an IKE gateway.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

address-assignment (Access)

Supported Platforms SRX Series

Syntax

```
address-assignment {
  abated-utilization percentage;
  abated-utilization-v6 percentage;
  high-utilization percentage;
  high-utilization-v6 percentage;
  neighbor-discovery-router-advertisement ndra-name;
  pool pool-name {
    family {
      inet {
        dhcp-attributes {
          boot-file boot-file-name;
          boot-server boot-server-name;
          domain-name domain-name;
          grace-period seconds;
          maximum-lease-time (seconds | infinite);
          name-server ipv4-address;
          netbios-node-type (b-node | h-node | m-node | p-node);
          next-server next-server-name;
          option dhcp-option-identifier-code {
            array {
              byte [8-bit-value];
              flag [ false| off |on |true];
              integer [32-bit-numeric-values];
              ip-address [ip-address];
              short [signed-16-bit-numeric-value];
              string [character string value];
              unsigned-integer [unsigned-32-bit-numeric-value];
              unsigned-short [16-bit-numeric-value];
            }
            byte 8-bit-value;
            flag (false | off | on | true);
            integer 32-bit-numeric-values;
            ip-address ip-address;
            short signed-16-bit-numeric-value;
            string character string value;
            unsigned-integer unsigned-32-bit-numeric-value;
            unsigned-short 16-bit-numeric-value;
          }
          option-match {
            option-82 {
              circuit-id match-value {
                range range-name;
              }
              remote-id match-value;
                range range-name;
              }
            }
          }
          propagate-ppp-settings [interface-name];
          propagate-settings interface-name;
```



```
            router ipv4-address;
            server-identifier ip-address;
            sip-server {
                ip-address ipv4-address;
                name sip-server-name;
            }
            tftp-server server-name;
            wins-server ipv4-address;
        }
        host hostname {
            hardware-address mac-address;
            ip-address reserved-address;
        }
        network network address;
        range range-name {
            high upper-limit;
            low lower-limit;
        }
        excluded-range range-name
            high upper-limit;
            low lower-limit;
        }
        xauth-attributes {
            primary-dns ip-address;
            primary-wins ip-address;
            secondary-dns ip-address;
            secondary-wins ip-address;
        }
    }
    inet6 {
        dhcp-attributes {
            dns-server ipv6-address;
            grace-period seconds;
            maximum-lease-time (seconds | infinite);
            option dhcp-option-identifier-code {
                array {
                    byte [8-bit-value];
                    flag [ false| off |on |true];
                    integer [32-bit-numeric-values];
                    ip-address [ip-address];
                    short [signed-16-bit-numeric-value];
                    string [character string value];
                    unsigned-integer [unsigned-32-bit-numeric-value];
                    unsigned-short [16-bit-numeric-value];
                }
                byte 8-bit-value;
                flag (false | off | on | true);
                integer 32-bit-numeric-values;
                ip-address ip-address;
                short signed-16-bit-numeric-value;
                string character string value;
                unsigned-integer unsigned-32-bit-numeric-value;
                unsigned-short 16-bit-numeric-value;
            }
            propagate-ppp-settings [interface-name];
            sip-server-address ipv6-address;
```



```
                sip-server-domain-name domain-name;
            }
            prefix ipv6-network-prefix;
            range range-name {
                high upper-limit;
                low lower-limit;
                prefix-length delegated-prefix-length;
            }
                excluded-range range-name
                    high upper-limit;
                    low lower-limit;
                }
        }
        link pool-name;
    }
}
```

Hierarchy Level [edit access]

Release Information Statement introduced in Junos OS Release 10.4 for SRX300, SRX320, SRX340, SRX345, SRX550HM devices.

Description The address-assignment pool feature enables you to create IPv4 and IPv6 address pools that different client applications can share. For example, multiple client applications, such as DHCPv4 or DHCPv6, can use an address-assignment pool to provide addresses for their particular clients.

Required Privilege Level access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

administrator

Supported Platforms SRX Series, vSRX

Syntax

```
administrator {
  e-mail-address e-mail-address ;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify an administrator e-mail address to which the certificate request is sent.

Options **e-mail-address** ***e-mail-address*** —E-mail address where the certificate request is sent. By default, there is no preset e-mail address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

advpn

Supported Platforms SRX Series, vSRX

Syntax

```
advpn {
  suggester {
    disable;
  }
  partner {
    connection-limit number;
    idle-threshold packets/sec;
    idle-time seconds;
    disable;
  }
}
```

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 12.3X48-D10. The range for the **idle-threshold** option and the range and default value for the **idle-time** option revised in Junos OS Release 12.3X48-D20.

Description Enable Auto Discovery VPN (ADVPN) protocol on the specified gateway.

Options **suggester**—VPN peer that can initiate a shortcut exchange to allow shortcut partners to establish dynamic security associations (SAs) with each other. Specify **disable** to disable this role on the gateway.

NOTE: Both suggester and partner roles are enabled if **advpn** is configured without explicitly configuring **suggester** or **partner** keywords. We do not support suggester and partner roles on the same gateway. You must explicitly configure **disable** with the **suggester** or **partner** keyword to disable that particular role. You cannot disable both suggester and partner roles on the same gateway.

partner—VPN peer that can receive a shortcut exchange suggesting that it should establish dynamic SAs with another peer. Specify **disable** to disable this role on the gateway.

The following options can be configured for the partner role:

connection-limit—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.

NOTE: Reducing the configured **connection-limit** value causes all active shortcut tunnels to be brought down. For example, if

connection-limit is configured as 100 and you later reconfigure the number to 80, all active shortcut tunnels are brought down. Increasing the configured **connection-limit** value does not cause shortcut tunnels to go down.

idle-threshold—Rate, in packets per second, below which the shortcut is brought down.

Range: 3 through 5,000 packets per second.

Default: 5 packets per second.

idle-time—Duration, in seconds, after which the shortcut is deleted if the traffic remains below the **idle-threshold** value.

Range: 60 seconds through 86,400 seconds.

Default: 300 seconds.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Auto Discovery VPN on page 575

algorithm (Security)

Supported Platforms SRX Series, vSRX

Syntax algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);

Hierarchy Level [edit security ipsec vpn *vpn-name* manual encryption]

Release Information Statement modified in Junos OS Release 8.5.

Description Select the encryption algorithm for the internal Routing-Engine-to-Routing-Engine IPsec security association (SA) configuration.

Options

- **3des-cbc**—3DES-CBC encryption algorithm.
- **aes-128-cbc**—AES-CBC 128-bit encryption algorithm.
- **aes-192-cbc**—AES-CBC 192-bit encryption algorithm.
- **aes-256-cbc**—AES-CBC 256-bit encryption algorithm.
- **des-cbc**—DES-CBC encryption algorithm.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

always-send

Supported Platforms SRX Series, vSRX

Syntax always-send;

Hierarchy Level [edit security ike gateway *gateway-name* dead-peer-detection]

Release Information Statement introduced in Junos OS Release 8.5.

Description Instructs the device to send dead peer detection (DPD) requests regardless of whether there is outgoing IPsec traffic to the peer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

authentication (IPsec SA for OSPF)

Supported Platforms SRX Series, vSRX

Syntax

```
authentication {
  algorithm (hmac-md5-96 | hmac-sha1-96);
  key {
    ascii-text key;
    hexadecimal key;
  }
}
```

Hierarchy Level [edit security ipsec security-association *sa-name* manual direction bidirectional]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Configure authentication parameters for a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link.

Options **algorithm**—Hash algorithm that authenticates packet data. It can be one of the following:

- **hmac-md5-96**—Produces a 128-bit digest. This is the default.
- **hmac-sha1-96**—Produces a 160-bit digest.

key—Type of authentication key. It can be one of the following:

- **ascii-text** ***key***—ASCII text key. For **hmac-md5-96**, the key is 16 ASCII characters; for **hmac-sha1-96**, the key is 20 ASCII characters.
- **hexadecimal** ***key***—Hexadecimal key. For **hmac-md5-96**, the key is 32 hexadecimal characters; for **hmac-sha1-96**, the key is 40 hexadecimal characters.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

authentication (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
authentication {
  algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
  key (ascii-text key | hexadecimal key );
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name* manual]

Release Information Statement modified in Junos OS Release 8.5. Support for **hmac-sha-256-128** added to SRX5400, SRX5600, and SRX5800 devices in Junos OS Release 12.1X46-D20.

Description Configure IPsec authentication parameters for a manual security association.

Options

- **algorithm**—Hash algorithm that authenticates packet data. It can be one of the following:
  - **hmac-md5-96**—Produces a 128-bit digest.
  - **hmac-sha-256-128**—Produces a 256-bit digest, truncated to 128 bits.
  - **hmac-sha1-96**—Produces a 160-bit digest.
- **key**—Type of authentication key. It can be one of the following:
  - **ascii-text *key***—ASCII text key. For **hmac-md5-96**, the key is 16 ASCII characters; for **hmac-sha1-96**, the key is 20 ASCII characters.
  - **hexadecimal *key***—Hexadecimal key. For **hmac-md5-96**, the key is 32 hexadecimal characters; for **hmac-sha1-96**, the key is 40 hexadecimal characters.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

authentication-algorithm (Security Group VPN IKE)

Supported Platforms SRX Series, vSRX

Syntax authentication-algorithm (sha-256 | sha-384);

Hierarchy Level [edit security group-vpn member ike proposal *proposal-name*]
[edit security group-vpn server ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure the Internet Key Exchange (IKE) authentication algorithm. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: The device does not delete existing IPsec SAs when you update the **authentication-algorithm** configuration in the IKE proposal. The device deletes existing IPsec SAs when you update the **authentication-algorithm** configuration in the IPsec proposal.

Options **authentication-algorithm**—Hash algorithm that authenticates packet data. It can be one of the following algorithms:

- **sha-256**—Produces a 256-bit digest. This is the default value.
- **sha-384**—Produces a 384-bit digest.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

authentication-algorithm (Security Group VPN IPsec)

Supported Platforms SRX Series, vSRX

Syntax authentication-algorithm hmac-sha-256-128;

Hierarchy Level [edit security group-vpn server ipsec proposal *proposal-name*]

Release Information Statement added in Junos OS Release 10.2.

Description Configure the IPsec authentication algorithm. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **hmac-sha-256-128**—Produces a 256-bit digest, truncated to 128 bits. This is the default value.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

authentication-algorithm (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax authentication-algorithm (md5 | sha-256 | sha-384 | sha1);

Hierarchy Level [edit security ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for the **sha-384** option added in Junos OS Release 12.1X45-D10.

Description Configure the Internet Key Exchange (IKE) authentication algorithm.

NOTE: The device does not delete existing IPsec SAs when you update the **authentication-algorithm** configuration in the IKE proposal. The device deletes existing IPsec SAs when you update the **authentication-algorithm** configuration in the IPsec proposal.

Options **authentication-algorithm**—Hash algorithm that authenticates packet data. It can be one of the following algorithms:

- **md5**—Produces a 128-bit digest.
- **sha-256**—Produces a 256-bit digest.
- **sha-384**—Produces a 384-bit digest.
- **sha1**—Produces a 160-bit digest.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

authentication-algorithm (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax authentication-algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);

Hierarchy Level [edit security ipsec proposal *proposal-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for **hmac-sha-256-128** added to SRX5400, SRX5600, and SRX5800 devices in Junos OS Release 12.1X46-D20.

Description Configure the IPsec authentication algorithm.

Options The hash algorithm to authenticate data can be one of the following:

- **hmac-md5-96**—Produces a 128-bit digest.
- **hmac-sha-256-128**—Produces a 256-bit digest, truncated to 128 bits.
- **hmac-sha1-96**—Produces a 160-bit digest.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

authentication-method

Supported Platforms SRX Series, vSRX

Syntax

```
authentication-method (dsa-signatures | ecdsa-signatures-256 | ecdsa-signatures-384 |
  pre-shared-keys | rsa-signatures);
```

Hierarchy Level [edit security ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **ecdsa-signatures-256** and **ecdsa-signatures-384** options added in Junos OS Release 12.1X45-D10.

Description Specify the method the device uses to authenticate the source of Internet Key Exchange (IKE) messages. The **pre-shared-keys** option refers to a preshared key, which is a key for encryption and decryption that both participants must have before beginning tunnel negotiations. The other options refer to types of digital signatures, which are certificates that confirm the identity of the certificate holder.

NOTE: **The device does not delete existing IPsec SAs when you update the authentication-method configuration in the IKE proposal.**

Options

- **dsa-signatures**—Specify that the Digital Signature Algorithm (DSA) is used.
- **ecdsa-signatures-256**—Specify that the Elliptic Curve DSA (ECDSA) using the 256-bit elliptic curve secp256r1, as specified in the *Federal Information Processing Standard (FIPS) Digital Signature Standard (DSS) 186-3*, is used.
- **ecdsa-signatures-384**—Specify that the ECDSA using the 384-bit elliptic curve secp384r1, as specified in the *FIPS DSS 186-3*, is used.
- **pre-shared-keys**—Specify that a preshared key, which is a secret key shared between the two peers, is used during authentication to identify the peers to each other. The same key must be configured for each peer. This is the default method.
- **rsa-signatures**—Specify that a public key algorithm, which supports encryption and digital signatures, is used.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

authentication-method (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax authentication-method pre-shared-keys;

Hierarchy Level [edit security group-vpn member ike proposal *proposal-name*]
[edit security group-vpn server ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify the method the device uses to authenticate the source of Internet Key Exchange (IKE) messages. The **pre-shared-keys** option refers to a preshared key, which is a key for encryption and decryption that both participants must have before beginning tunnel negotiations. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: **The device does not delete existing IPsec SAs when you update the authentication-method configuration in the IKE proposal.**

Options

- **pre-shared-keys**—Specify that a preshared key, which is a secret key shared between the two peers, is used during authentication to identify the peers to each other. The same key must be configured for each peer. This is the default method.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

auto-re-enrollment (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
auto-re-enrollment {
  cmpv2 {
    certificate-id certificate-id-name {
      ca-profile-name ca-profile-name ;
      re-enroll-trigger-time-percentage percentage ;
      re-generate-keypair;
    }
  }
  scep {
    certificate-id certificate-id-name {
      ca-profile-name ca-profile-name ;
      challenge-password password ;
      re-enroll-trigger-time-percentage percentage ;
      re-generate-keypair;
    }
  }
}
```

Hierarchy Level [edit security pki]

Release Information Statement modified in Junos OS Release 9.0. **cmpv2** and **scep** keywords and options added in Junos OS Release 15.1X49-D40.

Description Configure the automatic reenrollment of a local end-entity (EE) certificate.

Options **cmpv2**—Configure automatic reenrollment of a local certificate using CMPv2.

scep—Configure automatic reenrollment of a local certificate using Simple Certificate Enrollment Protocol (SCEP).

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Understanding Certificates and PKI on page 353

auxiliary-spi (IPsec SA for OSPF)

Supported Platforms SRX Series, vSRX

Syntax auxiliary-spi *auxiliary-spi-value*;

Hierarchy Level [edit security ipsec security-association *sa-name* mode transport manual direction bidirectional]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Configure an auxiliary security parameter index (SPI) for a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link.

Options **auxiliary-spi**—Auxiliary SPI for the manual IPsec SA. The SPI uniquely identifies the SA to use at the receiving host (the destination address in the packet).

Range: 256 through 16639

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding OSPF and OSPFv3 Authentication on SRX Series Devices on page 37

bind-interface

Supported Platforms SRX Series, vSRX

Syntax bind-interface *interface-name*;

Hierarchy Level [edit security ipsec vpn *vpn-name*]

Release Information Statement modified in Junos OS Release 8.5.

Description Configure the tunnel interface to which the route-based virtual private network (VPN) is bound.

Options *interface-name* —Tunnel interface.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

ca-identity (Security)

Supported Platforms SRX Series, vSRX

Syntax ca-identity *ca-identity*;

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement modified in Junos OS Release 11.1.

Description Specify the certificate authority (CA) identity to use in requesting digital certificates.

Options

- ***ca-identity*** —Name of CA identity. This name is typically the domain name of the CA.
- ***routing-instance-name*** —Name of routing instance. The routing instance name is chosen from the list of configured routing instances.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

ca-profile (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax

```
ca-profile ca-profile-name {
  administrator {
    e-mail-address e-mail-address;
  }
  ca-identity ca-identity ;
  enrollment {
      retry number;
      retry-interval seconds;
      url url-name;
  }
  revocation-check {
    disable;
    crl {
      disable {
        on-download-failure;
      }
      refresh-interval hours;
      url url-name;
    }
    disable;
    ocsp {
      connection-failure (disable | fallback-crl);
      disable-responder-revocation-check;
      nonce-payload (enable | disable);
      url ocsp-url;
    }
    use-ocsp;
  }
  routing-instance routing-instance-name ;
  source-address ip-address;
}
```

Hierarchy Level [edit security pki]

Release Information Statement modified in Junos OS Release 8.5. Support for **ocsp** and **use-ocsp** options added in Junos OS Release 12.1X46-D20.

Description Configure certificate authority (CA) profile.

Options ***ca-profile-name*** —Name of a trusted CA.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

ca-profile-name

Supported Platforms SRX Series, vSRX

Syntax ca-profile-name *ca-profile-name*;

Hierarchy Level [edit security pki auto-re-enrollment cmpv2 certificate-id *certificate-id-name*]
[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]

Release Information Statement modified in Junos OS Release 9.0. Support for **[edit security pki auto-re-enrollment cmpv2 certificate-id *certificate-id-name*]** and **[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]** hierarchies added in Junos OS Release 15.1X49-D40.

Description Specify the name of the certificate authority (CA) profile to be used for automatic reenrollment. The CA certificate must be present to initiate reenrollment.

Options ***ca-profile-name*** —Name of the CA profile.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

certificate

Supported Platforms SRX Series, vSRX

Syntax

```
certificate {
  local-certificate certificate-id;
  peer-certificate-type (pkcs7 | x509-signature);
  policy-oids [ oid ];
}
```

Hierarchy Level [edit security ike policy *policy-name*]

Release Information Statement introduced in Junos OS Release 8.5. **policy-oids** option added in Junos OS Release 12.3X48-D10.

Description Specify usage of a digital certificate to authenticate the virtual private network (VPN) initiator and recipient.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

certificate-id (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
certificate-id certificate-id-name {
  ca-profile-name ca-profile-name;
  challenge-password password;
  re-enroll-trigger-time-percentage percentage;
  re-generate-keypair;
}
```

Hierarchy Level

[edit security pki auto-re-enrollment cmpv2]
[edit security pki auto-re-enrollment scep]

Release Information Statement modified in Junos OS Release 9.0. Support for **[edit security pki auto-re-enrollment cmpv2]** and **[edit security pki auto-re-enrollment scep]** hierarchies added in Junos OS Release 15.1X49-D40.

Description Specify the certificate authority (CA) certificate to use for automatic reenrollment.

NOTE: **The challenge-password option is only applicable for SCEP reenrollment.**

Options *certificate-id-name* —Identifier of the end-entity (EE) certificate to be automatically reenrolled. The certificate must be already enrolled for reenrollment to be initiated.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

challenge-password (Security)

Supported Platforms SRX Series, vSRX

Syntax challenge-password *password*;

Hierarchy Level [edit security pki auto-re-enrollment certificate-id *certificate-id-name*]
[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]

Release Information Statement modified in Junos OS Release 9.0. Support for **[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]** hierarchy added in Junos OS Release 15.1X49-D40.

Description Specify the password used by the certificate authority (CA) for enrollment and revocation. If the CA does not provide the challenge password, choose your own password.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

clients (Security)

Supported Platforms SRX Series

Syntax

```
clients configuration-name {
  ipsec-vpn vpn-name;
  remote-exceptions ip-address/mask;
  remote-protected-resources ip-address/mask;
  user username;
  user-groups user-group-name;
}
```

Hierarchy Level [edit security dynamic-vpn]

Release Information Statement introduced in Junos OS Release 9.5.

Description Create a client configuration for the dynamic VPN feature. Within the configuration, specify a name for the configuration, reference a standard VPN configuration to use for IPsec negotiations, specify which resources to protect, define any exceptions, and list the users to which the dynamic VPN configuration applies. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Options ***configuration-name***—Name of the client configuration.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

config-check (Security Dynamic VPN)

Supported Platforms SRX Series

Syntax config-check;

Hierarchy Level [edit security dynamic-vpn]

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Enable extra dynamic VPN configuration checking. If you include this statement in your configuration, it is automatically enabled. If the statement is not present in your configuration, the configuration check option is not enabled. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Dynamic VPN Overview on page 849

connections-limit

Supported Platforms SRX Series, vSRX

Syntax connections-limit *number*;

Hierarchy Level [edit security ike gateway *gateway-name* dynamic]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure the number of concurrent connections that the group profile supports. When the maximum number of connections is reached, no more dynamic virtual private network (VPN) endpoints dialup users attempting to access an IPsec VPN are allowed to begin Internet Key Exchange (IKE) negotiations. This configuration applies to SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, and SRX4200 devices and vSRX instances, and to SRX5400, SRX5600, and SRX5800 devices configured for AutoVPN.

Options *number*—Maximum number of concurrent connections allowed.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

container

Supported Platforms SRX Series, vSRX

Syntax container *container-string*;

Hierarchy Level [edit security ike gateway *gateway-name* dynamic distinguished-name]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify that the value in the identity fields of a dynamic virtual private network (VPN) endpoint user's distinguished name exactly match the values in the group IKE user's distinguished name. The order of the identity fields in the fields of the distinguished name strings must be identical when matching.

Options ***container-string***—Distinguished name identity value to be matched. For example, **cn**=admin, **ou**=eng, **o**=example, **dc**=net.

NOTE: Add a space between each container string. For example, **edit security ike gateway jsr_gateway dynamic distinguished-name container o=example, ou=eng;**

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

crl (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
crl {
  disable {
    on-download-failure;
  }
  refresh-interval hours;
  url url-name;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name* revocation-check]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure the certificate revocation list (CRL). A CRL is a time-stamped list identifying revoked certificates, which is signed by a CA and made available to the participating IPsec peers on a regular periodic basis.

Options

- **disable on-download-failure**—(Optional) Override the default behavior and permit certificate verification even if the CRL fails to download.
- **refresh-interval** *hours*—Time interval, in hours, between CRL updates.

  Range — 0 through 8784 hours.
- **url** ***url-name*** —Name of the location from which to retrieve the CRL through HTTP or Lightweight Directory Access Protocol (LADP). You can specify one URL for each configured CA profile. By default, no location is specified. Use a fully qualified domain name (FQDN) or an IP address and, optionally, a port number. If no port number is specified, port 80 is used for HTTP and port 443 is used for LDAP.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

cryptographic-self-test

Supported Platforms SRX Series, vSRX

Syntax cryptographic-self-test;

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm when the device or switch detects a cryptographic self-test failure. Cryptographic self-tests are a set of preoperational tests that are performed after the device or switch is powered on. The self-tests run without operator intervention.

Default No alarm is raised upon failure of a cryptographic self-test.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

dead-peer-detection

Supported Platforms SRX Series, vSRX

Syntax

```
dead-peer-detection {
  (always-send | optimized | probe-idle-tunnel);
  interval seconds;
  threshold number;
}
```

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for the **optimized** and **probe-idle-tunnel** options added in Junos OS Release 12.1X46-D10.

Description Enable the device to use dead peer detection (DPD). DPD is a method used by devices to verify the current existence and availability of IPsec peers. A device performs this verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgements (R-U-THERE-ACK messages) from the peer.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding AutoVPN on page 461
- IPsec VPN Overview on page 3

dead-peer-detection (Security Group VPN Server)

Supported Platforms SRX Series, vSRX

Syntax

```
dead-peer-detection {
  always-send;
  interval seconds;
  threshold number;
}
```

Hierarchy Level [edit security group-vpn server ike gateway *gateway-name*]

Release Information Support for the Group VPN server added in Junos OS Release 15.1X49-D30 for vSRX. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Description Enable the device to use dead peer detection (DPD). DPD is a method used by devices to verify the current existence and availability of IPsec peers. A device performs this verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgements (R-U-THERE-ACK messages) from the peer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

decryption-failures

Supported Platforms SRX Series, vSRX

Syntax

```
decryption-failures {
  threshold value;
}
```

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm after exceeding a specified number of decryption failures. This statement is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX instances.

Default Multiple decryption failures do not cause an alarm to be raised.

Options ***failures***—Number of decryption failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

Range: 0 through 1 through 1,000,000,000.

Default: 1000

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

description (Security Policies)

Supported Platforms SRX Series, vSRX

Syntax description *description*;

Hierarchy Level
[edit security group-vpn member ike policy *policy-name*]
[edit security group-vpn member ike proposal *proposal-name*]
[edit security group-vpn server ike policy *policy-name*]
[edit security group-vpn server ipsec proposal *proposal-name*]
[edit security group-vpn server ike proposal *proposal-name*]
[edit security ike policy *policy-name*],
[edit security ike proposal *proposal-name*],
[edit security ipsec policy *policy-name*],
[edit security ipsec proposal *proposal-name*]
[edit security polices from-zone *zone-name* to-zone *zone-name* policy *policy-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for **group-vpn** hierarchies added in Junos OS Release 10.2. Support for the **security policies** hierarchy added in Junos OS Release 12.1.

Description Specify descriptive text for an IKE policy, an IPsec policy, an IKE proposal, an IPsec proposal, or a security policy.

Options *description* —Descriptive text about an IKE policy, an IPsec policy, an IKE proposal, an IPsec proposal, or a security policy.

Required Privilege Level
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

destination-ip (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax destination-ip ***ip-address***;

Hierarchy Level [edit security ipsec vpn *vpn-name* vpn-monitor]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the destination of the Internet Control Message Protocol (ICMP) pings. If this statement is used, the device uses the peer's gateway address by default.

Options ***ip-address***—Destination IP address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

df-bit

Supported Platforms SRX Series, vSRX

Syntax df-bit (clear | copy | set);

Hierarchy Level [edit security ipsec vpn *vpn-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify how the device handles the Don't Fragment (DF) bit in the outer header.

NOTE: On SRX5400, SRX5600, and SRX5800 devices, the DF-bit configuration for VPN only works if the original packet size is smaller than the st0 interface MTU, and larger than the external interface-ipsec overhead.

Options

- **clear**—Clear (disable) the DF bit from the outer header. This is the default.
- **copy**—Copy the DF bit to the outer header.
- **set**—Set (enable) the DF bit in the outer header.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

dh-group (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax dh-group (group1 | group14 | group19 | group2 | group20 | group24| group5);

Hierarchy Level [edit security ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for the **group14** option added in Junos OS Release 11.1. Support for **group19**, **group20**, and **group24** options added in Junos OS Release 12.1X45-D10.
Support for **group19** and **group20** options added in Junos OS Release 15.1X49-D70 for vSRX.

Description Specify the IKE Diffie-Hellman group.

NOTE: The device does not delete existing IPsec SAs when you update the dh-group configuration in the IKE proposal.

Options **dh-group**—Diffie-Hellman group for key establishment.

- **group1**—768-bit Modular Exponential (MODP) algorithm.
- **group14**—2048-bit MODP group.
- **group19**—256-bit random Elliptic Curve Groups modulo a Prime (ECP groups) algorithm.
- **group2**—1024-bit MODP algorithm.
- **group20**—384-bit random ECP groups algorithm.
- **group24**—2048-bit MODP Group with 256-bit prime order subgroup.
- **group5**—1536-bit MODP algorithm.

NOTE: We recommend using group14, group19, or group20 instead of group1, group2, or group5.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

dh-group (Security Group VPN IKE)

Supported Platforms SRX Series, vSRX

Syntax dh-group (group14 | group24);

Hierarchy Level [edit security group-vpn member ike proposal *proposal-name*]
[edit security group-vpn server ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 10.2. Support for the **group14** option added in Junos OS Release 11.1. Support for the **group24** option added in Junos OS Release 15.1X49-D30 for vSRX.

Description Specify the IKE Diffie-Hellman group. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: The device does not delete existing IPsec SAs when you update the dh-group configuration in the IKE proposal.

Options **dh-group**—Diffie-Hellman group for key establishment.

- **group14**—2048-bit group. This is the default value.
- **group24**—2048-bit, 256 bit subgroup.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

disable (PKI)

Supported Platforms SRX Series, vSRX

Syntax disable;

Hierarchy Level [edit security pki ca-profile *profile-name* revocation-check]

Release Information Statement modified in Junos OS Release 9.0.

Description Disable revocation checks.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Understanding Certificates and PKI on page 353

distinguished-name (Security)

Supported Platforms SRX Series, vSRX

Syntax distinguished-name <container *container-string*> <wildcard *wildcard-string*>

Hierarchy Level [edit security ike gateway *gateway-name* dynamic]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify a distinguished name as the identifier for the remote gateway with a dynamic IP address.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

dynamic (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
dynamic {
  connections-limit number;
  (distinguished-name <container container-string> <wildcard wildcard-string> | hostname
    domain-name | inet ip-address | inet6 ipv6-address | user-at-hostname e-mail-address);
  ike-user-type (group-ike-id | shared-ike-id);
}
```

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for the **inet6** option added in Junos OS Release 11.1.

Description Specify the identifier for the remote gateway with a dynamic IPv4 or IPv6 address. Use this statement to set up a VPN with a gateway that has an unspecified IPv4 or IPv6 address.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

dynamic (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
dynamic {
  (hostname domain-name | inet ip-address | user-at-hostname e-mail-address);
}
```

Hierarchy Level [edit security group-vpn server ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify the identifier for the remote gateway with a dynamic IPv4 address. Use this statement to set up a VPN with a gateway that has an unspecified IPv4 address. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: Configuring mode main for group VPN servers or members is not supported when the remote gateway has a dynamic address and the authentication method is pre-shared-keys.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

dynamic-vpn

Supported Platforms SRX Series

Syntax

```
dynamic-vpn {
  access-profile profile-name;
  clients configuration-name {
    ipsec-vpn vpn-name;
    remote-exceptions ip-address/mask;
    remote-protected-resources ip-address/mask;
    user username;
    user-groups user-group-name;
  }
  force-upgrade;
  config-check;
  interface;
  traceoptions {
    file filename;
    flag flag;
  }
}
```

Hierarchy Level [edit security]

Release Information Statement introduced in Junos OS Release Release 9.5.

Description Configure the dynamic VPN feature. The dynamic VPN feature simplifies remote access by enabling users to create IPsec VPN tunnels without having to manually configure settings on their PCs or laptops. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

encryption (IPsec SA for OSPF)

Supported Platforms SRX Series, vSRX

Syntax

```
encryption {
  algorithm (3des-cbc | des-cbc | null);
  key {
    ascii-text key;
    hexadecimal key;
  }
}
```

Hierarchy Level [edit security ipsec security-association *sa-name* manual direction bidirectional]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Configure encryption parameters for a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link.

Options **algorithm**—Type of encryption algorithm. It can be one of the following:

- **3des-cbc**—Has block size of 8 bytes (64 bits); its key size is 192 bits long.
- **des-cbc**—Has a block size of 8 bytes (64 bits); its key size is 48 bits long.
- **null**—With null encryption, you are choosing not to provide encryption on OSPFv3 headers.

key—Type of encryption key. It can be one of the following:

- **ascii-text *key***—ASCII text key. For the **des-cbc** option, the key contains 8 ASCII characters; for **3des-cbc**, the key contains 24 ASCII characters.
- **hexadecimal *key***—Hexadecimal key. For the **des-cbc** option, the key contains 16 hexadecimal characters; for the **3des-cbc** option, the key contains 48 hexadecimal characters.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding OSPF and OSPFv3 Authentication on SRX Series Devices on page 37

encryption (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
encryption {
  algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
  key (ascii-text key | hexadecimal key) ;
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name* manual]

Release Information Statement modified in Junos OS Release 8.5.

Description Configure an encryption algorithm and key for a manual Security Association (SA).

Options

- **algorithm**—Type of encryption algorithm. It can be one of the following:
  - **des-cbc**—Has a block size of 8 bytes (64 bits); its key size is 48 bits long.
  - **3des-cbc**—Has block size of 8 bytes (64 bits); its key size is 192 bits long

    NOTE: **For 3des-cbc, we recommend that the first 8 bytes be different from the second 8 bytes, and the second 8 bytes be the same as the third 8 bytes.**

  - **aes-128-cbc**—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
  - **aes-192-cbc**—Advanced Encryption Standard (AES) 192-bit encryption algorithm.
  - **aes-256-cbc**—Advanced Encryption Standard (AES) 256-bit encryption algorithm.
- **key**—Type of encryption key. It can be one of the following:
  - **ascii-text key**—ASCII text key. For the **des-cbc** option, the key contains 8 ASCII characters; for **3des-cbc**, the key contains 24 ASCII characters.
  - **hexadecimal key**—Hexadecimal key. For the **des-cbc** option, the key contains 16 hexadecimal characters; for the **3des-cbc** option, the key contains 48 hexadecimal characters.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

encryption-algorithm (Security Group VPN IKE)

Supported Platforms SRX Series, vSRX

Syntax encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);

Hierarchy Level [edit security group-vpn member ike proposal *proposal-name*]
[edit security group-vpn server ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **group-vpn** hierarchies added in Junos OS Release 10.2.

Description Configure an encryption algorithm for an IKE proposal. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: **The device does not delete existing IPsec SAs when you update the encryption-algorithm configuration in the IKE proposal.**

Options

- **aes-128-cbc**—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- **aes-192-cbc**—AES 192-bit encryption algorithm.
- **aes-256-cbc**—AES 256-bit encryption algorithm.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

encryption-algorithm (Security Group VPN IPsec)

Supported Platforms SRX Series, vSRX

Syntax encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);

Hierarchy Level [edit security group-vpn server ipsec proposal *proposal-name*]
[edit security group-vpn member ipsec proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure an encryption algorithm. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: The device deletes existing IPsec SAs when you update the encryption-algorithm configuration in the IPsec proposal.

Options

- **aes-128-cbc**—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- **aes-192-cbc**—AES 192-bit encryption algorithm.
- **aes-256-cbc**—AES 256-bit encryption algorithm. This is the default value.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

encryption-algorithm (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
encryption-algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-256-cbc |
   aes-256-gcm | des-cbc);
```

Hierarchy Level [edit security ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **aes-128-gcm** and **aes-256-gcm** options added in Junos OS Release 15.1X49-D40.

Description Configure an encryption algorithm for an IKE proposal.

NOTE: **The device does not delete existing IPsec SAs when you update the encryption-algorithm configuration in the IKE proposal.**

Options **3des-cbc**—Has a block size of 24 bytes; the key size is 192 bits long.

aes-128-cbc—Advanced Encryption Standard (AES) 128-bit encryption algorithm.

aes-128-gcm—AES 128-bit authenticated encryption algorithm supported with IKEv2 only. When this option is used, **aes-128-gcm** should be configured at the [**edit security ipsec proposal *proposal-name***] hierarchy level, and the **authentication-algorithm** option should not be configured at the [**edit security ike proposal *proposal-name***] hierarchy level.

aes-192-cbc—AES 192-bit encryption algorithm.

aes-256-cbc—AES 256-bit encryption algorithm.

aes-256-gcm—AES 256-bit authenticated encryption algorithm supported with IKEv2 only. When this option is used, **aes-256-gcm** should be configured at the [**edit security ipsec proposal *proposal-name***] hierarchy level, and the **authentication-algorithm** option should not be configured at the [**edit security ike proposal *proposal-name***] hierarchy level.

des-cbc—Has a block size of 8 bytes; the key size is 48 bits long.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

encryption-algorithm (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax encryption-algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-192-gcm | aes-256-cbc | aes-256-gcm | des-cbc);

Hierarchy Level [edit security ipsec proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **aes-128-gcm**, **aes-192-gcm**, and **aes-256-gcm** options added in Junos OS Release 12.1X45-D10.
Support for **aes-128-gcm**, **aes-192-gcm**, and **aes-256-gcm** options added in Junos OS Release 15.1X49-D70 for vSRX.

Description Configure an encryption algorithm.

NOTE: **The device deletes existing IPsec SAs when you update the encryption-algorithm configuration in the IPsec proposal.**

Options

- **3des-cbc**—Has a block size of 24 bytes; the key size is 192 bits long.
- **aes-128-cbc**—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- **aes-128-gcm**—AES Galois/Counter Mode (GCM) 128-bit encryption algorithm. This option is for IPsec proposals only.
- **aes-192-cbc**—AES 192-bit encryption algorithm.
- **aes-192-gcm**—AES GCM 192-bit encryption algorithm. This option is for IPsec proposals only.
- **aes-256-cbc**—AES 256-bit encryption algorithm.
- **aes-256-gcm**—AES GCM 256-bit encryption algorithm. This option is for IPsec proposals only.
- **des-cbc**—Has a block size of 8 bytes; the key size is 48 bits long.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

encryption-failures

Supported Platforms SRX Series, vSRX

Syntax

```
encryption-failures {
  threshold value;
}
```

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm after exceeding a specified number of encryption failures. This statement is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX instances.

Default Multiple encryption failures do not cause an alarm to be raised.

Options ***failures***—Number of encryption failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

Range: 1 through 1,000,000,000.

Default: 1000

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

enrollment (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
enrollment {
  retry number;
  retry-interval seconds;
  url url-name;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement introduced in Junos OS Release 9.0.

Description Specify the enrollment parameters for a certificate authority (CA).

Options

- **retry** ***number*** —Number of automated attempts for online enrollment to be retried in case enrollment response is pending.

Range: 0 through 1080
Default: 10

- **retry-interval** ***seconds*** —Time interval, in seconds, between the enrollment retries.

Range: 0 through 3600
Default: 900 seconds

- **url** ***url-name*** —Enrollment URL where the Simple Certificate Enrollment Protocol (SCEP) or CMPv2 request is sent to the certification authority (CA) as configured in this profile.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

establish-tunnels

Supported Platforms SRX Series, vSRX

Syntax establish-tunnels (immediately | on-traffic);

Hierarchy Level [edit security ipsec vpn *vpn-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify when IKE is activated: immediately after VPN information is configured and configuration changes are committed, or only when data traffic flows. If this configuration is not specified, IKE is activated only when data traffic flows.

Options

- **immediately**—IKE is activated immediately after VPN configuration changes are committed.

  NOTE: Starting with Junos OS Release 15.1X49-D70, a warning message is displayed if you configure the **establish-tunnels immediately** option at the [**edit security ipsec vpn** *vpn-name*] hierarchy level on AutoVPN hubs with point-to-point tunnel interfaces. Committing the configuration will succeed, however the **establish-tunnels immediately** configuration is ignored. The state of the point-to-point tunnel interface will be up all the time.

  The **establish-tunnels immediately** option is not appropriate for AutoVPN hubs with point-to-point tunnel interfaces because multiple VPN tunnels may be associated with a single AutoVPN configuration.

- **on-traffic**—IKE is activated only when data traffic flows and must to be negotiated with the peer gateway. This is the default behavior.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

external-interface (Security IKE Gateway)

Supported Platforms SRX Series, vSRX

Syntax external-interface *external-interface-name*;

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the outgoing interface for IKE SAs. This interface is associated with a zone that acts as its carrier, providing firewall security for it.

Options ***external-interface-name*** —Name of the interface to be used to send traffic to the IPsec VPN.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

external-interface (Security Manual SA)

Supported Platforms SRX Series, vSRX

Syntax external-interface ***external-interface-name***;

Hierarchy Level [edit security ipsec vpn *vpn-name* manual]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the outgoing interface for the manual SA.

Options ***external-interface-name*** —Name of the outgoing interface.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

fragmentation (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
fragmentation {
  disable;
  size bytes;
}
```

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D80.

Description Disable IKEv2 packet fragmentation and, optionally, configure the maximum size of an IKEv2 message before the message is split into fragments that are individually encrypted and authenticated. On the receiver, the message fragments are collected, verified, decrypted, and merged into the original message. IKEv2 messages larger than the configured maximum are fragmented as long as both VPN peers indicate support for fragmentation in their IKE_SA_INIT exchanges. IKEv2 message fragmentation allows IKEv2 to operate in environments where IP fragments otherwise might be blocked and peers would not be able to establish an IPsec security association.

Options **disable**—Disables IKEv2 fragmentation. IKEv2 fragmentation is enabled by default.

size *bytes*—Maximum size, in bytes, of an IKEv2 message before it is split into fragments. The size applies to both IPv4 and IPv6 messages.

Range: 500 to 1300 bytes

Default: 576 bytes for IPv4 messages and 1280 bytes for IPv6 messages

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding IKEv2 Fragmentation on page 153

gateway (Security Group VPN Member IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
gateway gateway-name {
  ike-policy policy-name;
  local address ip-address;
  local-identity {
    (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname
      e-mail-address);
  }
  remote-identity {
    (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
  }
  routing-instance routing-instance;
  server-address [ip-address];
}
```

Hierarchy Level [edit security group-vpn member ike]

Release Information Statement introduced in Junos OS Release 10.2. Support for the **routing-instance** option added in Junos OS Release 15.1X49-D30 for vSRX.

Description Configure IKE gateway for group VPN member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options ***gateway-name*** —Name of the gateway.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

gateway (Security Group VPN Server IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
gateway gateway-name {
  address ip-address ;
  dead-peer-detection {
    always-send;
    interval seconds;
    threshold number;
  }
  dynamic {
    (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
  }
  ike-policy policy-name;
  local-address ip-address;
  local-identity {
    (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
  }
  remote-identity {
    (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
  }
  routing-instance routing-instance;
}
```

Hierarchy Level [edit security group-vpn server ike]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure IKE gateway for group VPN server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options ***gateway-name*** —Name of the gateway.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

gateway (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
gateway gateway-name {
  aaa {
    access-profile profile-name;
  }
  address [ip-address-or-hostname];
  advpn {
    suggester {
      disable;
    }
    partner {
      connection-limit <number>;
      idle-threshold <packets/sec>;
      idle-time <seconds>;
      disable;
    }
  }
  dead-peer-detection {
    (always-send | optimized | probe-idle-tunnel);
    interval seconds;
    threshold number;
  }
  dynamic {
    connections-limit number;
    (distinguished-name <container container-string> <wildcard wildcard-string> | hostname
      domain-name | inet ip-address | inet6 ipv6-address | user-at-hostname e-mail-address);
    ike-user-type (group-ike-id | shared-ike-id);
  }
  external-interface external-interface-name;
  fragmentation {
    enable:
    size bytes;
  }
  general-ikeid;
  ike-policy policy-name;
  local-address (ipv4-address | ipv6-address);
  local-identity {
    (distinguished-name | hostname hostname | inet ip-address | inet6 ipv6-address |
      user-at-hostname e-mail-address);
  }
  nat-keepalive seconds;
  no-nat-traversal;
  remote-identity {
    (distinguished-name <container container-string> <wildcard wildcard-string> | hostname
      hostname | inet ip-address | inet6 ipv6-address | user-at-hostname e-mail-address);
  }
  tcp-encap-profile profile-name;
  version (v1-only | v2-only);
}
```


Hierarchy Level [edit security ike]

Release Information Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1. The **inet6** option added in Junos OS Release 11.1. Support for the **advpn** option added in Junos OS Release 12.3X48-D10.

Description Configure an IKE gateway.

Options ***gateway-name*** —Name of the gateway.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

gateway (Security IPsec VPN)

Supported Platforms SRX Series, vSRX

Syntax gateway *ip-address*;

Hierarchy Level [edit security ipsec vpn *vpn-name* ike]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the IP address of the peer.

Options *ip-address*—IP address of the peer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

gateway (Security Manual SA)

Supported Platforms SRX Series, vSRX

Syntax gateway *ip-address*;

Hierarchy Level [edit security ipsec vpn *vpn-name* manual]

Release Information Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1.

Description For a manual security association, specify the IPv4 or IPv6 address of the peer.

Options ***ip-address*** —IPv4 or IPv6 address of the peer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

general-ikeid

Supported Platforms SRX Series, vSRX

Syntax general-ikeid;

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Accept general peer IKE ID.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

group (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
group name {
  anti-replay-time-window milliseconds;
  description description;
  group-id number;
  ike-gateway gateway-name;
  ipsec-sa name {
    match-policy policy-name {
      destination ip-address/netmask;
      destination-port number;
      protocol number;
      source ip-address/netmask;
      source-port number;
    }
    proposal proposal-name;
  }
  member-threshold number;
  server-cluster {
    ike-gateway gateway-name;
    retransmission-period seconds;
    server-role (root-server | sub-server);
  }
  server-member-communication {
    certificate certificate-id;
    communication-type unicast;
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
    lifetime-seconds seconds;
    number-of-retransmission number;
    retransmission-period seconds;
    sig-hash-algorithm (sha-256 | sha-384);
  }
}
```

Hierarchy Level [edit security group-vpn server]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure group VPN on the group server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options *name*—Name of the group.

- **anti-replay-time-window** ***milliseconds***—Configure antireplay time in milliseconds. Specify a value from 1 to 60,000.

NOTE: We recommend that NTP be configured on Group VPNv2 devices to ensure proper antireplay operation.

NOTE: Group members that are running on vSRX instances on a host machine where the hypervisor is running under a heavy load may 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.

- **description** ***description***—Description of the group.
- **group-id** ***number***—Identifier for this group VPN. Specify a value from 1 to 4,294,967,295.
- **ike-gateway** ***gateway-name***—Define the group member for Phase 1 negotiation. There can be multiple instances of this option configured. When a group member sends its registration request to the server, the server checks to see that the member is configured for the group.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

group-vpn

Supported Platforms SRX Series, vSRX

Syntax

```
group-vpn {
  member {
    ike {
      gateway gateway-name {
        ike-policy policy-name;
        local address ip-address;
        local-identity {
          (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname
            e-mail-address);
        }
        remote-identity {
          (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
        }
        routing-instance routing-instance;
        server-address [ip-address];
      }
      policy policy-name {
        description description;
        mode (aggressive | main);
        pre-shared-key (ascii-text key | hexadecimal key);
        proposals [proposal-name];
      }
      proposal proposal-name {
        authentication-algorithm (sha-256 | sha-384);
        authentication-method pre-shared-keys;
        description description;
        dh-group (group14 | group24);
        encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
        lifetime-seconds seconds;
      }
      traceoptions {
        file {
          filename;
          files number;
          match regular-expression;
          size maximum-file-size;
          (world-readable | no-world-readable);
        }
        flag flag;
        gateway-filter {
          local-address ip-address;
          remote-address ip-address;
        }
        level (all | error | info | notice | verbose | warning);
        no-remote-trace;
      }
    }
    ipsec {
      vpn vpn-name {
        df-bit (clear | copy | set);
```

```
            exclude rule rule-name {
                source-address ip-address/mask;
                destination-address ip-address/mask;
                application application;
            }
            fail-open rule rule-name {
                source-address ip-address/mask;
                destination-address ip-address/mask;
                application application;
            }
            group id;
            group-vpn-external-interface interface;
            ike-gateway gateway-name;
            recovery-probe;
        }
    }
}
server {
    group name {
        anti-replay-time-window milliseconds;
        description description;
        group-id number;
        ike-gateway gateway-name;
        ipsec-sa name {
            match-policy policy-name {
                destination ip-address/netmask;
                destination-port number;
                protocol number;
                source ip-address/netmask;
                source-port number;
            }
            proposal proposal-name;
        }
        member-threshold number;
        server-cluster {
            ike-gateway gateway-name;
            retransmission-period seconds;
            server-role (root-server | sub-server);
        }
        server-member-communication {
            certificate certificate-id;
            communication-type unicast;
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
            lifetime-seconds seconds;
            number-of-retransmission number;
            retransmission-period seconds;
            sig-hash-algorithm (sha-256 | sha-384);
        }
    }
    ike {
        gateway gateway-name {
            address ip-address ;
            dead-peer-detection {
                always-send;
                interval seconds;
                threshold number;
```



```
            }
            dynamic {
                (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
            }
            ike-policy policy-name;
            local-address ip-address;
            local-identity {
                (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
            }
            remote-identity {
                (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
            }
            routing-instance routing-instance;
        }
        policy policy-name {
            description text;
            mode (aggressive | main);
            pre-shared-key (ascii-text key | hexadecimal key);
            proposals [proposal-name];
        }
        proposal proposal-name {
            authentication-algorithm (sha-256 | sha-384);
            authentication-method pre-shared-keys;
            description description;
            dh-group (group14 | group24);
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
        }
    }
    ipsec {
        proposal proposal-name {
            authentication-algorithm hmac-sha-256-128;
            description description;
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
            lifetime-seconds seconds;
        }
    }
    traceoptions {
        file {
            filename;
            files number;
            match regular-expression;
            size maximum-file-size;
            (world-readable | no-world-readable);
        }
        flag flag;
        gateway-filter {
            local-address ip-address;
            remote-address ip-address;
        }
        level (all | error | info | notice | verbose | warning);
        no-remote-trace;
    }
  }
}
```


Hierarchy Level [edit security]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure Group VPNs in Group VPNv2. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Group VPNv2 Overview on page 665

hostname

Supported Platforms SRX Series, vSRX

Syntax hostname *domain-name*;

Hierarchy Level [edit security ike gateway *gateway-name* dynamic]

Release Information Statement introduced in Junos OS Release 8.5.

Description Unique name by which a network-attached device is known on a network.

Options ***domain-name***—A fully qualified domain name (FQDN).

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

idle-time

Supported Platforms SRX Series, vSRX

Syntax idle-time *seconds*;

Hierarchy Level [edit security ipsec vpn *vpn-name* ike]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the maximum amount of idle time to delete a security association (SA).

Options ***seconds*** —Maximum amount of idle time.

Range: 60 through 999,999 seconds

Default: To be disabled

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

ike (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
ike {
  gateway gateway-name {
    aaa {
      access-profile profile-name;
    }
    address [ip-address-or-hostname];
    advpn {
      suggester {
        disable;
      }
      partner {
        connection-limit <number>;
        idle-threshold <packets/sec>;
        idle-time <seconds>;
        disable;
      }
    }
    dead-peer-detection {
      (always-send | optimized | probe-idle-tunnel);
      interval seconds;
      threshold number;
    }
    dynamic {
      connections-limit number;
      (distinguished-name <container container-string> <wildcard wildcard-string> |
        hostname domain-name | inet ip-address | inet6 ipv6-address | user-at-hostname
        e-mail-address);
      ike-user-type (group-ike-id | shared-ike-id);
    }
    external-interface external-interface-name;
    fragmentation {
      enable:
      size bytes;
    }
    general-ikeid;
    ike-policy policy-name;
    local-address (ipv4-address | ipv6-address);
    local-identity {
      (distinguished-name | hostname hostname | inet ip-address | inet6 ipv6-address |
        user-at-hostname e-mail-address);
    }
    nat-keepalive seconds;
    no-nat-traversal;
    remote-identity {
      (distinguished-name <container container-string> <wildcard wildcard-string> |
        hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname
        e-mail-address);
    }
    tcp-encap-profile profile-name;
    version (v1-only | v2-only);
```

```
  }
  policy policy-name {
    certificate {
      local-certificate certificate-id;
      peer-certificate-type (pkcs7 | x509-signature);
      policy-oids [ oid ];
    }
    description description;
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposal-set (basic | compatible | standard |suiteb-gcm-128 | suiteb-gcm-256);
    proposals [proposal-name];
    reauth-frequency number;
  }
  proposal proposal-name {
    authentication-algorithm (md5 | sha-256 | sha-384| sha1);
    authentication-method (dsa-signatures | ecdsa-signatures-256 | ecdsa-signatures-384
      | pre-shared-keys | rsa-signatures);
    description description;
    dh-group (group1 | group14 | group19 | group2 | group20 | group24 | group5);
    encryption-algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
    lifetime-seconds seconds;
  }
  respond-bad-spi <max-responses>;
  traceoptions {
    file {
      filename;
      files number;
      match regular-expression;
      size maximum-file-size;
      (world-readable | no-world-readable);
    }
    flag flag;
    no-remote-trace;
    rate-limit messages-per-second;
  }
}
```

Hierarchy Level [edit security]

Release Information Statement modified in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1. The **inet6** option added in Junos OS Release 11.1.

Description Define Internet Key Exchange (IKE) configuration.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3
- *ALG Overview*
- *Understanding Logical Systems for SRX Series Services Gateways*

ike (Security Group VPN Member)

Supported Platforms SRX Series, vSRX

Syntax

```
ike {
  gateway gateway-name {
    ike-policy policy-name;
    local address ip-address;
    local-identity {
      (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname
        e-mail-address);
    }
    remote-identity {
      (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
    }
    routing-instance routing-instance;
    server-address [ip-address];
  }
  policy policy-name {
    description description;
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposals [proposal-name];
  }
  proposal proposal-name {
    authentication-algorithm (sha-256 | sha-384);
    authentication-method pre-shared-keys;
    description description;
    dh-group (group14 | group24);
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
    lifetime-seconds seconds;
  }
  traceoptions {
    file {
      filename;
      files number;
      match regular-expression;
      size maximum-file-size;
      (world-readable | no-world-readable);
    }
    flag flag;
    gateway-filter {
      local-address ip-address;
      remote-address ip-address;
    }
    level (all | error | info | notice | verbose | warning);
    no-remote-trace;
  }
}
```

Hierarchy Level [edit security group-vpn member]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure IPsec group VPN on the group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

ike (Security Group VPN Server)

Supported Platforms SRX Series, vSRX

Syntax

```
ike {
  gateway gateway-name {
    address ip-address ;
    dead-peer-detection {
      always-send;
      interval seconds;
      threshold number;
    }
    dynamic {
      (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
    }
    ike-policy policy-name;
    local-address ip-address;
    local-identity {
      (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
    }
    remote-identity {
      (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
    }
    routing-instance routing-instance;
  }
  policy policy-name {
    description text;
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposals [proposal-name];
  }
  proposal proposal-name {
    authentication-algorithm (sha-256 | sha-384);
    authentication-method pre-shared-keys;
    description description;
    dh-group (group14 | group24);
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
  }
}
```

Hierarchy Level [edit security group-vpn server]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure Phase 1 security association (SA) with a member on the group server. The gateway is the group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Group VPNv2 Overview on page 665

ike (Security IPsec VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
ike {
  gateway gateway-name;
  idle-time seconds;
  install-interval seconds;
  ipsec-policy ipsec-policy-name;
  no-anti-replay;
  proxy-identity {
    local ip-prefix;
    remote ip-prefix;
    service (any | service-name);
  }
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1.

Description Define an IKE-keyed IPsec VPN.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

ike-phase1-failures

Supported Platforms SRX Series, vSRX

Syntax

```
ike-phase1-failures {
  threshold value;
}
```

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm after exceeding a specified number of Internet Key Exchange (IKE) Phase 1 failures. This statement is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX instances.

Default Multiple IKE phase 1 failures do not cause an alarm to be raised.

Options ***failures***—Number of IKE phase 1 failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

Range: 1 through 1,000,000,000.

Default: 20

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

ike-phase2-failures

Supported Platforms SRX Series, vSRX

Syntax

```
ike-phase2-failures {
  threshold value;
}
```

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm after exceeding a specified number of Internet Key Exchange (IKE) phase 2 failures. This statement is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX instances.

Default Multiple IKE phase 2 failures do not cause an alarm to be raised.

Options ***failures***—Number of IKE phase 2 failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

Range: 1 through 1,000,000,000.

Default: 20

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

ike-policy (Security Gateway)

Supported Platforms SRX Series, vSRX

Syntax ike-policy *policy-name*;

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the IKE policy to be used for the gateway.

Options ***policy-name*** —IKE policy name.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

ike-user-type

Supported Platforms SRX Series, vSRX

Syntax ike-user-type (group-ike-id | shared-ike-id);

Hierarchy Level [edit security ike gateway *gateway-name* dynamic]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure the type of IKE user for a remote access connection.

Options

- **group-ike-id**—E-mail address or fully qualified domain name (FQDN) shared by a group of remote access users so that each user does not need to configure a separate IKE profile. When group IKE IDs are configured, the IKE ID of each user is a concatenation of a user-specific part and a part that is common to all group IKE ID users. For example, the user Bob might use "Bob.example.net" as his full IKE ID, where ".example.net" is common to all users. The full IKE ID is used to uniquely identify each user connection. Group IKE IDs require the generation of a unique preshared key based on the username supplied during VPN connection, which can be viewed with the **show security ike pre-shared-key** command.
- **shared-ike-id**—E-mail address shared by a large number of remote access users so that each user does not need to configure a separate IKE profile. When a shared IKE ID is configured, all users share a single IKE ID and a single IKE preshared key. Each user is authenticated through the mandatory XAuth phase, where the credentials of individual users are verified either with an external RADIUS server or with a local access database. XAuth is required for shared IKE IDs.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

inet (Security Dynamic Peer)

Supported Platforms SRX Series, vSRX

Syntax inet *ip-address*;

Hierarchy Level [edit security ike gateway *gateway-name* dynamic]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify IP address to identify the dynamic peer.

Options ***ip-address***—IP address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

inet6 (Security IKE Gateway)

Supported Platforms SRX Series, vSRX

Syntax inet6 *ipv6-address*;

Hierarchy Level [edit security ike gateway *gateway-name* dynamic]

Release Information Statement introduced in Junos OS Release 11.1.

Description Specify an IPv6 address to identify the dynamic peer.

Options ***ipv6-address***—IPv6 address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

install-interval

Supported Platforms SRX Series, vSRX

Syntax install-interval *seconds*;

Hierarchy Level [edit security ipsec vpn *vpn-name* ike]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the maximum number of seconds to allow for the installation of a rekeyed outbound security association (SA) on the device.

Options *seconds* —Maximum amount of idle time.

Range: 0 through 10 seconds

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

interval (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax interval *seconds*;

Hierarchy Level [edit security ike gateway *gateway-name* dead-peer-detection]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the amount of time that the peer waits for traffic from its destination peer before sending a dead-peer-detection (DPD) request packet.

Options *seconds* —Number of seconds that the peer waits before sending a DPD request packet.

Range: 10 through 60 seconds

Default: 10 seconds

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

interface (Security Dynamic VPN)

Supported Platforms SRX Series

Syntax interface [***interface-names***];

Hierarchy Level [edit security dynamic-vpn]

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Specify a list of interfaces to set the interfaces that allow access to dynamic VPN. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Options ***interface-names*** —Names of one or more Interfaces that accept dynamic VPN client access, separated by spaces.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

ipsec (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
ipsec {
  policy policy-name {
    description description;
    perfect-forward-secrecy keys (group1 | group14 | group19 | group2 | group20 | group24 |
      group5);
    proposal-set (basic | compatible | standard | suiteb-gcm-128 | suiteb-gcm-256);
    proposals [proposal-name];
  }
  proposal proposal-name {
    authentication-algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
    description description;
    encryption-algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-192-gcm
      | aes-256-cbc | aes-256-gcm | des-cbc);
    lifetime-kilobytes kilobytes;
    lifetime-seconds seconds;
    protocol (ah | esp);
  }
  security-association sa-name {
    manual {
      direction bidirectional {
        authentication {
          algorithm (hmac-md5-96 | hmac-sha1-96);
          key {
            ascii-text key;
            hexadecimal key;
          }
        }
        auxiliary-spi auxiliary-spi-value;
        encryption {
          algorithm (3des-cbc | des-cbc | null);
          key {
            ascii-text key;
            hexadecimal key;
          }
        }
        protocol (ah | esp);
        spi spi-value;
      }
    }
    mode transport;
  }
  traceoptions {
    flag flag;
  }
  vpn vpn-name {
    bind-interface interface-name;
    copy-outer-dscp;
    establish-tunnels (immediately | on-traffic);
    ike {
      gateway gateway-name;
```



```
            idle-time seconds;
            install-interval seconds;
            ipsec-policy ipsec-policy-name;
            no-anti-replay;
            proxy-identity {
                local ip-prefix;
                remote ip-prefix;
                service (any | service-name);
            }
        }
        manual {
            authentication {
                algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
                key (ascii-text key | hexadecimal key);
            }
            encryption {
                algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
                key (ascii-text key | hexadecimal key);
            }
            external-interface external-interface-name;
            gateway ip-address;
            protocol (ah | esp);
            spi spi-value;
        }
        traffic-selector traffic-selector-name {
            local-ip ip-address/netmask;
            remote-ip ip-address/netmask;
        }
        vpn-monitor {
            destination-ip ip-address;
            optimized;
            source-interface interface-name;
            verify-path {
                destination-ip ip-address;
            }
        }
    }
    vpn-monitor-options {
        interval seconds;
        threshold number;
    }
}
```

Hierarchy Level [edit security]

Release Information Statement modified in Junos OS Release 8.5.

Description Define IPsec configuration.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

ipsec (Security Group VPN Member)

Supported Platforms SRX Series, vSRX

Syntax

```
ipsec {
  vpn vpn-name {
    df-bit (clear | copy | set);
    exclude rule rule-name {
      source-address ip-address/mask;
      destination-address ip-address/mask;
      application application;
    }
    fail-open rule rule-name {
      source-address ip-address/mask;
      destination-address ip-address/mask;
      application application;
    }
    group id;
    group-vpn-external-interface interface;
    ike-gateway gateway-name;
    recovery-probe;
  }
}
```

Hierarchy Level [edit security group-vpn member]

Release Information Statement introduced in Junos OS Release 10.2. **df-bit**, **exclude rule**, **fail-open rule**, and **recovery-probe** options added in Junos OS Release 15.1X49-D30 for vSRX.

Description Configure IPsec for Phase 2 exchange on the group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **vpn** *vpn-name*—Name of the VPN.

df-bit—Specifies pre-fragmentation and post-fragmentation of IPsec traffic on the group member. One of the following options can be configured:

- clear—Sets the outer IP do not fragment (DF) bit to 0. When the packet size is larger than the path maximum transmission unit (path MTU), pre-fragmentation is done if the DF bit is not set in the inner packet and post-fragmentation is done if the DF bit is set in the inner packet. This is the default.
- copy—Copies the DF bit from the inner header to the outer header. When the packet size is larger than the path PMTU, pre-fragmentation is done if the DF bit is not set in the inner packet. If the DF bit is set in the inner packet, the packet is dropped and an ICMP message is sent back.
- set—Sets the outer IP DF bit to 1. When the packet size is larger than the path MTU, pre-fragmentation is done if the DF bit is not set in the inner packet. If the DF bit is set in the inner packet, the packet is dropped and an ICMP message is sent back

exclude rule—Specifies traffic to be excluded from Group VPN encryption. A maximum of 10 exclude rules can be configured. Source and destination addresses must be specified in *ip-address/mask* format; address books and address sets are not supported. Predefined and user-defined applications are supported, but application sets are not supported.

fail-open rule—Specifies the traffic to be sent in cleartext mode if there is no valid SA key available to protect the traffic. Traffic that is not specified by the fail-open rule is blocked if there is no valid SA key available to protect the traffic. A maximum of 10 fail-open rules can be configured. Source and destination addresses must be specified in *ip-address/mask* format; address books and address sets are not supported. Predefined and user-defined applications are supported, but application sets are not supported.

group ***id***—Identifier configured for the Group VPN.

group-vpn-external-interface ***interface***—Interface used by the group member to connect to the Group VPN peers. The interface must belong to the same zone as the **to-zone** configured at the [**edit security ipsec-policy**] hierarchy level for Group VPN traffic.

ike-gateway ***gateway-name***—Name of the IKE gateway for the Group VPN.

recovery-probe—Enables initiation of **groupkey-pull** exchanges at specific intervals to update the member's SA from the group server if the group member is determined to be out of synchronization with the group server and other group members. This option is disabled by default.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

ipsec (Security Group VPN Server)

Supported Platforms SRX Series, vSRX

Syntax

```
ipsec {
  proposal proposal-name {
    authentication-algorithm hmac-sha-256-128;
    description description;
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
    lifetime-seconds seconds;
  }
}
```

Hierarchy Level [edit security group-vpn server]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure IPsec proposal for Phase 2 exchange on the group server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **proposal *proposal-name***—Name of the proposal. The proposal name can be up to 32 alphanumeric characters long.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

ipsec-performance-acceleration (Security Flow)

Supported Platforms SRX Series, vSRX

Syntax ipsec-performance-acceleration;

Hierarchy Level [edit security flow]

Release Information Statement introduced in Junos OS Release 12.1X46-D10.

Description Enables IPsec VPN performance acceleration. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices and vSRX instances.

Options None.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3
- *show security flow status*

ipsec-policy (Security)

Supported Platforms SRX Series, vSRX

Syntax ipsec-policy *ipsec-policy-name*;

Hierarchy Level [edit security ipsec vpn *vpn-name* ike]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the IPsec policy name.

Options *ipsec-policy-name* —Name of the IPsec policy.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

ipsec-policy (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax ipsec-policy from-zone *zone-name* to-zone *zone-name* ipsec-group-vpn *vpn-name*;

Hierarchy Level [edit security]

Release Information Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Specifies that matching traffic is checked against rules associated with the specified Group VPN. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. Exclude and fail-open rules are configured at the [**edit security group-vpn member ipsec vpn** ***vpn-name***] hierarchy level.

Options **from-zone** ***zone-name***—Specify the incoming zone for Group VPN traffic.

to-zone ***zone-name***—Specify the outgoing zone for Group VPN traffic.

NOTE: The **to-zone** zone must include the interface configured with the **group-vpn-external-interface** option at the [**edit security group-vpn member ipsec vpn** ***vpn-name***] hierarchy level.

ipsec-group-vpn ***vpn-name***—Specify the Group VPN to which the traffic applies. Only one Group VPN can be referenced by a specific from-zone/to-zone pair.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Traffic Steering
- Group VPNv2 Overview on page 665

ipsec-vpn (Security Dynamic VPNs)

Supported Platforms SRX Series

Syntax ipsec-vpn *vpn-name*;

Hierarchy Level [edit security dynamic-vpn clients *vpn-name*]

Release Information Statement introduced in Junos OS Release 9.5.

Description Use this statement to specify which IPsec VPN configuration the dynamic VPN feature should use to secure traffic. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

ipsec-sa (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
ipsec-sa name {
  match-policy policy-name {
    destination ip-address/netmask;
    destination-port number;
    protocol number;
    source ip-address/netmask;
    source-port number;
  }
  proposal proposal-name;
}
```

Hierarchy Level [edit security group-vpn server group *name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure the group SAs to be downloaded to members. There can be multiple group SAs downloaded to group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **ipsec-sa** ***name***—Define the group SAs to be downloaded to members.

- **match-policy** ***policy-name***—Configure the group policy with source address, source port, destination address, destination port, and protocol.

  Use **0.0.0.0** to specify any source or destination. Use **0** to specify any source port, destination port, or protocol.

- **proposal** ***proposal-name***—Specify the name of the IPsec proposal configured with the **proposal** configuration statement at the [**edit security group-vpn server ipsec**] hierarchy.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

ipsec-vpn (Security Flow)

Supported Platforms SRX Series, vSRX

Syntax

```
ipsec-vpn {
  mss value;
}
```

Hierarchy Level [edit security flow tcp-mss]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the TCP maximum segment size (TCP MSS) for the TCP packets that are about to go into an IPsec VPN tunnel. This value overrides the value specified in the **all-tcp-mss** statement.

Options **mss *value***—TCP MSS value for TCP packets entering an IPsec VPN tunnel. Value is optional.

Range: 64 through 65,535 bytes

Default: 1320 bytes

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

key-generation-self-test

Supported Platforms SRX Series, vSRX

Syntax key-generation-self-test;

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm when the device or switch detects a key generation self-test failure. Key generation is the process of generating keys for cryptography. A key is used to encrypt and decrypt data. The self-tests run without operator intervention.

Default No alarm is raised upon failure of a key generation self-test.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

lifetime-kilobytes

Supported Platforms SRX Series, vSRX

Syntax lifetime-kilobytes *kilobytes*;

Hierarchy Level [edit security ipsec proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the lifetime (in kilobytes) of an IPsec security association (SA).

Options ***kilobytes*** —Lifetime of the IPsec security association (SA). If this statement is not configured, the number of kilobytes used for the SA lifetime is unlimited.

Range: 64 through 1,048,576 kilobytes

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

lifetime-seconds (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax lifetime-seconds *seconds*;

Hierarchy Level [edit security group-vpn member ike proposal *proposal-name*]
[edit security group-vpn server ipsec proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify the lifetime (in seconds) of an IKE or IPsec security association (SA) for group VPN. When the SA expires, it is replaced by a new SA and security parameter index (SPI) or terminated. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options ***seconds***—Lifetime of the IKE or IPsec SA.

Range: 180 through 86,400 seconds

Default: 3600 seconds

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

lifetime-seconds (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax lifetime-seconds *seconds*;

Hierarchy Level [edit security ike proposal *proposal-name*]

Release Information Statement introduced in Junos OS Release 8.5. Default value modified in Junos OS Release 10.2.

Description Specify the lifetime (in seconds) of an IKE security association (SA). When the SA expires, it is replaced by a new SA and security parameter index (SPI) or terminated.

Options *seconds*—Lifetime of the IKE SA.

Range: 180 through 86,400 seconds

Default: 28,800 seconds

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3
- *Understanding User Authentication Methods*

lifetime-seconds (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax lifetime-seconds ***seconds*** ;

Hierarchy Level [edit security ipsec proposal ***proposal-name***]

Release Information Statement introduced in Junos OS Release 8.5. Default value modified in Junos OS Release 10.2.

Description Specify the lifetime (in seconds) of an IPsec security association (SA). When the SA expires, it is replaced by a new SA and security parameter index (SPI) or terminated.

Options ***seconds***—Lifetime of the IPsec SA.

Range: 180 through 86,400 seconds

Default: 3600 seconds

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

load-distribution

Supported Platforms SRX Series, vSRX

Syntax

```
load distribution {
  session-affinity ipsec;
}
```

Hierarchy Level [edit security flow]

Release Information Statement introduced in Junos OS Release 11.4R5.

Description Enable load distribution for a data flow. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices and vSRX instances.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

local (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax local *ip-prefix*;

Hierarchy Level [edit security ipsec vpn *vpn-name* ike proxy-identity]

Release Information Statement modified in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1.

Description Specify the local IPv4 or IPv6 address and subnet mask for the proxy identity.

Options ***ip-prefix*** —IPv4 or IPv6 address and subnet mask.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

local-address

Supported Platforms SRX Series, vSRX

Syntax local-address (*ipv4-address* | *ipv6-address*);

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 12.1X46-D10.

Description Specify the local gateway address. Multiple addresses in the same address family can be configured on an external physical interface to a VPN peer. If this is the case, we recommend that **local-address** be configured. If there is only one IPv4 and one IPv6 address configured on an external physical interface, **local-address** configuration is not necessary.

NOTE: The **local-address** value must be an IP address that is configured on an interface on the SRX Series device. 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.

Options ***ipv4-address***—IPv4 address for the local gateway.

ipv6-address—IPv6 address for the local gateway.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

local-address (Security Group VPN Member)

Supported Platforms SRX Series, vSRX

Syntax local–address *ip-address*;

Hierarchy Level [edit security group-vpn member ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure the IP address the member uses when accessing the group server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options *ip-address*—IPv4 address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

local-address (Security Group VPN Server)

Supported Platforms SRX Series, vSRX

Syntax local-address *ip-address*;

Hierarchy Level [edit security group-vpn server ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Configure the source IP address the group VPN server uses when communicating with a group member or a root-server. This statement is normally used when there are multiple IP addresses bound to an interface. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **local-address *ip-address***—Specify an IPv4 address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

local-certificate (Security)

Supported Platforms SRX Series, vSRX

Syntax local-certificate *certificate-id*;

Hierarchy Level [edit security ike policy *policy-name* certificate]

Release Information Statement modified in Junos OS Release 8.5.

Description Specify a particular certificate when the local device has multiple loaded certificates.

NOTE: The device deletes existing IKE and IPsec SAs when you update the local-certificate configuration in the IKE policy.

Options *certificate-id* —Name of the specific certificate to be used.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

local-identity

Supported Platforms SRX Series, vSRX

Syntax

```
local-identity {
  (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname
    e-mail-address);
}
```

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 8.5. The **inet6** option added in Junos OS Release 11.1.

Description Specify the local IKE identity to send in the exchange with the destination peer to establish communication. If you do not configure a local-identity, the device uses the IPv4 or IPv6 address corresponding to the local endpoint by default.

NOTE: For Network Address Translation Traversal (NAT-T), both local identity and remote identity must be configured.

Options

- **hostname *hostname***—Specify identity as a fully qualified domain name (FQDN).
- **inet *ip-address***—Specify identity as an IPv4 address.
- **user-at-hostname *e-mail-address***—Specify identity as an e-mail address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

local-identity (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
local-identity {
  (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname
    e-mail-address);
}
```

Hierarchy Level

[edit security group-vpn member ike gateway *gateway-name*]
[edit security group-vpn server ike gateway *gateway-name*]

Release Information Support for **group-vpn** hierarchies added in Junos OS Release 10.2.

Description Specify the local IKE identity to send in the exchange with the destination peer to establish communication. If you do not configure a local-identity, the device uses the IPv4 corresponding to the local endpoint by default. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- **hostname *hostname***—Specify identity as a fully qualified domain name (FQDN).
- **inet *ip-address***—Specify identity as an IPv4 address.
- **user-at-hostname *e-mail-address***—Specify identity as an e-mail address.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

manual (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
manual {
  authentication {
    algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
    key (ascii-text key | hexadecimal key );
  }
  encryption {
    algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
    key (ascii-text key | hexadecimal key );
  }
  external-interface external-interface-name;
  gateway ip-address;
  protocol (ah | esp);
  spi spi-value ;
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1.

Description Define a manual IPsec security association (SA).

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

member (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
member {
  ike {
    gateway gateway-name {
      ike-policy policy-name;
      local address ip-address;
      local-identity {
        (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname
          e-mail-address);
      }
      remote-identity {
        (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
      }
      routing-instance routing-instance;
      server-address [ip-address];
    }
    policy policy-name {
      description description;
      mode (aggressive | main);
      pre-shared-key (ascii-text key | hexadecimal key);
      proposals [proposal-name];
    }
    proposal proposal-name {
      authentication-algorithm (sha-256 | sha-384);
      authentication-method pre-shared-keys;
      description description;
      dh-group (group14 | group24);
      encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
      lifetime-seconds seconds;
    }
    traceoptions {
      file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
      }
      flag flag;
      gateway-filter {
        local-address ip-address;
        remote-address ip-address;
      }
      level (all | error | info | notice | verbose | warning);
      no-remote-trace;
    }
  }
  ipsec {
    vpn vpn-name {
      df-bit (clear | copy | set);
      exclude rule rule-name {
```



```
            source-address ip-address/mask;
            destination-address ip-address/mask;
            application application;
          }
          fail-open rule rule-name {
            source-address ip-address/mask;
            destination-address ip-address/mask;
            application application;
          }
          group id;
          group-vpn-external-interface interface;
          ike-gateway gateway-name;
          recovery-probe;
        }
      }
    }
```

Hierarchy Level [edit security group-vpn]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure group VPN member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options Configure group VPN member. You configure the following on the group member:

- Phase 1 IKE SA with the group server. The IKE gateway is the group server.

> NOTE: We recommend that you do not change the default value for **lifetime-seconds** for the IKE proposal on the member. Increasing the value might cause the member device to continue to use the existing Phase 1 IKE SA key even in the event of a crash; this can delay the recovery process.

- IPsec group VPN.

> NOTE: A scope policy must also be configured on the group member. To configure a scope policy, use the **policies** configuration statement at the [**edit security**] hierarchy and specify the IPsec group VPN for the **ipsec-group-vpn** option.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

member-threshold (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax member-threshold *number*;

Hierarchy Level [edit security group-vpn server group *group-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Specify the maximum number of group VPN members that can be accepted in the group. The same **member-threshold** value must be configured on the root-server and all sub-servers in a group server cluster. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **member-threshold** ***number***—Specify the maximum number of group VPN members that can be accepted in the group. There is no default number.

Range: 1 to 2,000.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Server Clusters on page 714

mode (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax mode (aggressive | main);

Hierarchy Level [edit security group-vpn member ike policy *policy-name*]
[edit security group-vpn server ike policy *policy-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **group-vpn** hierarchies added in Junos OS Release 10.2.

Description Define the mode used for Internet Key Exchange (IKE) Phase 1 negotiations. Use aggressive mode only when you need to initiate an IKE key exchange without ID protection, as when a peer unit has a dynamically assigned IP address. (The **main** option is not supported on dynamic VPN implementations.) Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE:

- IKEv2 protocol does not negotiate using mode configuration.
- The device deletes existing IKE and IPsec SAs when you update the mode configuration in the IKE policy.

Options

- **aggressive**—Aggressive mode.
- **main**—Main mode. Main mode is the recommended key-exchange method because it conceals the identities of the parties during the key exchange.

NOTE: Configuring **mode main** for group VPN servers or members is not supported when the remote gateway has a dynamic address and the authentication method is **pre-shared-keys**.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

mode (Security IKE Policy)

Supported Platforms SRX Series, vSRX

Syntax mode (aggressive | main);

Hierarchy Level [edit security ike policy *policy-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Define the mode used for Internet Key Exchange (IKE) Phase 1 negotiations. Use aggressive mode only when you need to initiate an IKE key exchange without ID protection, as when a peer unit has a dynamically assigned IP address.

NOTE:

- IKEv2 protocol does not negotiate using mode configuration.
- The device deletes existing IKE and IPsec SAs when you update the **mode** configuration in the IKE policy.

Options

- **aggressive**—Aggressive mode.
- **main**—Main mode. Main mode is the recommended key-exchange method because it conceals the identities of the parties during the key exchange.

NOTE: Configuring **mode main** for group VPN servers or members is not supported when the remote gateway has a dynamic address and the authentication method is **pre-shared-keys**.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

nat-keepalive

Supported Platforms SRX Series, vSRX

Syntax nat-keepalive *seconds*;

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 8.5. Default value changed from 5 seconds to 20 seconds in Junos OS Release 12.1X46-D10.

Description Specify the interval at which NAT keepalive packets can be sent so that NAT translation continues.

Options *seconds* —Maximum interval in seconds at which NAT keepalive packets can be sent.

Range: 1 through 300 seconds.

Default: 20 seconds.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

no-anti-replay (Security)

Supported Platforms SRX Series, vSRX

Syntax no-anti-replay;

Hierarchy Level [edit security ipsec vpn *vpn-name* ike]

Release Information Statement introduced in Junos OS Release 8.5.

Description Disable the antireplay checking feature of IPsec. By default, antireplay checking is enabled.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

no-nat-traversal

Supported Platforms SRX Series, vSRX

Syntax no-nat-traversal;

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Disables UDP encapsulation of IPsec Encapsulating Security Payload (ESP) packets, otherwise known as Network Address Translation Traversal (NAT-T). NAT-T is enabled by default.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

non-cryptographic-self-test

Supported Platforms SRX Series, vSRX

Syntax non-cryptographic-self-test;

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm when the device or switch detects a noncryptographic self-test failure. The self-tests run without operator intervention.

Default No alarm is raised upon failure of a noncryptographic self-test.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

ocsp (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax

```
ocsp {
  connection-failure (disable | fallback-crl);
  disable-responder-revocation-check;
  nonce-payload (enable | disable);
  url ocsp-url;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name* revocation-check]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Configure Online Certificate Status Protocol (OCSP) to check the revocation status of a certificate.

Options

connection-failure—(Optional) Specify action to take if there is a connection failure to the OCSP responder. If this option is not configured and there is no response from the OCSP responder, certificate validation will fail.

disable—Skip the revocation check if the OCSP responder is not reachable.

fallback-crl—Use CRL to check the revocation status of the certificate.

disable-responder-revocation-check —(Optional) Disable revocation check for the CA certificate received in an OCSP response. The certificates received in an OCSP response generally have shorter lifetimes and revocation check is not required.

nonce-payload—(Optional) Send a nonce payload to prevent replay attack. A nonce payload is sent by default unless it is explicitly disabled. If enabled, the SRX Series device expects OCSP responses to contain a nonce payload, otherwise the revocation check will fail. If OCSP responders are not capable of responding with a nonce payload, disable this option.

disable—Explicitly disable the sending of a nonce payload.

enable—Enable the sending of a nonce payload. This is the default.

url *ocsp-url*—Specify HTTP addresses for OCSP responders. A maximum of two HTTP URL addresses can be configured. If the configured URLs are not reachable, or URLs are not configured, the URL from the certificate being verified is checked.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

optimized

Supported Platforms SRX Series, vSRX

Syntax optimized;

Hierarchy Level [edit security ipsec vpn *vpn-name* vpn-monitor]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify that VPN monitoring optimization is enabled for the VPN object. When VPN monitoring optimization is enabled, the SRX Series device only sends ICMP echo requests (pings) when there is outgoing traffic and no incoming traffic from the configured peer through the VPN tunnel. If there is incoming traffic through the VPN tunnel, the SRX Series device considers the tunnel to be active and does not send pings to the peer.

Because ICMP echo requests are only sent when needed to determine peer liveliness, VPN monitoring optimization can save resources on the SRX Series device. Also, ICMP echo requests can activate costly backup links that would otherwise not be used.

This option is disabled by default.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

optimized (DPD)

Supported Platforms SRX Series, vSRX

Syntax optimized;

Hierarchy Level [edit security ike gateway *gateway-name* dead-peer-detection]

Release Information Statement introduced in Junos OS Release 12.1X46-D10.

Description Send dead peer detection (DPD) messages if there is no incoming IKE or IPsec traffic within the configured interval after outgoing packets are sent to the peer. This is the default DPD mode.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

peer-certificate-type

Supported Platforms SRX Series, vSRX

Syntax peer-certificate-type (pkcs7 | x509-signature);

Hierarchy Level [edit security ike policy *policy-name* certificate]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify a preferred type of certificate (PKCS7 or X509).

Options

- **pkcs7**—Public-Key Cryptography Standard #7.
- **x509-signature**—X509 is an ITU-T standard for public key infrastructure. This is the default value.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

perfect-forward-secrecy (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
perfect-forward-secrecy keys (group1 | group14 | group19 | group2 | group20 | group24 |
    group5);
```

Hierarchy Level [edit security ipsec policy *policy-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **group14** options added in Junos OS Release 11.1. Support for **group19**, **group20**, and **group24** options added in Junos OS Release 12.1X45-D10.

Description Specify Perfect Forward Secrecy (PFS) as the method that the device uses to generate the encryption key. PFS generates each new encryption key independently from the previous key.

NOTE: The device deletes existing IPsec SAs when you update the perfect-forward-secrecy configuration in the IPsec policy.

Options

- **group1**—Diffie-Hellman Group 1.
- **group14**—Diffie-Hellman Group 14.
- **group19**—Diffie-Hellman Group 19.
- **group2**—Diffie-Hellman Group 2.
- **group20**—Diffie-Hellman Group 20.
- **group24**—Diffie-Hellman Group 24.
- **group5**—Diffie-Hellman Group 5.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

pki

Supported Platforms SRX Series, vSRX

Syntax

```
pki {
  auto-re-enrollment {
    certificate-id certificate-id-name {
      ca-profile-name ca-profile-name ;
      challenge-password password ;
      re-enroll-trigger-time-percentage percentage ;
      re-generate-keypair;
    }
  }
  ca-profile ca-profile-name {
    administrator {
      e-mail-address e-mail-address;
    }
    ca-identity ca-identity;
    enrollment {
        retry number;
        retry-interval seconds;
        url url-name;
    }
    revocation-check {
      crl {
        disable {
          on-download-failure;
        }
        refresh-interval hours;
        url url-name;
      }
      disable;
      ocsp {
        connection-failure (disable | fallback-crl);
        disable-responder-revocation-check;
        nonce-payload (enable | disable);
        url ocsp-url;
      }
      use-ocsp;
    }
    routing-instance routing-instance-name;
    source-address ip-address;
  }
  traceoptions {
    file {
      filename;
      files number;
      match regular-expression;
      size maximum-file-size;
      (world-readable | no-world-readable);
    }
    flag flag;
    no-remote-trace;
  }
```

}

Hierarchy Level [edit security]

Release Information Statement modified in Junos OS Release 8.5.

Description Configure an IPsec profile to request digital certificates.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Understanding Certificates and PKI on page 353

pki-local-certificate

Supported Platforms SRX Series, vSRX

Syntax pki-local-certificate *name*;

Hierarchy Level [edit system services web-management https]

Release Information Statement introduced in Junos OS Release 9.1.

Description Specify the name of the certificate that is generated by public key infrastructure (PKI) and authenticated by certificate authority (CA).

Options *name* —Name of certificate.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- Understanding Certificates and PKI on page 353

policy (Security Group VPN IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
policy policy-name {
  description description;
  mode (aggressive | main);
  pre-shared-key (ascii-text key | hexadecimal key);
  proposals [proposal-name];
}
```

Hierarchy Level

[edit security group-vpn member ike]
[edit security group-vpn server ike]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure an IKE policy. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options ***policy-name***—Name of the IKE policy. The policy name can be up to 32 alphanumeric characters long.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

policy (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
policy policy-name {
  certificate {
    local-certificate certificate-id;
    peer-certificate-type (pkcs7 | x509-signature);
    policy-oids [ oid ];
  }
  description description;
  mode (aggressive | main);
  pre-shared-key (ascii-text key | hexadecimal key);
  proposal-set (basic | compatible | standard |suiteb-gcm-128 | suiteb-gcm-256);
  proposals [proposal-name];
  reauth-frequency number;
}
```

Hierarchy Level [edit security ike]

Release Information Statement modified in Junos OS Release 8.5. Support for **suiteb-gcm-128** and **suiteb-gcm-256** options added in Junos OS Release 12.1X45-D10. Support for **policy-oids** option added in Junos OS Release 12.3X48-D10.

Description Configure an IKE policy.

Options *policy-name*—Name of the IKE policy. The policy name can be up to 32 alphanumeric characters long.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

policy (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
policy policy-name {
  description description;
  perfect-forward-secrecy keys (group1 | group14 | group19 | group2 | group20 | group24 |
    group5);
  proposal-set (basic | compatible | standard | suiteb-gcm-128 | suiteb-gcm-256);
  proposals [proposal-name];
}
```

Hierarchy Level [edit security ipsec]

Release Information Statement modified in Junos OS Release 8.5. Support for group 14 is added in Junos OS Release 11.1.

Description Define an IPsec policy.

Options ***policy-name*** —Name of the IPsec policy.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

policy-oids

Supported Platforms SRX Series, vSRX

Syntax policy-oids [*oid*];

Hierarchy Level [edit security ike policy *policy-name* certificate]

Release Information Statement introduced in Junos OS Release 12.3X48-D10.

Description Configure policy object identifiers (OIDs). This configuration is optional.

Options *oid*—Policy OID contained in a peer's certificate or certificate chain. Up to five policy OIDs can be configured. Each OID can be up to 63 bytes long.

> NOTE: You must ensure that at least one of the configured policy OIDs 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.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation
- Understanding Digital Certificate Validation on page 363

pre-shared-key (Security IKE Policy)

Supported Platforms SRX Series, vSRX

Syntax pre-shared-key (ascii-text *key* | hexadecimal *key*);

Hierarchy Level [edit security ike policy *policy-name*]

Release Information Statement modified in Junos OS Release 8.5.

Description Define a preshared key for an IKE policy.

NOTE: The device deletes existing IKE and IPsec SAs when you update the pre-shared-key configuration in the IKE policy.

Options **ascii-text** ***key***—Specify a string of 1 to 255 ASCII text characters for the key. Characters **@ + -** or **=** are not allowed. To include the special characters **( ) [ ] { } , ;** enclose either the entire key string or the special character in quotation marks; for example **"str)ng"** or **str")"ng**. Other use of quotation marks within the string is not allowed. With **des-cbc** encryption, the key contains 8 ASCII characters. With **3des-cbc** encryption, the key contains 24 ASCII characters.

hexadecimal ***key***—Specify a string of 1 to 255 hexadecimal characters for the key. Characters must be hexadecimal digits **0** through **9**, or letters **a** through **f** or **A** through **F**. With **des-cbc** encryption, the key contains 16 hexadecimal characters. With **3des-cbc** encryption, the key contains 48 hexadecimal characters.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

probe-idle-tunnel

Supported Platforms SRX Series, vSRX

Syntax probe-idle-tunnel;

Hierarchy Level [edit security ike gateway *gateway-name* dead-peer-detection]

Release Information Statement introduced in Junos OS Release 12.1X46-D10.

Description Send dead peer detection (DPD) messages during idle traffic time between peers.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

profile (Access)

Supported Platforms SRX Series, vSRX

Syntax

```
profile profile-name {
  accounting {
    accounting-stop-on-access-deny;
    accounting-stop-on-failure;
    coa-immediate-update;
    duplication;
    immediate-update;
    order [accounting-method];
    statistics (time | volume-time);
    update-interval minutes;
  }
  accounting-order [accounting-method];
  address-assignment pool pool-name;
  authentication-order [ldap | none | password | securid];
  authorization-order [jsrc];
  client client-name {
    chap-secret chap-secret;
    client-group [ group-names ];
    firewall-user {
      password password;
    }
    no-rfc2486;
    pap-password pap-password;
    x-auth ip-address;
  }
  client-name-filter {
    count number;
    domain-name domain-name;
    separator special-character;
  }
  ldap-options {
    assemble {
      common-name common-name;
    }
    base-distinguished-name base-distinguished-name;
    revert-interval seconds;
    search {
      admin-search {
        distinguished-name distinguished-name;
        password password;
      }
      search-filter search-filter-name;
    }
  }
  ldap-server server-address {
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    source-address source-address;
    timeout seconds;
```



```
  }
  provisioning-order (gx-plus | jsrc);
  service {
    accounting-order {
      activation-protocol;
      radius;
    }
  }
  session-options {
    client-group [group-name];
    client-idle-timeout minutes;
    client-session-timeout minutes;
  }
}
```

Hierarchy Level [edit access]

Release Information Statement introduced in Junos OS Release 10.4.

Description Create a profile containing a set of attributes that define device management access.

Required Privilege Level access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

Related Documentation

- *Understanding Interfaces*
- *Understanding User Authentication for Security Devices*
- *Ethernet Switching and Layer 2 Transparent Mode Overview*

profile (TCP Encapsulation)

Supported Platforms SRX Series, vSRX

Syntax

```
profile profile-name;
   log ;
}
```

Hierarchy Level [edit security tcp-encap]

Release Information Statement introduced in Junos OS Release 15.1X49-D80. This statement is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices.

Description Configure a TCP encapsulation profile for a remote access client to a remote access gateway on an SRX Series device.

Options **profile *profile-name***—Name for the TCP encapsulation profile.

log—Enable logging for remote access client connections.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding SSL Remote Access VPNs with NCP Exclusive Remote Access Client on page 845

proposal (Security Group VPN Member IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
proposal proposal-name {
  authentication-algorithm (sha-256 | sha-384);
  authentication-method pre-shared-keys;
  description description;
  dh-group (group14 | group24);
  encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
  lifetime-seconds seconds;
}
```

Hierarchy Level [edit security group-vpn member ike]

Release Information Statement introduced in Junos OS Release 10.2.

Description Define an IKE proposal. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options ***proposal-name***—Name of the IKE proposal. The proposal name can be up to 32 alphanumeric characters long.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

proposal (Security Group VPN Server IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
proposal proposal-name {
  authentication-algorithm (sha-256 | sha-384);
  authentication-method pre-shared-keys;
  description description;
  dh-group (group14 | group24);
  encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
}
```

Hierarchy Level [edit security group-vpn server ike]

Release Information Statement introduced in Junos OS Release 10.2.

Description Define an IKE proposal for group VPN server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options *proposal-name*—Name of the IKE proposal. The proposal name can be up to 32 alphanumeric characters long.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

proposal (Security Group VPN Server IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
proposal proposal-name {
  authentication-algorithm hmac-sha-256-128;
  description description;
  encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
  lifetime-seconds seconds;
}
```

Hierarchy Level [edit security group-vpn server ipsec]

Release Information Statement introduced in Junos OS Release 10.2.

Description Define an IPsec proposal. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options *proposal-name*—Name of the IPsec proposal.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

proposal (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
proposal proposal-name {
  authentication-algorithm (md5 | sha-256 | sha-384| sha1);
  authentication-method (dsa-signatures | ecdsa-signatures-256 | ecdsa-signatures-384
    | pre-shared-keys | rsa-signatures);
  description description;
  dh-group (group1 | group14 | group19 | group2 | group20 | group24 | group5);
  encryption-algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
  lifetime-seconds seconds;
}
```

Hierarchy Level [edit security ike]

Release Information Statement modified in Junos OS Release 8.5. Support for **dh-group group 14** and **dsa-signatures** added in Junos OS Release 11.1. Support for **sha-384**, e**cdsa-signatures-256**, **ecdsa-signatures-384**, **group19**, **group20**, and **group24** options added in Junos OS Release 12.1X45-D10.

Description Define an IKE proposal.

Options ***proposal-name***—Name of the IKE proposal. The proposal name can be up to 32 alphanumeric characters long.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

proposal (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
proposal proposal-name {
  authentication-algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
  description description;
  encryption-algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-192-gcm
    | aes-256-cbc | aes-256-gcm | des-cbc);
  lifetime-kilobytes kilobytes;
  lifetime-seconds seconds;
  protocol (ah | esp);
}
```

Hierarchy Level [edit security ipsec]

Release Information Statement modified in Junos OS Release 8.5.

Description Define an IPsec proposal.

Options *proposal-name*—Name of the IPsec proposal.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

proposals (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax proposals [*proposal-name*];

Hierarchy Level [edit security group-vpn member ike policy *policy-name*]
[edit security group-vpn server ike policy *policy-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for **group-vpn** hierarchies added in Junos OS Release 10.2.

Description Specify up to four Phase 1 proposals for an IKE policy. If you include multiple proposals, use the same Diffie-Hellman group in all of the proposals. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options ***proposal-name***—Names of up to four configured Phase 1 proposals.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Group VPNv2 Overview on page 665

proposals (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax proposals [*proposal-name*];

Hierarchy Level [edit security ike policy *policy-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for **group-vpn** hierarchies added in Junos OS Release 10.2.

Description Specify up to four Phase 1 proposals for an IKE policy. If you include multiple proposals, use the same Diffie-Hellman group in all of the proposals.

Options *proposal-name*—Names of up to four configured Phase 1 proposals.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

proposals (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax proposals [*proposal-name*];

Hierarchy Level [edit security ipsec policy *policy-name*]

Release Information Statement modified in Junos OS Release 8.5.

Description Specify one or more proposals for an IPsec policy.

Options *proposal-name*—Name of a configured proposal.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

proposal-set (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax proposal-set (basic | compatible | prime-128 | prime-256 | standard | suiteb-gcm-128 | suiteb-gcm-256);

Hierarchy Level [edit security ike policy *policy-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **suiteb-gcm-128** and **suiteb-gcm-256** options added in Junos OS Release 12.1X45-D10. Support for **prime-128** and **prime-256** options added in Junos OS Release 15.1X49-D40.

Description Specify a set of default Internet Key Exchange (IKE) proposals.

NOTE: The **prime-128** and **prime-256** proposal sets require IKEv2 and certificate-based authentication.

Options

- **basic**—Includes a basic set of two IKE proposals:
  - Proposal 1—Preshared key, Data Encryption Standard (DES) encryption, and Diffie-Hellman (DH) group 1 and Secure Hash Algorithm 1 (SHA-1) authentication.
  - Proposal 2—Preshared key, DES encryption, and DH group 1 and Message Digest 5 (MD5) authentication.
- **compatible**—Includes a set of four commonly used IKE proposals:
  - Proposal 1—Preshared key, triple DES (3DES) encryption, and Gnutella2 (G2) and SHA-1 authentication.
  - Proposal 2—Preshared key, 3DES encryption, and DH group 2 and MD5 authentication.
  - Proposal 3—Preshared key, DES encryption, and DH group 2 and SHA-1 authentication.
  - Proposal 4—Preshared key, DES encryption, and DH group 2 and MD5 authentication.
- **prime-128**—Provides the following proposal set (this option is not supported on Group VPNv2):
  - Authentication method—Elliptic Curve Digital Signature Algorithm (ECDSA) 256-bit signatures.
  - Diffie-Hellman Group—19.
  - Encryption algorithm—Advanced Encryption Standard (AES) 128-bit Galois/Counter Mode (GCM).
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication).

When this option is used, **prime-128** should also be configured at the [**edit security ipsec policy *policy-name* proposal-set**] hierarchy level.

- **prime-256**—Provides the following proposal set (this option is not supported on Group VPNv2):
  - Authentication method—ECDSA 384-bit signatures.
  - Diffie-Hellman Group—20.
  - Encryption algorithm—AES 256-bit GCM.
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication).

  When this option is used, **prime-256** should also be configured at the [**edit security ipsec policy *policy-name* proposal-set**] hierarchy level.

- **standard**—Includes a standard set of two IKE proposals:
  - Proposal 1— Preshared key, 3DES encryption, and DH group 2 and SHA-1 authentication.
  - Proposal 2—Preshared key, AES 128-bit encryption, and DH group 2 and SHA-1 authentication.
- **suiteb-gcm-128**—Provides the following Suite B proposal set (this option is not supported on Group VPNv2):
  - Authentication method—ECDSA 256-bit signatures
  - Diffie-Hellman Group—19
  - Encryption algorithm—Advanced Encryption Standard (AES) 128-bit cipher block chaining (CBC)

    NOTE: CBC mode is used instead of GCM.

  - Authentication algorithm—SHA-256
- **suiteb-gcm-256**—Provides the following Suite B proposal set (this option is not supported on Group VPNv2):
  - Authentication method—ECDSA 384-bit signatures
  - Diffie-Hellman Group—20
  - Encryption algorithm—AES 256-bit CBC

    NOTE: CBC mode is used instead of GCM.

  - Authentication algorithm—SHA-384

Required Privilege Level
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

proposal-set (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
proposal-set (basic | compatible | prime-128 | prime-256 | standard | suiteb-gcm-128 |
   suiteb-gcm-256);
```

Hierarchy Level [edit security ipsec policy *policy-name*]

Release Information Statement introduced in Junos OS Release 10.4. Support for **suiteb-gcm-128** and **suiteb-gcm-256** options added in Junos OS Release 12.1X45-D10. Support for **prime-128** and **prime-256** options added in Junos OS Release 15.1X49-D40.

Description Define a set of default IPsec proposals.

Options

- **basic** —nopfs-esp-des-sha and nopfs-esp-des-md5
- **compatible**—nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, and nopfs-esp-des-md5
- **prime-128**—Provides the following proposal set:
  - Encapsulating Security Payload (ESP) protocol
  - Encryption algorithm—Advanced Encryption Standard Galois/Counter mode (AES-GCM)128-bit
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication)

  NOTE: This option is not supported on Group VPNv2.

- **prime-256**—Provides the following proposal set:
  - ESP protocol
  - Encryption algorithm—AES-GCM 256-bit
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication)

  NOTE: This option is not supported on Group VPNv2.

- **standard**—g2-esp-3des-sha and g2-esp-aes128-sha
- **suiteb-gcm-128**—Provides the following proposal set:

- ESP protocol
- Encryption algorithm—AES-GCM 128-bit
- Authentication algorithm—None (AES-GCM provides both encryption and authentication)

NOTE: This option is not supported on Group VPNv2.

- **suiteb-gcm-256**—Provides the following proposal set:
  - ESP protocol
  - Encryption algorithm—AES-GCM 256-bit
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication)

NOTE: This option is not supported on Group VPNv2.

NOTE: The Perfect Forward Secrecy setting in IPsec policy overrides the settings in proposal sets.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

protocol (IPsec SA for OSPF)

Supported Platforms SRX Series, vSRX

Syntax protocol (ah | esp);

Hierarchy Level [edit security ipsec security-association *sa-name* mode transport manual direction bidirectional]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Configure the IPsec protocol for a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link.

Options **protocol**—Define the IPsec protocol for the manual SA. The protocol can be one of the following:

- **ah**—Authentication Header (AH) protocol.
- **esp**—Encapsulating Security Payload (ESP) protocol. This is the default.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding OSPF and OSPFv3 Authentication on SRX Series Devices on page 37

protocol (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax protocol (ah | esp);

Hierarchy Level [edit security ipsec proposal ***proposal-name***]

Release Information Statement modified in Junos OS Release 8.5.

Description Define the IPsec protocol for a manual or dynamic security association (SA).

NOTE: **The device deletes existing IPsec SAs when you update the encryption-algorithm configuration in the IPsec proposal.**

Options

- **ah**—Authentication Header protocol.
- **esp**—Encapsulating Security Payload (ESP) protocol.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

protocol (Security IPsec Manual SA)

Supported Platforms SRX Series, vSRX

Syntax protocol (ah | esp)

Hierarchy Level [edit security ipsec vpn *vpn-name* manual]

Release Information Statement modified in Junos OS Release 8.5.

Description Define the IPsec protocol for the manual security association.

Options

- **ah**—Authentication Header protocol.
- **esp**—ESP protocol (To use the ESP protocol, you must also use the **tunnel** statement at the **[edit security ipsec security-association** ***sa-name*** **mode**] hierarchy level.)

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

proxy-identity

Supported Platforms SRX Series, vSRX

Syntax

```
proxy-identity {
  local ip-prefix;
  remote ip-prefix;
  service (all | service-name);
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name* ike]

Release Information Statement introduced in Junos OS Release 8.5. Support for IPv6 added in Junos OS Release 12.1X46-D10.

Description Optionally specify the IPsec proxy ID to use in negotiations. The default is the identity based on the IKE gateway. If the IKE gateway is an IPv6 site-to-site gateway, the default proxy ID is ::/0. If the IKE gateway is an IPv4 gateway or a dynamic endpoint or dialup gateway, the default proxy ID is 0.0.0.0/0.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

reauth-frequency

Supported Platforms SRX Series, vSRX

Syntax reauth-frequency *number*;

Hierarchy Level [edit security ike policy *policy-name*];

Release Information Statement introduced in Junos OS Release 15.1X49-D60.

Description Configure the reauthentication frequency to trigger a new IKEv2 reauthentication. Reauthentication creates a new IKE SA, creates new child SAs within the IKE SA, and then deletes the old IKE SA.

Default This feature is disabled by default.

Options **reauth-frequency** ***number***—Number of IKE rekeys that occurs before reauthentication occurs. If **reauth-frequency** is **1**, reauthentication occurs every time there is an IKE rekey. If **reauth-frequency** is **2**, reauthentication occurs at every other IKE rekey. If **reauth-frequency** is **3**, reauthentication occurs at every third IKE rekey.

Default: 0 (disable)

Range: 0-100

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding IKEv2 Reauthentication on page 152

re-enroll-trigger-time-percentage (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax re-enroll-trigger-time-percentage *percentage*;

Hierarchy Level [edit security pki auto-re-enrollment cmpv2 certificate-id *certificate-id-name*]
[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]

Release Information Statement modified in Junos OS Release 9.0. Support for **[edit security pki auto-re-enrollment cmpv2 certificate-id *certificate-id-name*]** and **[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]** hierarchies added in Junos OS Release 15.1X49-D40.

Description Specify the certificate reenrollment trigger as a percentage of the end-entity (EE) certificate’s lifetime that remains before certificate reenrollment is initiated. For example, if the renewal request is to be sent when the certificate's remaining lifetime is 10 percent, then configure 10 for **re-enroll-trigger-time-percentage** value. The time at which the certificate reenrollment is initiated is based on the certificate expiry date.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- Understanding Certificates and PKI on page 353

re-generate-keypair

Supported Platforms SRX Series, vSRX

Syntax re-generate-keypair;

Hierarchy Level [edit security pki auto-re-enrollment cmpv2 certificate-id *certificate-id-name*]
[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]

Release Information Statement introduced in Junos OS Release 8.5. Support for **[edit security pki auto-re-enrollment cmpv2 certificate-id *certificate-id-name*]** and **[edit security pki auto-re-enrollment scep certificate-id *certificate-id-name*]** hierarchies added in Junos OS Release 15.1X49-D40.

Description Specifies new key pair generation for automatic certificate reenrollment. If this statement is not configured, the current key pair is used. If the key pair does not change, the CA does not issue new certificates. We recommend that a new key pair be generated during reenrollment as it provides better security.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation
- Understanding Certificates and PKI on page 353

refresh-interval

Supported Platforms SRX Series, vSRX

Syntax refresh-interval *hours*;

Hierarchy Level [edit security pki ca-profile *ca-profile-name* revocation-check crl]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the amount of time between certificate revocation list (CRL) updates.

Options *number-of-hours*—Time interval, in hours, between CRL updates.

Range: 0 through 8784

Default: 6

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- crl (Security) on page 963

remote (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax remote *ip-prefix*;

Hierarchy Level [edit security ipsec vpn *vpn-name* ike proxy-identity]

Release Information Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1.

Description Specify the remote IPv4 or IPv6 address and subnet mask for the proxy identity.

Options *ip-prefix* —IPv4 or IPv6 address and subnet mask.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

remote-exceptions

Supported Platforms SRX Series

Syntax remote-exceptions *ip-address/mask*;

Hierarchy Level [edit security dynamic-vpn clients *configuration-name*]

Release Information Statement introduced in Junos OS Release 9.5. This statement is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Description Use this statement to specify exceptions to the remote protected resources list for the specified dynamic VPN configuration. Traffic to the specified IP address will not go through the dynamic VPN tunnel and therefore will not be protected by the firewall's security policies.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

remote-identity

Supported Platforms SRX Series, vSRX

Syntax

```
remote-identity {
  (distinguished-name <container container-string> <wildcard wildcard-string> | hostname
    hostname | inet ip-address | inet6 ipv6-address | user-at-hostname e-mail-address);
}
```

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 11.4.

Description Specify the remote IKE identity to exchange with the destination peer to establish communication. If you do not configure a remote-identity, the device uses the IPv4 or IPv6 address corresponding to the remote endpoint by default.

NOTE: For Network Address Translation Traversal (NAT-T), both remote identity and local identity must be configured.

Options

- **distinguished-name**—Specify identity as the distinguished name (DN) from the certificate. If there is more than one certificate on the device, use the **security ike gateway** ***gateway-name*** **policy** ***policy-name*** **certificate local-certificate** ***certificate-id***.

  Optional container and wildcard strings can be specified:

  - **container** ***container-string***—Specify a string for the container.
  - **wildcard** ***wildcard-string***—Specify a string for the wildcard.
- **hostname** ***hostname***—Specify identity as a fully qualified domain name (FQDN).
- **inet** ***ip-address***—Specify identity as an IPv4 address.
- **inet6** ***ipv6-address***—Specify identity as an IPv6 address.
- **user-at-hostname** ***e-mail-address***—Specify identity as an e-mail address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

remote-identity (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
remote-identity {
   (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
}
```

Hierarchy Level

[edit security group-vpn member ike gateway *gateway-name*]
[edit security group-vpn server ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Specify the remote IKE identity of the destination peer. If you do not configure a remote identity, the device uses, by default, the IPv4 address that corresponds to the destination peer. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

hostname *hostname*—Specify a fully qualified domain name (FQDN).

inet *ip-address*—Specify an IPv4 address.

user-at-hostname *username_FQDN*—Specify a fully qualified username.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

remote-protected-resources

Supported Platforms SRX Series

Syntax remote-protected-resources *ip-address/mask*;

Hierarchy Level [edit security dynamic-vpn clients *configuration-name*]

Release Information Statement introduced in Junos OS Release 9.5. This statement is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Description Use this statement to specify which resources to protect using the dynamic VPN feature. Traffic to the protected resource will go through the specified dynamic VPN tunnel and will therefore be protected by the firewall's security policies.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

replay-attacks

Supported Platforms SRX Series, vSRX

Syntax

```
replay-attacks {
  threshold value;
}
```

Hierarchy Level [edit security alarms potential-violation]

Release Information Statement introduced in Junos OS Release 11.2.

Description Raise a security alarm when the device detects a replay attack. A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.

Default Replay attacks do not raise security alarms.

Options

- **threshold** ***value***—Number of reply attacks up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

Range: Range: 0 through 100,00,00,000.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

respond-bad-spi

Supported Platforms SRX Series, vSRX

Syntax respond-bad-spi <*max-responses*>;

Hierarchy Level [edit security ike]

Release Information Statement introduced in Junos OS Release 8.5.

Description Enable response to invalid IPsec Security Parameter Index (SPI) values. If the security associations (SAs) between two peers of an IPsec VPN become unsynchronized, the device resets the state of a peer so that the two peers are synchronized.

Options ***max-responses***—(Optional) Number of times to respond to invalid SPI values per gateway.

Range: 1 through 30

Default: 5

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

revocation-check (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax

```
revocation-check {
  crl {
    disable {
      on-download-failure;
    }
    refresh-interval hours;
    url url-name;
  }
  disable;
  ocsp {
    connection-failure (disable | fallback-crl);
    disable-responder-revocation-check;
    nonce-payload (enable | disable);
    url ocsp-url;
  }
  use-ocsp;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for **ocsp** and **use-ocsp** options added in Junos OS Release 12.1X46-D20.

Description Specify the method the device uses to verify the revocation status of digital certificates.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

routing-instance (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax routing-instance *routing-instance*;

Hierarchy Level [edit security group-vpn member ike gateway *gateway-name*]
[edit security group-vpn server ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Configure the routing instance that the group VPN server or member uses when communicating with a group member or server. This statement is used when the IKE gateway is not configured in the default routing instance. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options routing-instance *routing-instance*—Specify the name of a routing instance. If this is not specified, the default inet.0 routing instance is used.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

routing-instance (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax routing-instance *routing-instance-name*

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement modified in Junos OS Release 9.0.

Description Specify the routing-instance to be used.

Options

- *routing-instance-name*—Name of the routing instance.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

security-association

Supported Platforms SRX Series, vSRX

Syntax

```
security-association sa-name {
  manual {
    direction bidirectional {
      authentication {
        algorithm (hmac-md5-96 | hmac-sha1-96);
        key {
          ascii-text key;
          hexadecimal key;
        }
      }
      auxiliary-spi auxiliary-spi-value;
      encryption {
        algorithm (3des-cbc | des-cbc | null);
        key {
          ascii-text key;
          hexadecimal key;
        }
      }
      protocol (ah | esp);
      spi spi-value;
    }
  }
  mode transport;
}
```

Hierarchy Level [edit security ipsec]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Configure a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link. IPsec can provide authentication and confidentiality to OSPF or OSPFv3 routing packets.

Options ***sa-name***—Name of the SA.

mode—SA mode. For this feature, the mode must be **transport**.

direction—Direction of the manual SA. For this feature, the direction must be **bidirectional**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding OSPF and OSPFv3 Authentication on SRX Series Devices on page 37

server (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
server {
  group name {
    anti-replay-time-window milliseconds;
    description description;
    group-id number;
    ike-gateway [gateway-name];
    ipsec-sa name {
      match-policy policy-name {
        destination ip-address/netmask;
        destination-port number;
        protocol number;
        source ip-address/netmask;
        source-port number;
      }
      proposal proposal-name;
    }
    member-threshold number;
    server-cluster {
      ike-gateway gateway-name;
      retransmission-period seconds;
      server-role (root-server | sub-server);
    }
    server-member-communication {
      certificate certificate-id;
      communication-type unicast;
      encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
      lifetime-seconds seconds;
      number-of-retransmission number;
      retransmission-period seconds;
      sig-hash-algorithm (sha-256 | sha-384);
    }
  }
  ike {
    gateway gateway-name {
      address ip-address ;
      dead-peer-detection {
        always-send;
        interval seconds;
        threshold number;
      }
      dynamic {
        (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
      }
      ike-policy policy-name;
      local-address ip-address;
      local-identity {
        (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
      }
      remote-identity {
        (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
```



```
            }
            routing-instance routing-instance;
        }
        policy policy-name {
            description text;
            mode (aggressive | main);
            pre-shared-key (ascii-text key | hexadecimal key);
            proposals [proposal-name];
        }
        proposal proposal-name {
            authentication-algorithm (sha-256 | sha-384);
            authentication-method pre-shared-keys;
            description description;
            dh-group (group14 | group24);
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
        }
    }
    ipsec {
        proposal proposal-name {
            authentication-algorithm hmac-sha-256-128;
            description description;
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
            lifetime-seconds seconds;
        }
    }
    traceoptions {
        file {
            filename;
            files number;
            match regular-expression;
            size maximum-file-size;
            (world-readable | no-world-readable);
        }
        flag flag;
        gateway-filter {
            local-address ip-address;
            remote-address ip-address;
        }
        level (all | error | info | notice | verbose | warning);
        no-remote-trace;
    }
}
```

Hierarchy Level [edit security group-vpn]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure group VPN server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances. You configure the following on the group server:

- Phase 1 IKE SA for group members
- Phase 2 IPsec proposal
- Group identifier, group members, server-member communications, and group policies to be downloaded to members
- Group VPN trace options

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

server-address (Security Group VPN Member)

Supported Platforms SRX Series, vSRX

Syntax server-address [*ip-address*];

Hierarchy Level [edit security group-vpn member ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Specify the group server that this member registers through a **groupkey-pull** exchange. Up to four server IP addresses can be configured. The 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. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: We recommend that group members only register with sub-servers in a server cluster and not the root-server.

Options *ip-address*—IPv4 address of an IKE gateway.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

server-cluster (Security Group VPN Server)

Supported Platforms SRX Series, vSRX

Syntax

```
server-cluster {
  ike-gateway gateway-name;
  retransmission-period seconds;
  server-role (root-server | sub-server);
}
```

Hierarchy Level [edit security group-vpn server group *name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Configure the Group Domain of Interpretation (GDOI) group controller/key server (GCKS) cluster for the specified group. All servers in a group VPN server cluster must be SRX Series devices. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

ike-gateway *gateway-name*—(Required) Specify the name of the IKE gateway for the local device in the group server cluster. IKE gateways are configured at the [**edit security group-vpn server ike**] hierarchy level.

If the local device is a root-server, the IKE gateway name must be a sub-server in the cluster; up to four sub-server IKE gateways can be specified.

If the local device is a sub-server, the IKE gateway name must be the root-server.

retransmission-period *seconds*—(Optional) Specify the time after which the root-server retransmits a **cluster-update** message if it has not received an acknowledgement from a sub-server.

Range: 2 to 60 seconds.

Default: 10 seconds.

server-role—(Required) Assign the role of the local device in the group server cluster, either **root-server** or **sub-server**. Only one device in the cluster can be configured as the root-server. You can configure up to four other devices as a sub-server in a group server cluster.

NOTE: You must ensure that there is only one root-server at any time for a group VPN server cluster.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Server Clusters on page 714

server-member-communication (Security Group VPN Server)

Supported Platforms SRX Series, vSRX

Syntax

```
server-member-communication {
  certificate certificate-id;
  communication-type unicast;
  encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
  lifetime-seconds seconds;
  number-of-retransmission number;
  retransmission-period seconds;
  sig-hash-algorithm (sha-256 | sha-384);
}
```

Hierarchy Level [edit security group-vpn server group *name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Enable and configure server to member communication. When these options are configured, group members receive new keys before current keys expire. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- **certificate** ***certificate-id***—Specify the certificate identification. Only RSA keys are supported.
- **communication-type**—Configure **unicast** (the default).
- **encryption-algorithm**—Encryption used for communications between the group server and group member. Specify **aes-128-cbc**, **aes-192-cbc**, or **aes-256-cbc**.
- **lifetime-seconds** ***seconds***—Lifetime, in seconds, of the key encryption key (KEK). Specify a value from 180 to 86,400. The default is 3600 seconds.
- **number-of-retransmission** ***number***—For unicast communications, the number of times the group server retransmits messages to a group member when there is no reply. Specify a value from 0 to 60. The default is 2.
- **retransmission-period** ***seconds***—The time period between a transmission and the first retransmission when there is no reply from the group member. Specify a value from 2 to 60. The default is 10 seconds.
- **sig-hash-algorithm**—Authentication algorithm used to authenticate the group member to the group server. Specify **sha-256** or **sha-384**.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

service (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax service (all | *service-name*);

Hierarchy Level [edit security ipsec vpn *vpn-name* ike proxy-identity]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the service (port and protocol combination) to protect.

Options ***service-name*** —Name of the service, as defined with **system-services** (**Interface Host-Inbound Traffic**) and **system-services** (**Zone Host-Inbound Traffic**).

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

session-affinity

Supported Platforms SRX Series, vSRX

Syntax session-affinity ipsec

Hierarchy Level [edit security flow load-distribution]

Release Information Statement introduced in Junos OS Release 11.4R5.
Starting with Junos OS Release 15.1X49-D10, IPsec session affinity is supported for IPsec tunnel-based traffic by the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) for SRX5400, SRX5600, and SRX5800 devices through improved flow module and session cache.

Description Enable VPN session affinity. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices and vSRX instances.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

source-address (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax source-address *ip-address*;

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D60.

Description Specify a source IP address to be used instead of the IP address of the egress interface for communications with external servers. External servers are used for certificate enrollment and reenrollment using Simple Certificate Enrollment Protocol (SCEP) or Certificate Management Protocol version 2 (CMPv2), downloading certificate revocation lists (CRLs) using HTTP or LDAP, or checking certificate revocation status with Online Certificate Status Protocol (OCSP).

Default If this option is not specified, the IP address of the egress interface is used as the source address.

Options **source-address *ip-address***—IPv4 address used to communicate with external servers.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

source-interface (Security)

Supported Platforms SRX Series, vSRX

Syntax source-interface *interface-name* ;

Hierarchy Level [edit security ipsec vpn *vpn-name* vpn-monitor]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the source interface for ICMP requests (VPN monitoring “hellos”). If no source interface is specified, the device automatically uses the local tunnel endpoint interface.

Options *interface-name* —Name of the interface for the ICMP requests.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

spi (IPsec SA for OSPF)

Supported Platforms SRX Series, vSRX

Syntax spi *spi-value*;

Hierarchy Level [edit security ipsec security-association *sa-name* mode transport manual direction bidirectional]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Configure a security parameter index (SPI) for a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link.

Options **spi**—SPI for the manual SA. The SPI uniquely identifies the SA to use at the receiving host (the destination address in the packet).

Range: 256 through 16,639

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation
- Understanding OSPF and OSPFv3 Authentication on SRX Series Devices on page 37

spi (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax spi *spi-value*;

Hierarchy Level [edit security ipsec vpn *vpn-name* manual]

Release Information Statement modified in Junos OS Release 8.5.

Description Configure a security parameter index (SPI) for a security association (SA).

Options ***spi-value*** —An arbitrary value that uniquely identifies which security association (SA) to use at the receiving host (the destination address in the packet).

Range: 256 through 16,639

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

tcp-encap

Supported Platforms SRX Series, vSRX

Syntax

```
tcp-encap {
  global-options {
    enable-tunnel-tracking;
  }
  profile profile-name;
    log ;
  }
  traceoptions {
    file {
      filename;
      files number;
      match regular-expression;
      size maximum-file-size;
      (world-readable | no-world-readable);
    }
    flag (all | configuration | session | tunnel);
    level (all | error | info | notice | verbose | warning);
    no-remote-trace'
  }
```

Hierarchy Level [edit security]

Release Information Statement introduced in Junos OS Release 15.1X49-D80. This statement is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices.

Description Specify TCP encapsulation operations for a remote access client to a remote access gateway on an SRX Series device.

Options **global-options**—Specify global settings for TCP encapsulation.

Values: **enable-tunnel-tracking**—Maintain additional information about IPsec tunnels related to the TCP encapsulation session. The additional information can be used for debugging.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation
- Understanding SSL Remote Access VPNs with NCP Exclusive Remote Access Client on page 845

tcp-encap-profile

Supported Platforms SRX Series, vSRX

Syntax tcp-encap-profile *profile-name*;

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D80. This statement is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices.

Description Specify the TCP encapsulation profile to be used for TCP connections for remote access clients. The profile is configured with the **tcp-encap** configuration statement at the [**edit security**] hierarchy level.

Options ***profile-name***—Name of the TCP encapsulation profile.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding SSL Remote Access VPNs with NCP Exclusive Remote Access Client on page 845
- **tcp-encap on page 1098**

threshold (Security IKE Gateway)

Supported Platforms SRX Series, vSRX

Syntax threshold *number*;

Hierarchy Level [edit security ike gateway *gateway-name* dead-peer-detection]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the maximum number of unsuccessful dead peer detection (DPD) requests to be sent before the peer is considered unavailable.

Options ***number*** —Maximum number of unsuccessful DPD requests to be sent.

Range: 1 through 5

Output: 5

NOTE: The threshold number for the IKEv2 protocol is predefined as 5.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

traceoptions (Security Dynamic VPN)

Supported Platforms SRX Series

Syntax

```
traceoptions {
  file filename;
  flag {
    all <detail | extensive | terse>;
  }
}
```

Hierarchy Level [edit security dynamic-vpn]

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Configure dynamic VPN tracing options. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Options

- **file**—Configure the trace file options.

  file *filename*—Name of the file to receive the output of the tracing operation.

- **flag**—Trace operation to perform. To specify more than one trace operation, include multiple **flag** statements.
  - **all**—Enable all tracing operations
    - **detail**—Display moderate amount of data in trace.
    - **extensive**—Display extensive amount of data in trace.
    - **terse**—Display minimum amount of data in trace.

Required Privilege Level trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

traceoptions (Security Group VPN)

Supported Platforms SRX Series, vSRX

Syntax

```
traceoptions {
  file {
    filename;
    files number;
    match regular-expression;
    size maximum-file-size;
    (world-readable | no-world-readable);
  }
  flag flag;
  gateway-filter {
    local-address ip-address;
    remote-address ip-address;
  }
  level (all | error | info | notice | verbose | warning);
  no-remote-trace;
}
```

Hierarchy Level [edit security group-vpn member ike]
[edit security group-vpn server]

Release Information Statement introduced in Junos OS Release 10.2. Support for **gateway-filter** options and for the [**edit security group-vpn member ike**] hierarchy level added in Junos OS Release 15.1X49-D30 for vSRX.

Description Configure group VPN trace options. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- **file**—Configure the trace file options.
  - ***filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.
  - **files *number***—Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed to ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

    If you specify a maximum number of files, you also must specify a maximum file size with the **size** option and a filename.

    Range: 2 through 1000 files

    Default: 10 files
  - **match *regular-expression***—Refine the output to include lines that contain the regular expression.

- **size** ***maximum-file-size***—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the **trace-file** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

  If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option and filename.

  Syntax: ***x*** **k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

  Range: 10 KB through 1 GB

  Default: 128 KB

- **world-readable | no-world-readable**—By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.

- **flag**—Trace operation to perform. To specify more than one trace operation, include multiple **flag** statements.
  - **all**—Trace all activity.
  - **certificates**—Trace certificate-related activity.
  - **config**—Trace configuration activity.
  - **database**—Trace SA-related database activity.
  - **general**—Trace general activity.
  - **high-availability**—Trace high-availability operations.
  - **ike**—Trace IKE protocol activity.
  - **next-hop-tunnels**—Trace next-hop tunnel operations.
  - **parse**—Trace configuration processing.
  - **policy-manager**—Trace IKE callback activity.
  - **routing-socket**—Trace routing socket activity.
  - **thread**—Trace thread processing.
  - **timer**—Trace timer activity.

- **gateway-filter**—Configure debugging for the tunnel between the group VPN server and a group member. This option is configured on a group VPN server or member.
  - **local-address**—When configured on a server, the IP address of the group VPN server. When configured on a member, the IP address of the group VPN member.
  - **remote-address**—When configured on a server, the IP address of the group VPN member. When configured on a member, the IP address of the group VPN server.

- **level**—Set the level of debugging.
  - **all**—Match all levels.
  - **error**—Match error conditions.
  - **info**—Match informational messages.
  - **notice**—Match conditions that should be handled specifically.
  - **verbose**—Match verbose messages.
  - **warning**—Match warning messages.
- **no-remote-trace**—Disable remote tracing.

Required Privilege Level

trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

Related Documentation

- Group VPNv2 Overview on page 665

traceoptions (Security IKE)

Supported Platforms SRX Series, vSRX

Syntax

```
traceoptions {
  file {
    filename;
    files number;
    match regular-expression;
    size maximum-file-size;
    (world-readable | no-world-readable);
  }
  flag flag;
  no-remote-trace;
  rate-limit messages-per-second;
}
```

Hierarchy Level [edit security ike]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure IKE tracing options.

Options

- file—Configure the trace file options.
  - ***filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.
  - **files *number***—Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed to ***trace-file*.0,** then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

    If you specify a maximum number of files, you also must specify a maximum file size with the **size** option and a filename.

    Range: 2 through 1000 files

    Default: 10 files
  - **match *regular-expression***—Refine the output to include lines that contain the regular expression.
  - **size *maximum-file-size***—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the **trace-file** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

    If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option and filename.

Syntax: ***x*** **k** to specify KB, ***x*** **m** to specify MB, or ***x*** **g** to specify GB

Range: 10 KB through 1 GB

Default: 128 KB

- **world-readable | no-world-readable**—By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.

- **flag**—Trace operation to perform. To specify more than one trace operation, include multiple **flag** statements.
  - **all**—Trace all iked process modules activity
  - **certificates**—Trace certificate-related activity
  - **config**—Trace configuration download processing
  - **database**—Trace VPN-related database activity
  - **general**—Trace general activity
  - **high-availability**—Trace high-availability operations
  - **ike**—Trace IKE protocol activity
  - **next-hop-tunnels**—Trace next-hop tunnels operations
  - **parse**—Trace VPN parsing activity
  - **policy-manager**—Trace iked callback activity
  - **routing-socket**—Trace routing socket activity
  - **thread**—Trace thread processing
  - **timer**—Trace timer activity
- **no-remote-trace**—Set remote tracing as disabled.
- **rate-limit** ***messages-per-second***—Configure the incoming rate of trace messages.

  Range: 0 through 4,294,967,295

Required Privilege Level

trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

traceoptions (Security IPsec)

Supported Platforms SRX Series, vSRX

Syntax

```
traceoptions {
  flag flag;
}
```

Hierarchy Level [edit security ipsec]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure IPsec tracing options.

NOTE: Configure IPsec tracing options only when instructed to do so by your Juniper support representative.

Trace operations are written to the trace file **/var/log/kmd**.

Options

- **flag**—To specify more than one trace operation, include multiple **flag** statements.
  - **all**—Trace with all flags enabled
  - **next-hop-tunnel-binding**—Trace next-hop tunnel binding events
  - **packet-drops**—Trace packet drop activity
  - **packet-processing**—Trace data packet processing events
  - **security-associations**—Trace security association (SA) management events

Required Privilege Level trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

traceoptions (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax

```
traceoptions {
  file {
    filename;
    files number;
    match regular-expression;
    size maximum-file-size;
    (world-readable | no-world-readable);
  }
  flag flag;
  no-remote-trace;
}
```

Hierarchy Level [edit security pki]

Release Information Statement modified in Junos OS Release 8.5.

Description Configure public key infrastructure (PKI) tracing options.

Options

- **file**—Configure the trace file options.
  - ***filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. By default, the name of the file is the name of the process being traced.
  - **files *number***—Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed to ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

    If you specify a maximum number of files, you also must specify a maximum file size with the **size** option and a filename.

    Range: 2 through 1000 files

    Default: 10 files
  - **match *regular-expression***—Refine the output to include lines that contain the regular expression.
  - **size *maximum-file-size***—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the **trace-file** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

    If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option and a filename.

Syntax: ***x*** **K** to specify KB, ***x*** **m** to specify MB, or ***x*** **g** to specify GB

Range: 10 KB through 1 GB

Default: 128 KB

- **world-readable** | **no-world-readable**—By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.

- **flag**—Trace operation to perform. To specify more than one trace operation, include multiple **flag** statements.
  - **all**—Trace with all flags enabled
  - **certificate-verification**—Trace PKI certificate verification events
  - **online-crl-check**—Trace PKI online certificate revocation list (CRL) events
- **no-remote-trace**—Set remote tracing as disabled.

Required Privilege Level

trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

traceoptions (TCP Encapsulation)

Supported Platforms SRX Series, vSRX

Syntax

```
traceoptions {
  file {
    filename;
    files number;
    match regular-expression;
    size maximum-file-size;
    (world-readable | no-world-readable);
  }
  flag (all | configuration | session | tunnel);
  level (all | error | info | notice | verbose | warning);
  no-remote-trace;
}
```

Hierarchy Level [edit security tcp-encap]

Release Information Statement introduced in Junos OS Release 15.1X49-D80. This statement is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices.

Description Configure TCP encapsulation tracing options.

Options **file**—Configure the trace file options.

- ***filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.
- **files *number***—Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed to ***trace-file*.0,** then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

  If you specify a maximum number of files, you also must specify a maximum file size with the **size** option and a filename.

  Range: 2 through 1000 files

  Default: 10 files
- **match *regular-expression***—Refine the output to include lines that contain the regular expression.
- **size *maximum-file-size***—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**. When ***trace-file*.0** reaches its maximum size, it is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

  If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option and filename.

Syntax: ***x*** **k** to specify KB, ***x*** **m** to specify MB, or ***x*** **g** to specify GB

Range: 10 KB through 1 GB

Default: 128 KB

- **world-readable | no-world-readable**—By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.

flag—Trace operation to perform. To specify more than one trace operation, include multiple flag statements.

- **all**—Trace all activity.
- **configuration**—Trace configuration events.
- **session**—Trace session related events.
- **tunnel**—Trace tunnel events.

level—Set the level of debugging.

- **all**—Match all levels.
- **error**—Match error conditions.
- **info**—Match informational messages.
- **notice**—Match conditions that should be handled specifically.
- **verbose**—Match verbose messages.
- **warning**—Match warning messages.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding SSL Remote Access VPNs with NCP Exclusive Remote Access Client on page 845
- **tcp-encap on page 1098**

traffic-selector

Supported Platforms SRX Series, vSRX

Syntax

```
traffic-selector traffic-selector-name {
  local-ip ip-address/netmask;
  remote-ip ip-address/netmask;
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name*]

Release Information Statement introduced in Junos OS Release 12.1X46-D10.

Description Configure local and remote IP addresses for a traffic selector.

Options

local-ip *ip-address/netmask*—A local IP address or a local subnetwork protected by the local VPN device.

remote-ip *ip-address/netmask*—A remote IP address or a remote subnetwork protected by the peer VPN device.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

trusted-ca (Security IKE Policy)

Supported Platforms SRX Series, vSRX

Syntax trusted-ca (*ca-index* | use-all);

Hierarchy Level [edit security ike policy *policy-name* certificate]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the preferred certificate authority (CA) to use when requesting a certificate from the peer. If no value is specified, then no certificate request is sent (although incoming certificates are still accepted).

Options

- *ca-index*—Preferred certificate authority ID for the device to use.
- **use-all**—Device uses all configured certificate authorities.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

use-ocsp (Security PKI)

Supported Platforms SRX Series, vSRX

Syntax use-ocsp;

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement introduced in Junos OS Release 12.1X46-D20.

Description Specify the Online Certificate Status Protocol (OCSP) as the method to check the revocation status of a certificate. CRL is the default method.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Certificates and PKI on page 353

user (Security Dynamic VPN)

Supported Platforms SRX Series

Syntax user *username*;

Hierarchy Level [edit security dynamic-vpn client *configuration-name*]

Release Information Statement introduced in Junos OS Release 9.5.

Description Specify which users can access the selected dynamic VPN configuration. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

user-at-hostname

Supported Platforms SRX Series, vSRX

Syntax user-at-hostname *e-mail-address*;

Hierarchy Level [edit security ike gateway *gateway-name* dynamic]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure an e-mail address.

Options ***e-mail-address*** —Valid e-mail address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

user-groups (Security Dynamic VPN)

Supported Platforms SRX Series

Syntax user-groups *user-group-name*;

Hierarchy Level [edit security dynamic-vpn client *configuration-name*]

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Specify which users can access the selected dynamic VPN configuration. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

verify-path

Supported Platforms SRX Series, vSRX

Syntax

```
verify-path {
  destination-ip ip-address;
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name* vpn-monitor]

Release Information Statement introduced in Junos OS Release 15.1X49-D70.

Description Verify the IPsec datapath before the secure tunnel (st0) interface is activated and route(s) associated with the interface are installed in the Junos OS forwarding table. This configuration is useful in network topologies where there is a transit firewall located between the VPN tunnel endpoints, and IPsec data traffic that uses active routes for an established VPN tunnel on the st0 interface might be blocked by the transit firewall.

When this option is configured, the source interface and destination IP addresses that can be configured for VPN monitor operation are not used for IPsec datapath verification. The source for the ICMP requests in the IPsec datapath verification is the local tunnel endpoint.

Options **destination-ip *ip-address***—Original, untranslated IP address of the peer tunnel endpoint that is behind a NAT device. This IP address must not be the NAT translated IP address. This option is required if the peer tunnel endpoint is behind a NAT device. The verify-path ICMP request is sent to this IP address so that the peer can generate an ICMP response.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding IPsec DataPath Verification on page 906

version (Security IKE Gateway)

Supported Platforms SRX Series, vSRX

Syntax version (v1-only | v2-only);

Hierarchy Level [edit security ike gateway *gateway-name*]

Release Information Statement introduced in Junos OS Release 11.3.

Description Specify the IKE version to use to initiate the connection.

Options **v1-only**—The connection must be initiated using IKE version 1. This is the default.

v2-only—The connection must be initiated using IKE version 2.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

vpn (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
vpn vpn-name {
  bind-interface interface-name;
  copy-outer-dscp;
  establish-tunnels (immediately | on-traffic);
  ike {
    gateway gateway-name;
    idle-time seconds;
    install-interval seconds;
    ipsec-policy ipsec-policy-name;
    no-anti-replay;
    proxy-identity {
      local ip-prefix;
      remote ip-prefix;
      service (any | service-name);
    }
  }
  manual {
    authentication {
      algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
      key (ascii-text key | hexadecimal key);
    }
    encryption {
      algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
      key (ascii-text key | hexadecimal key);
    }
    external-interface external-interface-name;
    gateway ip-address;
    protocol (ah | esp);
    spi spi-value;
  }
  traffic-selector traffic-selector-name {
    local-ip ip-address/netmask;
    remote-ip ip-address/netmask;
  }
  vpn-monitor {
    destination-ip ip-address;
    optimized;
    source-interface interface-name;
    verify-path {
      destination-ip ip-address;
    }
  }
}
```

Hierarchy Level [edit security ipsec]

Release Information Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1. Support for **copy-outer-dscp** added in Junos OS Release 15.1X49-D30.

Description Configure an IPsec VPN.

Options *vpn-name* —Name of the VPN.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

vpn-monitor

Supported Platforms SRX Series, vSRX

Syntax

```
vpn-monitor {
  destination-ip ip-address;
  optimized;
  source-interface interface-name;
  verify-path {
    destination-ip ip-address;
  }
}
```

Hierarchy Level [edit security ipsec vpn *vpn-name*]

Release Information Statement introduced in Junos OS Release 8.5. **verify-path** keyword and option added in Junos OS Release 15.1X49-D70.

Description Configure settings for VPN monitoring.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- IPsec VPN Overview on page 3

vpn-monitor-options

Supported Platforms SRX Series, vSRX

Syntax

```
vpn-monitor-options {
  interval seconds;
  threshold number;
}
```

Hierarchy Level [edit security ipsec]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure VPN monitoring options.

Options

- **interval** ***seconds*** —Interval at which to send ICMP requests to the peer.

  Range: 2 through 3600 seconds
  Default: 10 seconds

- **threshold** ***number*** —Number of consecutive unsuccessful pings before the peer is declared unreachable.

  Range: 1 through 65,536 pings
  Default: 10 pings

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

wildcard

Supported Platforms SRX Series, vSRX

Syntax wildcard *string*;

Hierarchy Level [edit security ike gateway *gateway-name* dynamic distinguished-name]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify that the values of a dynamic virtual private network (VPN) endpoint user's distinguished name's identity fields match the values in the group IKE user's distinguished name's fields. The order of the identity fields in the distinguished name strings does not matter during a match.

Options ***string***—Distinguished name identity values to be matched.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- IPsec VPN Overview on page 3

xauth-attributes

Supported Platforms SRX Series, vSRX

Syntax

```
xauth-attributes {
  primary-dns IP address;
  primary-wins IP address;
  secondary-dns IP address;
  secondary-wins IP address;
}
```

Hierarchy Level [edit access address-assignment pool <name> family (inet | inet6)]

Release Information Statement introduced in Junos OS Release 10.4.

Description Configure XAuth attributes.

Options

- **apply-groups**—Groups from which to inherit configuration data.
- **apply-groups-except**—Do not inherit configuration data from these groups.
- **primary-dns**—Specify the primary-dns IP address.
- **secondary-dns**—Specify the secondary-dns IP address.
- **primary-wins**—Specify the primary-wins IP address.
- **secondary-wins**—Specify the secondary-wins IP address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Dynamic VPN Overview on page 849

CHAPTER 33

Operational Commands

- clear security dynamic-vpn all
- clear security dynamic-vpn user
- clear security group-vpn member group
- clear security group-vpn member ike security-associations
- clear security group-vpn member ipsec security-associations
- clear security group-vpn member ipsec security-associations statistics
- clear security group-vpn member ipsec statistics
- clear security group-vpn server
- clear security group-vpn server server-cluster statistics
- clear security group-vpn server statistics
- clear security ike respond-bad-spi-count
- clear security ike security-associations
- clear security ipsec security-associations
- clear security ipsec statistics
- clear security ipsec tunnel-events-statistics
- clear security pki key-pair (Local Certificate)
- clear security pki local-certificate (Device)
- clear security tcp-encap statistics
- request security pki ca-certificate ca-profile-group load
- request security pki ca-certificate enroll (Security)
- request security pki ca-certificate load (Security)
- request security pki ca-certificate verify (Security)
- request security pki crl load (Security)
- request security pki generate-certificate-request (Security)
- request security pki generate-key-pair (Security)
- request security pki key-pair export
- request security pki local-certificate enroll cmpv2
- request security pki local-certificate enroll scep

- request security pki local-certificate export
- request security pki local-certificate generate-self-signed (Security)
- request security pki local-certificate load
- request security pki local-certificate re-enroll cmpv2
- request security pki local-certificate re-enroll scep
- request security pki local-certificate verify (Security)
- request security pki verify-integrity-status
- show network-access address-assignment pool (View)
- show security dynamic-policies
- show security dynamic-vpn users
- show security dynamic-vpn users terse
- show security group-vpn member ike security-associations
- show security group-vpn member ipsec inactive-tunnels
- show security group-vpn member ipsec security-associations
- show security group-vpn member ipsec statistics
- show security group-vpn member kek security-associations
- show security group-vpn member policy
- show security group-vpn server ike security-associations
- show security group-vpn server ipsec security-associations
- show security group-vpn server kek security-associations
- show security group-vpn server registered-members
- show security group-vpn server server-cluster
- show security group-vpn server statistics
- show security ike active-peer
- show security ike pre-shared-key
- show security ike security-associations
- show security ike tunnel-map
- show security ipsec control-plane-security-associations
- show security ipsec inactive-tunnels
- show security ipsec next-hop-tunnels
- show security ipsec security-associations
- show security ipsec statistics
- show security ipsec traffic-selector
- show security ipsec tunnel-events-statistics
- show security pki ca-certificate (View)
- show security pki certificate-request (View)
- show security pki crl (View)

- show security pki local-certificate (View)
- show security tcp-encap connection
- show security tcp-encap statistics

clear security dynamic-vpn all

Supported Platforms SRX Series

Syntax clear security dynamic-vpn all

Release Information Command introduced in Junos Release 10.4.

Description Clear all dynamic VPN user connections. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level clear

Related Documentation

- show security dynamic-vpn users on page 1173
- show security dynamic-vpn users terse on page 1175

List of Sample Output clear security dynamic-vpn all on page 1126

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security dynamic-vpn all

```
user@host> clear security dynamic-vpn all
2 user connection entries cleared
```

clear security dynamic-vpn user

Supported Platforms SRX Series

Syntax clear security dynamic-vpn user *username* ike-id *id*

Release Information Command introduced in Junos Release 10.4.

Description Clear the dynamic VPN user connection for the specified username. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level clear

Related Documentation

- show security dynamic-vpn users on page 1173
- show security dynamic-vpn users terse on page 1175

List of Sample Output clear security dynamic-vpn user on page 1127

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security dynamic-vpn user

```
user@host> clear security dynamic-vpn user user ike-id bob.example.net
Connection entry for user user has been cleared
```

clear security group-vpn member group

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn member group <vpn *vpn-name*> <group-id *group-id*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Clear all current information for IKE, TEK, and KEK SAs. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Clear SA information for all groups.

vpn ***vpn-name***—(Optional) Clear SA information for the specified VPN name.

group-id ***group-id***—(Optional) Clear SA information for the specified group identifier.

Required Privilege Level clear

Related Documentation

- Group VPNv2 Overview on page 665

Output Fields This command produces no output.

clear security group-vpn member ike security-associations

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn member ike security-associations [index *SA-index*] [*peer-ipaddress*]

Release Information Command introduced in Junos OS Release 10.2.

Description Clear IKE security association (SA) for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Clear all IKE SAs for the group member.
- **index**—(Optional) Clear the IKE SA with this index number.
- *peer-ipaddress*—(Optional) Clear the IKE SA with this peer.

Required Privilege Level clear

Related Documentation

- **show security group-vpn member ike security-associations on page 1177**
- Group VPNv2 Overview on page 665

Output Fields This command produces no output.

clear security group-vpn member ipsec security-associations

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn member ipsec security-associations [index *SA-index*]

Release Information Command introduced in Junos OS Release 10.2.

Description Clear group VPN SA for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Clear all group VPN SAs for the group member.
- **index**—(Optional) Clear the group VPN SA with this index number.

Required Privilege Level clear

Related Documentation

- **show security group-vpn member ipsec security-associations on page 1184**
- Group VPNv2 Overview on page 665

Output Fields This command produces no output.

clear security group-vpn member ipsec security-associations statistics

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn member ipsec security-associations statistics <group-id *group-id*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Clear IPsec SA statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Clear IPsec SA statistics for all groups.

group-id *group-id*—(Optional) Clear IPsec SA statistics for the specified group identifier.

Required Privilege Level clear

Related Documentation

- Group VPNv2 Overview on page 665

Output Fields This command produces no output.

clear security group-vpn member ipsec statistics

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn member ipsec statistics <index *index*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Clear IPsec statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Clear IPsec statistics for all groups.

index ***index***—(Optional) Clear the IPsec statistics for the SA with this index number.

Required Privilege Level clear

Related Documentation

- Group VPNv2 Overview on page 665

Output Fields This command produces no output.

clear security group-vpn server

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn server [group *group-name* | group-id *group-id*] [now]

Description Clear active members for a specified group. If no options are specified, members are cleared from all groups. After this command is issued, members will need to reregister. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: An IKE SA can be used by a group member to register to multiple groups. When you clear members for a specified group, all existing IKE SAs that could be used to register to the group are also cleared.

Options

- none—All members are cleared from all groups.
- **group**—(Optional) Clear members and SAs for the specified group name.
- **group-id**—(Optional) Clear members and SAs for the specified group identifier.
- **now**—(Optional) Immediately clear all group-related information.

Required Privilege Level clear

Related Documentation

- **show security group-vpn server registered-members on page 1207**
- Group VPNv2 Overview on page 665

Output Fields If there is a problem with the command, one of the following messages appears:

- Group does not exist
- Group is in the process of deletion
- Error in clear members
- Warning Message; Fail to push delete to members as server-member-communication is not configured.

clear security group-vpn server server-cluster statistics

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn server server-cluster statistics <group *group-name*> <group-id *group-id*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Clear Group VPNv2 server cluster statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Clear Group VPNv2 server cluster statistics for all groups.

group ***group-name***—(Optional) Clear Group VPNv2 server cluster statistics for the specified group name.

group-id ***group-id***—(Optional) Clear Group VPNv2 server cluster statistics for the specified group identifier.

Required Privilege Level clear

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Server Clusters on page 714

Output Fields This command produces no output.

clear security group-vpn server statistics

Supported Platforms SRX Series, vSRX

Syntax clear security group-vpn server statistics <group *group-name*> <group-id *group-id*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Clear group statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Clear statistics for all groups.

group *group-name*—(Optional) Clear statistics for the specified group name.

group-id *group-id*—(Optional) Clear statistics for the specified group identifier.

Required Privilege Level clear

Related Documentation

- show security group-vpn server statistics on page 1212
- Group VPNv2 Overview on page 665

Output Fields This command produces no output.

clear security ike respond-bad-spi-count

Supported Platforms SRX Series, vSRX

Syntax clear security ike respond-bad-spi-count
< ***gateway-name*** >

Release Information Command introduced in Junos OS Release 8.5.

Description Clear information about invalid Internet Key Exchange (IKE) security parameter index (SPI) counters.

Options

- none—Clear all invalid SPI counters.
- ***gateway-name*** —(Optional) Clear the invalid SPI counters for the given gateway.

Required Privilege Level clear

Related Documentation

- respond-bad-spi on page 1083

Output Fields This command produces no output.

clear security ike security-associations

Supported Platforms SRX Series, vSRX

Syntax

clear security ike security-associations
< *peer-address* >
< port >
<fpc *slot-number*>
<index *SA-index-number*>
<kmd-instance (all | *kmd-instance-name*)>
<pic *slot-number*>
port
<family (inet | inet6)>

Release Information Command introduced in Junos OS Release 8.5. The **fpc**, **pic**, and **kmd-instance** options added in Junos OS Release 9.3. The **port** option added in Junos OS Release 10.0. The **family** option added in Junos OS Release 11.1.

Description Clear information about the current Internet Key Exchange security associations (IKE SAs). For IKEv2, the device clears the information about the IKE SAs and the associated IPSec SA.

Options

- none—Clear all IKE SAs.
- ***peer-address*** —(Optional) Clear IKE SAs for the destination peer at this IP address.
- **fpc *slot-number*** —Specific to SRX Series devices. Clear information about existing IKE SAs in this Flexible PIC Concentrator (FPC) slot.
- **index *SA-index-number*** —(Optional) Clear the IKE SA with this index number.
- **port**—(Optional) Port number of SA (1 through 65,535).
- **kmd-instance**—Specific to SRX Series devices. Clear information about existing IKE SAs in the key management process (the daemon, which in this case is KMD) identified by FPC ***slot-number*** and PIC ***slot-number***.
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.
- **pic *slot-number*** —Specific to SRX Series devices. Clear information about existing IKE SAs in this PIC slot.
- **family**—(Optional) Clear IKE SAs by family.
  - **inet**—IPv4 address family.
  - **inet6**—IPv6 address family.

Required Privilege Level clear

Related Documentation

- show security ike security-associations on page 1217

Output Fields This command produces no output.

clear security ipsec security-associations

Supported Platforms SRX Series, vSRX

Syntax

clear security ipsec security-associations
fpc *slot-number*
<index *SA-index-number*>
kmd-instance (all | *kmd-instance-name*)
pic *slot-number*
<family (inet | inet6)>

Release Information Command introduced in Junos OS Release 8.5. The **fpc**, **pic,** and **kmd-instance** options added in Junos OS Release 9.3. The **family** option added in Junos OS Release 11.1.

Description Clear information about IPsec security associations (SAs).

Options

- none—Clear all IPsec SAs.
- **fpc** *slot-number* —Specific to SRX Series devices. Clear information about existing IPsec SAs in this Flexible PIC Concentrator (FPC) slot.
- **index** *SA-index-number* —(Optional) Clear the IPsec SA with this index number.
- **kmd-instance**—Specific to SRX Series devices. Clear information about existing IPsec SAs in the key management process (the daemon, which in this case is KMD) identified by FPC *slot-number* and PIC *slot-number* .
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.

pic *slot-number* —Specific to SRX Series devices. Clear information about existing IPsec SAs in this PIC slot.

family—(Optional) Clear SAs by family.

- **inet**—IPv4 address family.
- **inet6**—IPv6 address family.

Required Privilege Level clear

Related Documentation

- show security ipsec security-associations on page 1236

Output Fields This command produces no output.

clear security ipsec statistics

Supported Platforms SRX Series, vSRX

Syntax

```
clear security ike statistics
<fpc slot-number>
<index SA-index-number>
<kmd-instance (all | kmd-instance-name )>
<pic slot-number>
```

Release Information Command introduced in Junos OS Release 8.5. **fpc and pic** options added in Junos OS Release 9.3. **kmd-instance** option added in Junos OS Release 10.4.

Description Clear IPsec statistics on the device.

Options

- none—Clear all IPsec statistics.
- **fpc** ***slot-number*** —Specific to SRX Series devices. Clear statistics about existing IPsec security associations (SAs) in this Flexible PIC Concentrator (FPC) slot.
- **index** ***SA-index-number*** —(Optional) Clear the IPsec statistics for the SA with this index number.
- **kmd-instance**—Specific to SRX Series devices. Clear information about existing IKE SAs in the key management process (the daemon, which in this case is KMD) identified by FPC ***slot-number*** and PIC ***slot-number*** .
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.
- **pic** ***slot-number*** —Specific to SRX Series devices. Clear statistics about existing IPsec SAs in this PIC slot.

Required Privilege Level clear

Related Documentation

- show security ipsec statistics on page 1248

Output Fields This command produces no output.

clear security ipsec tunnel-events-statistics

Supported Platforms SRX Series, vSRX

Syntax clear security ipsec tunnel-events-statistics

Release Information Command introduced in Junos OS Release 12.3X48-D10.

Description Clear IPsec tunnel event statistics.

Required Privilege Level clear

Related Documentation

- *show security ipsec tunnel-events-statistics*

Output Fields This command produces no output.

clear security pki key-pair (Local Certificate)

Supported Platforms SRX Series, vSRX

Syntax clear security pki key-pair (all | certificate-id *certificate-id*)

Release Information Command introduced in Junos OS Release 8.5.

Description Clear public key infrastructure (PKI) key pair information for local digital certificates on the device.

Options
- **all**—Clear key pair information for all local certificates.
- **certificate-id** ***certificate-id*** —Clear key pair information for the local certificate with this certificate ID.

Required Privilege Level clear and security

Related Documentation
- show security pki certificate-request (View) on page 1258

Output Fields This command produces no output.

clear security pki local-certificate (Device)

Supported Platforms SRX Series, vSRX

Syntax clear security pki local-certificate (all | certificate-id *certificate-id* | system-generated)

Release Information Command modified in Junos OS Release 9.1.

Description Clear public key infrastructure (PKI) information for local digital certificates on the device.

Options

- **all**—Clear information for all the local digital certificates on the device.

NOTE: You cannot clear the automatically generated self-signed certificate using **clear security pki local-certificate all** command. To clear the self-signed certificate you need to use **system-generated** as an option.

- **certificate-id** ***certificate-id***—Clear the specified local digital certificate with this certificate ID.
- **system-generated**—Clear the existing automatically generated self-signed certificate and generate a new self-signed certificate.

Required Privilege Level clear and security

Related Documentation

- show security pki local-certificate (View) on page 1264
- request security pki local-certificate generate-self-signed (Security) on page 1160

List of Sample Output clear security pki local-certificate all on page 1143
clear security pki local-certificate system-generated on page 1143

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security pki local-certificate all

```
user@host> clear security pki local-certificate all
```

Sample Output

clear security pki local-certificate system-generated

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

clear security tcp-encap statistics

Supported Platforms SRX Series, vSRX

Syntax clear security tcp-encap statistics

Release Information Command introduced in Junos OS Release 15.1X49-D80. This command is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices.

Description Clear TCP encapsulation statistics.

Required Privilege Level clear

Related Documentation
- show security tcp-encap statistics on page 1271

Output Fields This command produces no output.

request security pki ca-certificate ca-profile-group load

Supported Platforms SRX Series, vSRX

Syntax request security pki ca-certificate ca-profile-group load ca-group-name *ca-group-name* filename [*path/filename* | default]

Release Information Command introduced in Junos OS Release 12.1; **default** option added in Junos OS Release 12.1X47-D10.

Description For SSL forward proxy, you need to load trusted CA certificates on your system. By default, Junos OS provides a list of trusted CA certificates that include default certificates used by common browsers. Alternatively, you can define your own list of trusted CA certificates and import them on to your system.

Use this command to load the default certificates or to specify a path and filename of trusted CA certificates that you define.

Options **ca-group-name *ca-group-name***—Load the specified CA group profile.

filename *path/filename*—Directory location and filename of the trusted CA certificates defined by you.

filename default—Load the trusted CA certificates available by default.

Required Privilege Level maintenance

Related Documentation

- *show security pki ca-certificate*
- Understanding Certificates and PKI on page 353

List of Sample Output
request security pki ca-certificate ca-profile-group load (default) on page 1145
request security pki ca-certificate ca-profile-group load (path/filename) on page 1146

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

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

`user@host>` **request security pki ca-certificate ca-profile-group load ca-group-name ca-default filename default**

```
Do you want to load this CA certificate ? [yes,no] (no) yes
Loading 157 certificates for group 'ca-default'.
ca-default_1: Loading done.
ca-default_2: Loading done.
```

```
ca-default_3: Loading done.
......
```

Sample Output

request security pki ca-certificate ca-profile-group load (path/filename)

`user@host>` **request security pki ca-certificate ca-profile-group load ca-group-name ca-manual filename /var/tmp/firefox-all.pem**

```
Do you want to load this CA certificate ? [yes,no] (no) yes

Loading 196 certificates for group 'ca-manual'.
ca-manual_1_sysgen: Loading done.
ca-manual_2_sysgen: Loading done.
ca-manual_3_sysgen: Loading done.
ca-manual_4_sysgen: Loading done.
ca-manual_5_sysgen: Loading done.
ca-manual_6_sysgen: Loading done.

...
ca-manual_195_sysgen: Loading done.
ca-manual_196_sysgen: Loading done.
ca-profile-group 'ca-manual’ successfully loaded. Success[193] Skipped[3]
```

request security pki ca-certificate enroll (Security)

Supported Platforms SRX Series, vSRX

Syntax request security pki ca-certificate enroll ca-profile *ca-profile-name*

Release Information Command introduced in Junos OS Release 7.5.

Description Request a digital certificate from a certificate authority (CA) online by using the Simple Certificate Enrollment Protocol (SCEP).

Options **ca-profile *ca-profile-name***—CA profile name.

Required Privilege Level maintenance

Related Documentation

- show security pki ca-certificate (View) on page 1254
- Understanding Certificates and PKI on page 353

List of Sample Output request security pki ca-certificate enroll on page 1147

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki ca-certificate enroll

```
user@host> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=example, CN=First Officer
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
  Certificate: C=us, O=example, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=example
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
Do you want to load the above CA certificate ? [yes,no] (no) yes
```


request security pki ca-certificate load (Security)

Supported Platforms SRX Series, vSRX

Syntax request security pki ca-certificate load ca-profile *ca-profile-name* filename *path/filename*

Release Information Command introduced in Junos OS Release 7.5.

Description Manually load a certificate authority (CA) digital certificate from a specified location.

Options **ca-profile *ca-profile-name***—Load the specified CA profile.

filename *path/filename*—Directory location and filename of the CA digital certificate.

Required Privilege Level maintenance

Related Documentation

- *show security pki ca-certificate*
- Understanding Certificates and PKI on page 353

List of Sample Output request security pki ca-certificate load on page 1148

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki ca-certificate load

```
user@host> request security pki ca-certificate load ca-profile 2Kkey filename /var/tmp/2Kkey.pem

Fingerprint:
  a0:08:bb:1f:75:96:76:cd:ee:db:36:10:b6:c6:d8:df:5e:02:05:05 (sha1)
  f5:58:6b:de:7c:d6:cd:90:5a:18:c3:0e:3d:95:da:25 (md5)
Do you want to load this CA certificate ? [yes,no] (no) yes

CA certificate for profile 2Kkey loaded successfully
```

request security pki ca-certificate verify (Security)

Supported Platforms SRX Series, vSRX

Syntax **request security pki ca-certificate verify ca-profile** ***ca-profile-name***

Release Information Command introduced in Junos OS Release 8.5.

Description Verify the digital certificate installed for the specified certificate authority (CA).

Options **ca-profile** ***ca-profile-name*** —Display the specified CA profile.

Required Privilege Level maintenance and security

Related Documentation

- ca-profile (Security PKI) on page 955
- show security pki ca-certificate (View) on page 1254
- Understanding Certificates and PKI on page 353

List of Sample Output request security pki ca-certificate verify ca-profile ca1 (CRL downloaded) on page 1149
request security pki ca-certificate verify ca-profile ca1 (CRL not downloaded) on page 1149

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

This user has downloaded the certificate revocation list (CRL).

request security pki ca-certificate verify ca-profile ca1 (CRL downloaded)

```
user@host> request security pki ca-certificate verify ca-profile ca1
CA certificate ca1 verified successfully
```

Sample Output

This user has not downloaded the certificate revocation list (CRL).

request security pki ca-certificate verify ca-profile ca1 (CRL not downloaded)

```
user@host> request security pki ca-certificate verify ca-profile ca1
CA certificate ca1: CRL verification in progress. Please check the PKId debug
logs for completion status
```


request security pki crl load (Security)

Supported Platforms SRX Series, vSRX

Syntax request security pki crl load ca-profile *ca-profile-name* filename *path/filename*

Release Information Command introduced in Junos OS Release 8.1.

Description Manually install a certificate revocation list (CRL) on the device from a specified location.

Options **ca-profile *ca-profile-name*** —Load the specified certificate authority (CA) profile.

filename *path/filename* —Directory location and filename of the CRL.

Required Privilege Level maintenance

Related Documentation

- Understanding Certificates and PKI on page 353

List of Sample Output request security pki crl load on page 1150

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki crl load

```
user@host> request security pki crl load ca-profile ca-test filename example-inter-ca.crl
CRL for CA profile ca-test loaded successfully
```


request security pki generate-certificate-request (Security)

Supported Platforms SRX Series, vSRX

Syntax

request security pki generate-certificate-request certificate-id *certificate-id-name*
domain-name *domain-name* subject *subject-distinguished-name*
<add-ca-constraint>
<digest (sha1 | sha256)>
<email *email-address*>
<filename (*path* | terminal)>
<ip-address *ip-address*>

Release Information Command introduced in Junos OS Release 7.5. Support for **digest** option added in Junos OS Release 12.1X45-D10.

Description Manually generate a local digital certificate request in the Public-Key Cryptography Standards #10 (PKCS-10) format.

Options

certificate-id *certificate-id-name*—Name of the local digital certificate and the public/private key pair.

domain-name *domain-name*—Fully qualified domain name (FQDN) provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

subject *subject-distinguished-name*—Distinguished name format contains the following information:

- **DC**—Domain component
- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **L**—Locality
- **ST**—State
- **C**—Country

digest—(Optional) Hash algorithm used to sign the certificate request.

- **sha1**—SHA-1 digests (default value for RSA or DSA only).
- **sha256**—SHA-256 digests for RSA or ECDSA only (default value for ECDSA).
- **sha-384**—SHA-384 digests for ECDSA only.

email *email-address*—(Optional) E-mail address of the certificate holder.

filename (*path* | terminal)—(Optional) Location where the local digital certificate request should be placed or the login terminal.

ip-address ***ip-address***—(Optional) IP address of the router.

Required Privilege Level
maintenance

Related Documentation
- show security pki certificate-request (View) on page 1258

List of Sample Output
request security pki generate-certificate-request on page 1152

Output Fields
When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki generate-certificate-request

```
user@host> request security pki generate-certificate-request certificate-id local-entrust2
domain-name router2.example.net filename entrust-req2 subject cn=router2.example.net

Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHAxLmp1bmlwZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFklQws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8o0Xyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH406gQ3G
3iHOZfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ70fO9Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
0d:90:b8:d2:56:74:fc:84:59:62:b9:78:71:9c:e4:9c:54:ba:16:97 (sha1)
1b:08:d4:f7:90:f1:c4:39:08:c9:de:76:00:86:62:b8 (md5)
```

request security pki generate-key-pair (Security)

Supported Platforms SRX Series, vSRX

Syntax request security pki generate-key-pair certificate-id *certificate-id-name*
<size (256 | 384 | 1024 | 2048 | 4096)>
<type (dsa | ecdsa | rsa)>

Release Information Command introduced in Junos OS Release 11.1. Options to support Elliptic Curve Digital Signature Algorithm (ECDSA) added in Junos OS Release 12.1X45-D10.

Description Generate a public key infrastructure (PKI) public/private key pair for a local digital certificate.

Options **certificate-id** ***certificate-id-name***—Name of the local digital certificate and the public/private key pair.

size—Key pair size. The key pair size can be 256, 384, 1024, 2048, or 4096 bits. Key pair sizes of 256 and 384 bits are compatible with ECDSA. If a key pair size is not specified, the default value, **1024** bits, is applied.

type—The algorithm to be used for encrypting the public/private key pair:

- **ecdsa**—ECDSA encryption
- **dsa**—Digital Signal Algorithm (DSA) encryption
- **rsa**—Rivest Shamir Adleman (RSA) encryption (default)

Required Privilege Level maintenance

Related Documentation

- Understanding Certificates and PKI on page 353

List of Sample Output request security pki generate-key-pair on page 1153

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki generate-key-pair

```
user@switch> request security pki generate-key-pair type rsa size 1024 certificate-id test
Generated key pair test, key size 1024 bits
```


request security pki key-pair export

Supported Platforms SRX Series, vSRX

Syntax **request security pki key-pair export certificate-id *certificate-id* filename *filename***
<passphrase *string*>
< type (der | pem)>

Release Information Command introduced in Junos OS Release 15.1X49-D60.

Description Export the keypair for an end-entity (EE) certificate. The exported keypair can be imported along with the EE certificate.

Options **certificate-id *certificate-id***—Name of the local digital certificate.

filename *filename*—Target directory location and filename of the CA digital certificate.

passphrase *passphrase*—(Optional) Passphrase to protect the keypair data for PEM format. The passphrase can be up to 64 characters. If specified, the passphrase must be used when importing the keypair.

type (der | pem)—(Optional) Type of format, either DER or PEM. PEM is the default.

Required Privilege Level maintenance

Related Documentation

- request security pki local-certificate export on page 1159

Output Fields This command produces no output.

request security pki local-certificate enroll cmpv2

Supported Platforms SRX Series, vSRX

Syntax

```
request security pki local-certificate enroll cmpv2
  ca-dn subject-dn
  ca-profile ca-profile name
  ca-reference reference
  ca-secret shared-secret
  certificate-id certificate-id-name
  domain-name domain-name
  email email-address
  ip-address ip-address
  ipv6-address ipv6-address
  subject subject-distinguished-name
```

Release Information Command introduced in Junos OS Release 15.1X49-D40.

Description Enroll and install a local digital certificate online by using CMPv2. This command loads both end-entity (EE) and CA certificates based on the CA server configuration. Certificate revocation list (CRL) or Online Certificate Status Protocol (OCSP) can be used to check the revocation status of a certificate.

Options

ca-dn *subject-dn*—The distinguished name (DN) of the CA enrolling the EE certificate must be specified during enrollment. This optional parameter is mandatory if the CA certificate is not already enrolled. If the CA certificate is already enrolled, the subject DN is extracted from the CA certificate.

ca-profile *ca-profile-name*—CA profile name.

ca-reference *reference*—Out-of-band reference value received from the CA server.

ca-secret *shared-secret*—Out-of-band secret value received from the CA server.

certificate-id *certificate-id-name*—Name of the local digital certificate and the public/private key pair.

domain-name *domain-name*—Fully qualified domain name (FQDN). The FQDN provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

email *email-address*—E-mail address of the certificate holder.

ip-address *ip-address*—IP address of the router.

ipv6-address *ipv6-address*—IPv6 address of the router for the alternate subject.

subject *subject-distinguished-name*—Distinguished Name (DN) format that contains the domain component, common name, department, serial number, company name, state, and country in the following format: DC, CN, OU, O, SN, L, ST, C.

- **DC**—Domain component
- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **SN**—Serial number of the device

NOTE: 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).

- **ST**—State
- **C**—Country

Required Privilege Level maintenance and security

Related Documentation

- show security pki local-certificate (View) on page 1264
- clear security pki local-certificate (Device) on page 1143

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

```
user@host> request security pki local-certificate enroll cmpv2 ca-profile root-552 ca-dn
DC=example,CN=root-552 certificate-id tc552 email tc552-root@example.net domain-name
example.net ip-address 192.0.2.22 ca-secret example ca-reference 51892 subject
CN=example,OU=SBU,O=552-22

Certificate enrollment has started. To view the status of your enrollment, check
 the public key infrastructure log (pkid) log file at /var/log/pkid.
```


request security pki local-certificate enroll scep

Supported Platforms SRX Series, vSRX

Syntax

```
request security pki local-certificate enroll scep
  ca-profile ca-profile name
  certificate-id certificate-id-name
  challenge-password challenge-password
  digest (sha-1 | sha-256)
  domain-name domain-name
  email email-address
  ip-address ip-address
  ipv6-address ipv6-address
  scep-digest-algorithm (md5 | sha-1)
  scep-encryption-algorithm (des | des3)
  subject subject-distinguished-name
```

Release Information Command introduced in Junos OS Release 9.1. Serial number (SN) option added to the subject string output field in Junos OS Release 12.1X45. **scep** keyword and **ipv6-address** option added in Junos OS Release 15.1X49-D40.

Description Enroll and install a local digital certificate online by using Simple Certificate Enrollment Protocol (SCEP).

Options

ca-profile *ca-profile-name*—CA profile name.

certificate-id *certificate-id-name*—Name of the local digital certificate and the public/private key pair.

challenge-password *password*—Password set by the administrator and normally obtained from the SCEP enrollment webpage of the CA. The password is 16 characters in length.

digest (sha-1 | sha-256)—Hash algorithm used for signing RSA certificates, either SHA-1 or SHA-256. SHA-1 is the default.

domain-name *domain-name*—Fully qualified domain name (FQDN). The FQDN provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

email *email-address*—E-mail address of the certificate holder.

ip-address *ip-address*—IP address of the router.

ipv6-address *ipv6-address*—IPv6 address of the router for the alternate subject.

scep-digest-algorithm (md5 | sha-1)—Hash algorithm digest, either MD5 or SHA-1; SHA-1 is the default.

scep-encryption-algorithm (des | des3)—Encryption algorithm, either DES or DES3; DES3 is the default.

subject ***subject-distinguished-name***—Distinguished Name (DN) format that contains the domain component, common name, department, serial number, company name, state, and country in the following format: DC, CN, OU, O, SN, L, ST, C.

- **DC**—Domain component
- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **SN**—Serial number of the device

NOTE: 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).

- **ST**—State
- **C**—Country

Required Privilege Level maintenance and security

Related Documentation

- show security pki local-certificate (View) on page 1264
- clear security pki local-certificate (Device) on page 1143

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

```
user@host> request security pki local-certificate enroll scep certificate-id r3-entrust-scep
ca-profile entrust domain-name router3.example.net subject
"CN=router3,OU=Engineering,O=example,C=US" challenge-password 123

Certificate enrollment has started. To view the status of your enrollment, check
 the public key infrastructure log (pkid) log file at /var/log/pkid. Please save
 the challenge-password for revoking this certificate in future.  Note that this
 password is not stored on the router.
```

request security pki local-certificate export

Supported Platforms SRX Series, vSRX

Syntax request security pki local-certificate export

Release Information Command introduced in Junos OS Release 12.1.

Description Export a generated self-signed certificate from the default location (var/db/certs/common/local) to a specific location within the device.

Options **certificate id** ***certificate-id-name***—Name of the local digital certificate.

filename ***path/filename***—Target directory location and filename of the CA digital certificate.

type (der | pem)—Certificate format: DER (distinguished encoding rules) or PEM (privacy-enhanced mail).

Required Privilege Level maintenance

Related Documentation

- Understanding Certificates and PKI on page 353

List of Sample Output request security pki local-certificate export on page 1159

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki local-certificate export

```
user@host> request security pki local-certificate export filename /var/tmp/my-cert.pem
certificate-id nss-cert type pem
certificate exported successfully
```


request security pki local-certificate generate-self-signed (Security)

Supported Platforms SRX Series, vSRX

Syntax

request security pki local-certificate generate-self-signed certificate-id
  *certificate-id-name*domain-name *domain-name* subject *subject-distinguished-name*
<add-ca-constraint>
<digest (sha1 | sha256)>
<email *email-address*>
<ip-address *ip-address*>

Release Information Command introduced in Junos OS Release 9.1. Support for **digest** option added in Junos OS Release 12.1X45-D10.

Description Manually generate a self-signed certificate for the given distinguished name.

Options

certificate-id ***certificate-id-name***—Name of the certificate and the public/private key pair.

domain-name ***domain-name***—Fully qualified domain name (FQDN) provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

subject ***subject-distinguished-name***—Distinguished name format contains the following information:

- **DC**—Domain component
- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **L**—Locality
- **ST**—State
- **C**—Country

add-ca-constraint—(Optional) Specifies that the certificate can be used to sign other certificates.

digest—(Optional) Hash algorithm used to sign the certificate.

- **sha1**—SHA-1 digest (default)
- **sha256**—SHA-256 digest

email ***email-address***—(Optional) E-mail address of the certificate holder.

Required Privilege Level maintenance and security

Related Documentation

- clear security pki local-certificate (Device) on page 1143
- show security pki local-certificate (View) on page 1264

List of Sample Output

request security pki local-certificate generate-self-signed certificate-id self-cert subject cn=abc domain-name example.net email mholmes@example.net on page 1161

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki local-certificate generate-self-signed certificate-id self-cert subject cn=abc domain-name example.net email mholmes@example.net

```
user@host> request security pki local-certificate generate-self-signed certificate-id self-cert
subject cn=abc domain-name example.net email mholmes@example.net
Self-signed certificate generated and loaded successfully
```

request security pki local-certificate load

Supported Platforms SRX Series, vSRX

Syntax request security pki local-certificate load filename *ssl_proxy_ca.crt* key *ssl_proxy_ca.key* certificate-id *certificate id*

Release Information Command introduced in Junos OS Release 11.4.

Description Manually load a local digital certificate from a specified location.

Options **filename** — Filename that contains the certificate to load

key— File pathname that contains the private key/key-pair to loaded

certificate-id —Name of the certificate identifier

Required Privilege Level maintenance and security

Related Documentation

- show security pki local-certificate (View) on page 1264
- clear security pki local-certificate (Device) on page 1143
- request security pki local-certificate verify (Security) on page 1165
- Understanding Certificates and PKI on page 353

List of Sample Output request security pki local-certificate load on page 1162

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki local-certificate load

```
user@host> request security pki local-certificate load filename cert_name.crt key key_name.key
certificate-id test
Local certificate cert_name.crt loaded successfully
```


request security pki local-certificate re-enroll cmpv2

Supported Platforms SRX Series, vSRX

Syntax

request security pki local-certificate re-enroll cmpv2 certificate-id *certificate-id*
<ca-profile-name *ca-profile*>
<re-generate-keypair>

Release Information Command introduced in Junos OS Release 15.1X49-D60.

Description Manually reenroll an end-entity (EE) certificate with Certificate Management Protocol version 2 (CMPv2). This command allows the administrator to initiate renewal of the EE certificate using CMPv2 and can be used in conjunction with the **set security pki auto-re-enrollment cmpv2** automatic enrollment configuration.

Options

certificate-id ***certificate-id-name***—Name of the local digital certificate.

ca-profile-name ***ca-profile-name***—(Optional) CA profile name.

re-generate-keypair—(Optional) Generate a PKI public/private key pair for the EE certificate.

NOTE: Key generation might take a few seconds.

Required Privilege Level maintenance and security

Related Documentation

- request security pki local-certificate enroll cmpv2 on page 1155

Output Fields This command produces no output.

request security pki local-certificate re-enroll scep

Supported Platforms SRX Series, vSRX

Syntax

request security pki local-certificate re-enroll scep certificate-id *certificate-id*
<ca-profile-name *ca-profile*>
<challenge-password *password*>
<re-generate-keypair>
<scep-digest-algorithm (md5 | sha-1)>
<scep-encryption-algorithm (des | des3)>

Release Information Command introduced in Junos OS Release 15.1X49-D60.

Description Manually reenroll an end-entity (EE) certificate with Simple Certificate Enrollment Protocol (SCEP). This command allows the administrator to initiate renewal of the EE certificate using SCEP and can be used in conjunction with the **set security pki auto-re-enrollment scep** automatic enrollment configuration.

Options

certificate-id ***certificate-id-name***—Name of the local digital certificate.

ca-profile-name ***ca-profile-name***—(Optional) CA profile name.

challenge-password ***password***—(Optional) Password set by the administrator and normally obtained from the SCEP enrollment webpage of the CA. The password is 16 characters in length.

re-generate-keypair—(Optional) Generate a PKI public/private key pair for the EE certificate.

NOTE: Key generation might take a few seconds.

scep-digest-algorithm —(Optional) Hash algorithm digest, either MD5 or SHA-1; SHA-1 is the default.

scep-encryption-algorithm —(Optional) Encryption algorithm, either DES or DES3; DES3 is the default.

Required Privilege Level maintenance and security

Related Documentation

- request security pki local-certificate enroll scep on page 1157

Output Fields This command produces no output.

request security pki local-certificate verify (Security)

Supported Platforms SRX Series, vSRX

Syntax request security pki local-certificate verify certificate-id *certificate-id-name*

Release Information Command introduced in Junos OS Release 8.5.

Description Verify the validity of the local digital certificate identifier.

Options certificate-id *certificate-id-name* — Name of the local digital certificate identifier.

Required Privilege Level maintenance and security

Related Documentation

- request security pki local-certificate load on page 1162
- show security pki local-certificate (View) on page 1264
- clear security pki local-certificate (Device) on page 1143
- Understanding Certificates and PKI on page 353

List of Sample Output request security pki local-certificate verify certificate-id bme1 (not downloaded) on page 1165
request security pki local-certificate verify certificate bme1 (downloaded) on page 1165

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

You receive the following response before the certificate revocation list (CRL) is downloaded:

request security pki local-certificate verify certificate-id bme1 (not downloaded)

```
user@host> request security pki local-certificate verify certificate-id bme1
Local certificate bme1: CRL verification in progress. Please check the PKId debug
 logs for completion status
```

Sample Output

You receive the following response after the certificate revocation list (CRL) is downloaded:

request security pki local-certificate verify certificate bme1 (downloaded)

```
user@host> request security pki local-certificate verify certificate-id bme1
Local certificate bme1 verification success
```


request security pki verify-integrity-status

Supported Platforms SRX Series, vSRX

Syntax request security pki verify-integrity-status

Release Information Command introduced in Junos OS Release 11.2.

> NOTE: Do not use this command for non-FIPS or Common Criteria releases. We recommend that you do not use this command for any Junos OS Release 15.1X49-D40 or later releases.

Description Verify the integrity of public key infrastructure (PKI) files. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices and vSRX instances.

Required Privilege Level maintenance

Related Documentation

List of Sample Output request security pki verify-integrity-status on page 1166

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki verify-integrity-status

```
user@host> request security pki verify-integrity-status
All PKI objects: verification success
```


show network-access address-assignment pool (View)

Supported Platforms SRX Series

Syntax show network-access address-assignment pool *name*

Release Information Command introduced in Junos OS Release 10.4.

Description Display information summary about a specific pool.

Required Privilege Level view

Output Fields Table 91 on page 1167 lists the output fields for the **show network-access address-assignment pool** command. Output fields are listed in the approximate order in which they appear.

Table 91: show network-access address-assignment pool Output Fields

Field Name	Field Description
IP address	IP address assigned to a client.
Hardware address	MAC address of the client. For XAuth clients, the value is NA.
Host/User	For static IP address assignment, the user name and profile are displayed in the format *username@profile*. If the client is assigned an IP address from an address pool and a user name exists, the user name is displayed. For DHCP applications, if the host name is configured the host name is displayed; otherwise NA is displayed.
Type	Either XAuth or DHCP attributes are configured.

Sample Output

```
user@host> show network-access address-assignment pool xauth1
IP address       Hardware address        Host/User                 Type
192.0.2.1         NA                      jason@dvpn-auth           XAUTH
192.0.2.2         NA                      jacky                     XAUTH
192.0.2.3         00:00:5E:00:53:01       host1                     DHCP
192.0.2.4         00:00:5E:00:53:02       NA                        DHCP
```


show security dynamic-policies

Supported Platforms SRX Series

Syntax show security dynamic-policies [detail] [from-zone *zone*] [scope-id *id*] [to-zone *zone*]

Release Information Command introduced in Junos OS Release 10.2.

Description Display dynamic policies downloaded on the group member. This command is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

Options

- none—Display basic information about all policies installed on the group member.
- **detail**—(Optional) Display a detailed view of all of the policies installed on the group member.
- **from-zone**—(Optional) Display information about the policies installed on the group member for the specified source zone.
- **scope-id**—(Optional) Display information about the policies installed on the group member for the specified policy identifier.
- **to-zone**—(Optional) Display information about the policies installed on the group member for the specified destination zone.

Required Privilege Level view

Related Documentation

- *show security policies*
- Group VPNv2 Overview on page 665

List of Sample Output

Output Fields Table 92 on page 1168 lists the output fields for the **show security dynamic-policies** command. Output fields are listed in the approximate order in which they appear.

Table 92: show security dynamic-policies Output Fields

Field Name	Field Description
Policy	Name of the applicable Policy.

Table 92: show security dynamic-policies Output Fields *(continued)*

Field Name	Field Description
State	Status of the policy: • **enabled**: The policy can be used in the policy lookup process, which determines access rights for a packet and the action taken in regard to it. • **disabled**: The policy cannot be used in the policy lookup process, and therefore it is not available for access control.
Index	An internal number associated with the policy.
Scope Policy	Policy identifier.
Sequence number	Number of the policy within a given context. For example, three policies that are applicable in a from-zoneA-to-zoneB context might be ordered with sequence numbers 1, 2, and 3. Also, in a from-zoneC-to-zoneD context, four policies might have sequence numbers 1, 2, 3, and 4.
Source addresses	For standard display mode, the names of the source addresses for a policy. Address sets are resolved to their individual names. (In this case, only the names are given, not their IP addresses.) For detail display mode, the names and corresponding IP addresses of the source addresses for a policy. Address sets are resolved to their individual address name-IP address pairs.
Destination addresses	Name of the destination address (or address set) as it was entered in the destination zone's address book. A packet's destination address must match this value for the policy to apply to it.
Application	Name of a preconfigured or custom application whose type the packet matches, as specified at configuration time. • **IP protocol**: The IP protocol used by the application—for example, TCP, UDP, ICMP. • **ALG**: If an ALG is associated with the session, the name of the ALG. Otherwise, 0. • **Inactivity timeout**: Elapse time without activity after which the application is terminated. • **Source port range**: The low-high source port range for the session application. • **Destination port range**: The low-high destination port range for the session application.
action-type	Must be permit.
Policy Type	Must be dynamic.
From zone	Name of the source zone.
To zone	Name of the destination zone.
Tunnel	Tunnel name, type (IPsec), and index number.

Sample Output

show security dynamic-policies

```
user@host> show security dynamic-policies
Policy: policy_forward-0001, State: enabled, Index: 1048580, Scope Policy: 4
  Sequence number: 1
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
    Applications: Unknown
action-type: permit, tunnel:
Policy: policy_forward-0002, State: enabled, Index: 2097156, Scope Policy: 4
  Sequence number: 2
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
    Applications: Unknown
action-type: permit, tunnel:
```

Sample Output

show security dynamic-policies detail

```
user@host> show security dynamic-policies detail
Policy: policy_forward-0001, action-type: permit, State: enabled, Index:
1048580,AI: disabled, Scope Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Host, To zone: untrust
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [23-23]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1001
Policy: policy_backward-0001, action-type: permit, State: enabled, Index:
1048582,AI: disabled, Scope Policy: 6
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: Host
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1003
Policy: policy_internal-0001, action-type: permit, State: enabled, Index:
1048583,AI: disabled, Scope Policy: 7
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: Host
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1005
Policy: policy_external-0001, action-type: permit, State: enabled, Index:
```

```
1048584,AI: disabled, Scope Policy: 8
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: untrust
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1006
Policy: policy_forward-0002, action-type: permit, State: enabled, Index:
2097156,AI: disabled, Scope Policy: 4
  Policy Type: Dynamic
  Sequence number: 2
  From zone: Host, To zone: untrust
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1002
Policy: policy_backward-0002, action-type: permit, State: enabled, Index:
2097158,AI: disabled, Scope Policy: 6
  Policy Type: Dynamic
  Sequence number: 2
  From zone: untrust, To zone: Host
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [23-23]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1004
```

Sample Output

show security dynamic-policies from-zone Internal

```
user@host> show security dynamic-policies from-zone Internal
Policy: policy_internal-0001, State: enabled, Index: 1048583, Scope Policy: 7
  Sequence number: 1
    Applications: Unknown
action-type: permit, tunnel:
Policy: policy_external-0001, State: enabled, Index: 1048584, Scope Policy: 8
  Sequence number: 1
    Applications: Unknown
action-type: permit, tunnel:
```

Sample Output

show security dynamic-policies scope-id 8 from-zone Internal

```
user@host> show security dynamic-policies scope-id 8 from-zone Internal
Policy: policy_external-0001, State: enabled, Index: 1048584, Scope Policy: 8
  Sequence number: 1
    Applications: Unknown
action-type: permit, tunnel:
```

Sample Output

show security dynamic-policies detail from-zone Internal

```
user@host> show security dynamic-policies detail from-zone Internal
Policy: policy_internal-0001, action-type: permit, State: enabled, Index:
1048583,AI: disabled, Scope Policy: 7
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: Host
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1005
Policy: policy_external-0001, action-type: permit, State: enabled, Index:
1048584,AI: disabled, Scope Policy: 8
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: untrust
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1006
```

Sample Output

show security dynamic-policies detail from-zone Internal to-zone Host

```
user@host> show security dynamic-policies detail from-zone Internal to-zone Host
Policy: policy_internal-0001, action-type: permit, State: enabled, Index:
1048583,AI: disabled, Scope Policy: 7
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: Host
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1005
```

show security dynamic-vpn users

Supported Platforms SRX Series

Syntax show security dynamic-vpn users

Release Information Command introduced in Junos OS Release 10.0.

Description Display all relevant user information. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level view

Related Documentation

- show security dynamic-vpn users terse on page 1175
- clear security dynamic-vpn user on page 1127
- clear security dynamic-vpn all on page 1126
- Dynamic VPN Overview on page 849

Output Fields Table 93 on page 1173 lists the output fields for the **show security dynamic-vpn users** command. Output fields are listed in the approximate order in which they appear.

Table 93: show security dynamic-vpn users Output Fields

Field Name	Field Description
User	Username.
User-groups	Remote IPSec VPN usergroups
Number of connections	Number of connections currently active.
Remote IP	IP address of the client.
IPsec VPN	Name of the IPsec VPN.
IKE gateway	Name of the IKE gateway.
IKE ID	IKE ID configured for the client.
Status	Status of the connection.

Sample Output

`user@host>` **show security dynamic-vpn users**

```
User: alice , User group: group-one , Number of connections: 1
Remote IP: 192.168.2.10
    IPSEC VPN: dyn_vpn2
    IKE gateway: gw2
    IKE ID   : alicegw2.example.net
    IKE Lifetime: 72000
    IPSEC Lifetime: 3600
    Status: CONNECTED
```


show security dynamic-vpn users terse

Supported Platforms SRX Series

Syntax show security dynamic-vpn users terse

Release Information This command introduced in Junos OS Release 10.0.

Description Display all relevant user information. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Required Privilege Level view

Related Documentation

- show security dynamic-vpn users on page 1173
- clear security dynamic-vpn user on page 1127
- clear security dynamic-vpn all on page 1126
- Dynamic VPN Overview on page 849

Output Fields Table 94 on page 1175 lists the output fields for the **show security dynamic-vpn users terse** command. Output fields are listed in the approximate order in which they appear.

Table 94: show security dynamic-vpn users terse Output Fields

Field Name	Field Description
User	Username.
User-groups	Remote IPSec VPN usergroups
Remote IP	IP address of the client.
IKE ID	IKE ID configured for the client.
Status	Status of the connection.
Client Config Name	Name of the client configuration.
Time Established	Time that the user connection was established.

Sample Output

user@host> **show security dynamic-vpn users terse**

```
User   User     Remote      IKE      Status    IKE      IPSEC    Client Time
       Groups   IP          ID                 Lifetime Lifetime Config
```



```
Established
                                                              Name
alice group-one 192.168.2.10 alicegw2.CONNECTED 72000    3600     group   Wed
                             example.                                     Aug 8
10:
                             net                                          26:39
2012
```

show security group-vpn member ike security-associations

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn member ike security-associations [brief | detail] [index *sa-index*] [*peer-ipaddress*]

Release Information Command introduced in Junos OS Release 10.2.

Description Display IKE security associations (SAs) for group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Display summary information about all IKE SAs for the group members.
- **brief**—(Optional) Display summary output.
- **detail**—(Optional) Display detailed output.
- **index** *sa-index*—(Optional) Display detailed information about the specified SA identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.
- *peer-ipaddress*—(Optional) Display information about the SA with the specified peer.

Required Privilege Level view

Related Documentation

- clear security group-vpn member ike security-associations on page 1129
- Group VPNv2 Overview on page 665

List of Sample Output

show security group-vpn member ike security-associations on page 1179
show security group-vpn member ike security-associations detail on page 1179

Output Fields Table 95 on page 1177 lists the output fields for the **show security group-vpn member ike security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 95: show security group-vpn member ike security-associations Output Fields

Field Name	Field Description
Index	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
State	State of the IKE security associations: • **DOWN**—SA has not been negotiated with the peer. • **UP**—SA has been negotiated with the peer.

Table 95: show security group-vpn member ike security-associations Output Fields *(continued)*

Field Name	Field Description
Initiator cookie	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
Responder cookie	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received. A cookie is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine the cookie's authenticity.
Mode	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • **main**—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • **aggressive**—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
Remote Address	IP address of the destination peer with which the local peer communicates.
IKE Peer	IP address of the destination peer with which the local peer communicates.
Exchange type	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • **main**—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • **aggressive**—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
Authentication method	Method the server uses to authenticate the source of IKE messages: • **pre-shared-keys**—Preshared key for encryption and decryption that both participants must have before beginning tunnel negotiations.
Local	Address of the local peer.
Lifetime	Number of seconds remaining until the IKE SA expires.

Table 95: show security group-vpn member ike security-associations Output Fields *(continued)*

Field Name	Field Description

Algorithms	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process:
	• **Authentication**—Type of authentication algorithm used.
	  ◦ **sha-256**—Secure Hash Algorithm 256 authentication.
	  ◦ **sha-384**—Secure Hash Algorithm 384 authentication.
	• **Encryption**—Type of encryption algorithm used.
	  ◦ **aes-256-cbc**—Advanced Encryption Standard (AES) 256-bit encryption.
	  ◦ **aes-192-cbc**— AES192-bit encryption
	  ◦ **aes-128-cbc**—AES 128-bit encryption.

Traffic statistics	• **Input bytes—**Number of bytes received. • **Output bytes—**Number of bytes transmitted. • **Input packets—**Number of packets received. • **Output packets—**Number of packets transmitted.

Sample Output

show security group-vpn member ike security-associations

```
user@host> show security group-vpn member ike security-associations
Index    State  Initiator cookie  Responder cookie  Mode           Remote Address

4736345 UP     70611c65603d53da  6e0888777ad10f8d  Main           192.0.2.3
```

Sample Output

show security group-vpn member ike security-associations detail

```
user@host> show security group-vpn member ike security-associations detail
IKE peer 192.0.2.5, Index 5824842, Gateway Name: group1_2
  Role: Initiator, State: UP
  Initiator cookie: fc866556b8afe4cd, Responder cookie: 1238de6b8a89de44
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 192.0.2.7:848, Remote: 192.0.2.5:848
  Lifetime: Expires in 2 seconds
  Peer ike-id: 192.0.2.5
  Xauth user-name: not available
  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  :                2044
   Output bytes  :                 900
   Input  packets:                   7
   Output packets:                   7
  Flags: IKE SA is created
```

show security group-vpn member ipsec inactive-tunnels

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn member ipsec inactive-tunnels <brief> <detail> <group-id *group-id*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Show inactive Group VPNs. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Display information for all groups.

brief—(Optional) Display summary output.

detail—(Optional) Display detailed output.

group-id ***group-id***—(Optional) Display information for the specified group identifier.

Required Privilege Level view

Related Documentation

- Group VPNv2 Overview on page 665

List of Sample Output
show security group-vpn member ipsec inactive-tunnels on page 1182
show security group-vpn member ipsec inactive-tunnels detail on page 1183

Output Fields Table 96 on page 1181 lists the output fields for the **show security group-vpn member ipsec inactive-tunnels** command. Output fields are listed in the approximate order in which they appear.

Table 96: show security group-vpn member ipsec inactive-tunnels Output Fields

Field Name	Field Description
Server	Server on which group member is registered.
Port	UDP port number.
GId	Group identifier.
lsys	Logical system.

Table 96: show security group-vpn member ipsec inactive-tunnels Output Fields *(continued)*

Field Name	Field Description
Reason	Reason that the tunnel is inactive: • The tunnel was cleared through the CLI. • The hard lifetime has expired. • There are too many TEKs. • There was a configuration change. • There was an SA installation error. • The TEK is stale. • The tunnel was deleted from the server.
Virtual-system	Logical system name.
Group VPN Name	Name of the Group VPN.
Local Gateway	IP address of the local IKE gateway.
GDOI Server	IP address of the group server.
Group Id	Group identifier.
Recovery Probe	Status of the recovery probe, either enabled or disabled (default).
DF-bit	Fragmentation of IPsec traffic on the group member—clear (default), copy, or set.
Stats	Statistics for GDOI groupkey-pull and groupkey-push exchanges, server failovers, deletes received, number of times the maximum number of keys and policies were exceeded, and the number of unsupported algorithms received.
Down Reason	Reason that the tunnel is inactive: • The tunnel was cleared through the CLI. • The hard lifetime has expired. • There are too many TEKs. • There was a configuration change. • There was an SA installation error. • The TEK is stale. • The tunnel was deleted from the server. • The tunnel is not initiated.

Sample Output

show security group-vpn member ipsec inactive-tunnels

`user@host>` **show security group-vpn member ipsec inactive-tunnels**

```
Total inactive tunnels: 1
Server           Port  GId lsys  Reason
192.168.1.50       848   1000 root uninitiated
```

show security group-vpn member ipsec inactive-tunnels detail

```
user@host> show security group-vpn member ipsec inactive-tunnels detail
  Virtual-system: root Group VPN Name: group1000
  Local Gateway: 192.168.1.101, GDOI Server: 192.168.1.50
  Group Id: 1000
  Recovery Probe: Disabled
  DF-bit: clear
  Stats:
      Pull Succeeded             :   0
      Pull Failed                :   8841
      Pull Timeout               :   7996
      Pull Aborted               :   0
      Push Succeeded             :   0
      Push Failed                :   0
      Server Failover            :   0
      Delete Received            :   0
      Exceed Maximum Keys(4)     :   0
      Exceed Maximum Policies(10):   0
      Unsupported Algo           :   0
  Down Reason: uninitiated
```


show security group-vpn member ipsec security-associations

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn member ipsec security-associations [brief | detail] [index *sa-index*]

Release Information Command introduced in Junos OS Release 10.2.

Description Display group VPN security associations (SAs) for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Display information about all group VPN SAs for the group member.
- **brief**—(Optional) Display summary output.
- **detail**—(Optional) Display detailed output.
- **index** *sa-index*—(Optional) Display detailed information about the specified SA identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.

Required Privilege Level view

Related Documentation

- clear security group-vpn member ipsec security-associations on page 1130
- Group VPNv2 Overview on page 665

List of Sample Output show security group-vpn member ipsec security-associations on page 1186
show security group-vpn member ipsec security-associations detail on page 1186

Output Fields Table 97 on page 1184 lists the output fields for the **show security group-vpn member ipsec security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 97: show security group-vpn member ipsec security-associations

Field Name	Field Description
Total active tunnels	Total number of active IPsec tunnels.
ID	Index number of the SA. You can use this number to get additional information about the SA.
Server	IP address of the group server (remote gateway).
Port	If Network Address Translation-Traversal (NAT-T) is used, this value is 4500. Otherwise it is the standard IKE port, 500.

Table 97: show security group-vpn member ipsec security-associations *(continued)*

Field Name	Field Description
Algorithm	Cryptography used to secure exchanges between peers during the IKE Phase 2 negotiations includes • An authentication algorithm used to authenticate exchanges between the peers. Options are **sha-256** or **sha-384** • An encryption algorithm used to encrypt data traffic. Options are **aes-128**, **aes-192**, and **aes-256**.
SPI	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI.
Life: sec/kb	The lifetime of the SA, after which it expires, expressed either in seconds or kilobytes.
GId	Group identifier.
vsys or Virtual-system	The root system.
Local Gateway	Gateway address of the local system.
GDOI Server	IP address of the group server.
Local Identity	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IPv4 address, fully qualified domain name, e-mail address, or distinguished name.
Remote Identity	IPv4 address of the destination peer gateway.
DF-bit	State of the don't fragment bit: set or cleared.
Policy name	Name of the applicable policy.
Direction	Direction of the security association; it can be inbound or outbound.
AUX-SPI	Value of the auxiliary security parameter index. • When the value is AH or ESP, AUX-SPI is always 0. • When the value is AH+ESP, AUX-SPI is always a positive integer.
Hard lifetime	The hard lifetime specifies the lifetime of the SA. • **Expires in seconds**—Number of seconds left until the SA expires.
Lifesize Remaining	The lifesize remaining specifies the usage limits in kilobytes. If there is no lifesize specified, it shows unlimited. • **Expires in kilobytes**—Number of kilobytes left until the SA expires.

Table 97: show security group-vpn member ipsec security-associations *(continued)*

Field Name	Field Description
Soft lifetime	The soft lifetime informs the IPsec key management system that the SA is about to expire. Each lifetime of a security association has two display options, hard and soft, one of which must be present for a dynamic security association. This allows the key management system to negotiate a new SA before the hard lifetime expires. • **Expires in seconds**—Number of seconds left until the SA expires.
Mode	Mode of the security association: • transport—Protects host-to-host connections. • tunnel—Protects connections between security gateways.
Protocol	Protocol supported. Transport mode supports Encapsulation Security Protocol (ESP).
Anti-replay service	State of the service that prevents packets from being replayed. It can be **Enabled** or **Disabled**.

Sample Output

show security group-vpn member ipsec security-associations

```
user@host> show security group-vpn member ipsec security-associations
  Total active tunnels: 2
  ID    Server            Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>49157 192.168.1.53     848   ESP:3des/sha1   c0792f86 114/  unlim   2000 root

  <>49156 192.168.1.53     848   ESP:aes-256/md5 7def169d 18/   unlim   2000 root

  <>49156 192.168.1.53     848   ESP:aes-256/md5 86c48448 146/  unlim   2000 root
```

Sample Output

show security group-vpn member ipsec security-associations detail

```
user@host> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: group2000
  Local Gateway: 192.168.1.70, GDOI Server: 192.168.1.53
  Group Id: 2000
  Routing Instance: vr1
  Recovery Probe: Enabled
  DF-bit: clear

  Stats:
      Pull Succeeded            :   3
      Pull Failed               :   0
      Pull Timeout              :   6
      Pull Aborted              :   0
      Push Succeeded            :   1773
      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: 49157
      Source IP: ipv4_subnet(any:900,[0..7]=192.168.1.0/24)
      Destination IP: ipv4_subnet(any:901,[0..7]=192.168.1.0/24)

      Direction: bi-directional, SPI: c0792f86
      Protocol: ESP, Authentication: sha1, Encryption: 3des
      Hard lifetime: Expires in 81 seconds, Activated
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expired
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled, Window size: 3000 milliseconds

      Direction: bi-directional, SPI: a645b381
      Protocol: ESP, Authentication: sha1, Encryption: 3des
      Hard lifetime: Expires in 207 seconds, Activated in 51 seconds
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 117 seconds
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled, Window size: 3000 milliseconds
```


show security group-vpn member ipsec statistics

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn member ipsec statistics <index *index*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Show IPsec statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Display information for all IPsec SAs.

index ***index***—(Optional) Display detailed information about the specified SA, identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.

Required Privilege Level view

Related Documentation

- Group VPNv2 Overview on page 665

List of Sample Output show security group-vpn member ipsec statistics on page 1189

Output Fields Table 98 on page 1188 lists the output fields for the **show security group-vpn member ipsec statistics** command. Output fields are listed in the approximate order in which they appear.

Table 98: show security group-vpn member ipsec statistics Output Fields

Field Name	Field Description
ESP Statistics	Numbers of encrypted and decrypted bytes and encrypted and decrypted packets.
AH Statistics	Numbers of input and output bytes and input and output packets.
Errors	Numbers of AH failures, replay errors, ESP authentication failures, ESP decryption failures, bad headers, and bad trailers.
D3P Statistics	Numbers of old timestamp packets, new timestamp packets, no timestamp packets, unexpected D3P header packets, invalid type packets, invalid length packets, and invalid next header packets.
Exclude Statistics	Numbers of created and invalidated sessions.
Dynamic Policy Statistics	Numbers of created and invalidated sessions.

Table 98: show security group-vpn member ipsec statistics Output Fields *(continued)*

Field Name	Field Description
Fail-Open Statistics	Numbers of created and invalidated sessions.
Fail-Close Statistics	Number of dropped packets.

Sample Output

show security group-vpn member ipsec statistics

```
user@host> show security group-vpn member ipsec statistics
ESP Statistics:
  Encrypted bytes:            54712
  Decrypted bytes:            16800
  Encrypted packets:            381
  Decrypted packets:            200
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
D3P Statistics:
  Old timestamp packets:                  0
  New timestamp packets:                  0
  No timestamp packets:                   0
  Unexpected D3P header packets:          0
  Invalid type packets:                   0
  Invalid length packets:                 0
  Invalid next header packets:            0
Exclude Statistics:
  Created sessions:                       0
  Invalidated sessions:                   0
Dynamic Policy Statistics:
  Created sessions:                     381
  Invalidated sessions:                   0
Fail-Open Statistics:
  Created sessions:                       0
  Invalidated sessions:                   0
Fail-Close Statistics:
  Dropped packets:                        0
```


show security group-vpn member kek security-associations

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn member kek security-associations [brief | detail | display xml] [index *sa-index*] [*peer-ipaddress*]

Release Information Command introduced in Junos OS Release 10.2.

Description Display Group VPNv2 security associations (SAs) for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

NOTE: Group VPNv2 is the name of the Group VPN technology on MX5, MX10, MX40, MX80, MX240, MX480, and MX960 routers. Group VPNv2 is different from the Group VPN technology implemented on SRX Security Gateways.

For more information about Group VPN on SRX Security Gateway devices, see "Group VPNv2 Overview" on page 665.

Options

- none—Display information about all Group VPNv2 SAs for the group member.
- **brief**—(Optional) Display summary output.
- **detail**—(Optional) Display detailed output.
- **display xml**—(Optional) Display xml.
- **index** *sa-index*—(Optional) Display detailed information about the specified SA identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.
- *peer-ipaddress*—(Optional) Display information about the SA with the specified peer.

Required Privilege Level view

Related Documentation

- Group VPNv2 Overview on page 665

List of Sample Output

show security group-vpn member kek security-associations on page 1192
show security group-vpn member kek security-associations detail on page 1193
show security group-vpn member kek security-associations detail | display xml on page 1193

Output Fields Table 99 on page 1191 lists the output fields for the **show security group-vpn member kek security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 99: show security group-vpn member kek security-associations

Field Name	Field Description
Index	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
Remote Address	IP address of the destination peer with which the local peer communicates.
State	State of the KEK security associations: • **DOWN**—SA is not active. • **UP**—SA is active.
Initiator cookie	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
Responder cookie	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received.
SPI	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI.
GroupID	Group identifier.
KEK Peer	IP address of the destination peer with which the local peer communicates.
Role	For the member, it is always responder.
State	State of the KEK security associations, which is always up.
Authentication method	RSA is the supported authentication method.
Local	Address of the local peer.
Remote	Address of the remote peer.
Lifetime	Number of seconds remaining until the IKE SA expires.

Table 99: show security group-vpn member kek security-associations *(continued)*

Field Name	Field Description

Algorithms	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process:
	• **Sig-hash**—Type of authentication algorithm used.
	  ◦ **sha-256**–Secure Hash Algorithm 256 (sha-256) authentication.
	  ◦ **sha-384**–Secure Hash Algorithm 394 (sha-384) authentication.
	• **Sig key length (bits)**—Size of signature key in bits.
	• **Encryption**—Type of encryption algorithm used.
	  ◦ **aes-256-cbc**—Advanced Encryption Standard (AES) 256-bit encryption.
	  ◦ **aes-192-cbc**— AES192-bit encryption
	  ◦ **aes-128-cbc**—AES 128-bit encryption.
	  ◦ **3des-cbc**—3 Data Encryption Standard (DES) encryption.
	  ◦ **des-cbc**—DES encryption.

Traffic statistics	• **Input bytes—**Number of bytes received. • **Output bytes—**Number of bytes transmitted. • **Input packets—**Number of packets received. • **Output packets—**Number of packets transmitted.
Server Info Version	Identify the latest set of information maintained in the server.
Server Heartbeat Interval	Interval in seconds at which the server sends heartbeats to group members.
Member Heartbeat Threshold	The heartbeat threshold configured on the group member for the IPsec VPN. If this number of heartbeats is missed on the member, the member reregisters with the server.
Heartbeat Timeout Left	Number of heartbeats until the heartbeat threshold is reached, at which time the member reregisters with the server. NOTE: When this number reaches 0, reregistration happens within 60 seconds.
Server Activation Delay	Number of seconds before a group member can use a new key when the member reregisters with the server.
Server Multicast Group	Multicast IP address to which the server sends rekey messages.
Server Replay Window	Antireplay time window value in milliseconds. 0 means antireplay is disabled.
Group Key Push sequence number	Sequence number of the KEK SA groupkey-push message. This number is incremented with every groupkey-push message.

Sample Output

show security group-vpn member kek security-associations

`user@host>` **show security group-vpn member kek security-associations**

```
Index    Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5824843 192.168.2.53      166       46871e26227f08f3  f0a463a4d5c3737b  1
```

Sample Output

show security group-vpn member kek security-associations detail

```
user@host> show security group-vpn member kek security-associations detail
  Index 5824843, Group Id: 1
  Group VPN Name: group1_2
  Local Gateway: 192.168.2.170, GDOI Server: 192.168.2.53
  Initiator cookie: 46871e26227f08f3, Responder cookie: f0a463a4d5c3737b
  Lifetime: Expires in 155 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-md5-96
   Encryption            : 3des-cbc
  Traffic statistics:
   Input  bytes  :                  0
   Output bytes  :                  0
   Input  packets:                  0
   Output packets:                  0
  Stats:
      Push received          :   0
      Delete received        :   0
```

show security group-vpn member kek security-associations detail | display xml

```
user@host> show security group-vpn member kek security-associations detail | display xml

<rpc-reply xmlns:junos="http://xml.example.net/junos/15.1/junos">
    <gvpn-kek-security-associations-information junos:style="detail">
        <kek-security-associations-block>
            <security-association-index>2987691</security-association-index>
            <group-id>400</group-id>
            <group-vpn-name>gvpn400</group-vpn-name>
            <local-address>192.168.1.100</local-address>
            <server-address>192.168.1.1</server-address>
            <initiator-cookie>510f854307a03675</initiator-cookie>
            <responder-cookie>690e5f121fba6de7</responder-cookie>
            <lifetime-remaining>Expires in 23729 seconds</lifetime-remaining>
            <push-sequence-number>364</push-sequence-number>
            <ike-security-associations>
                <ike-sa-algorithms>

<ike-sa-authentication-algorithm>hmac-sha1-96</ike-sa-authentication-algorithm>
                    <ike-sa-sig-key-length>2048</ike-sa-sig-key-length>

<ike-sa-encryption-algorithm>aes128-cbc</ike-sa-encryption-algorithm>
                </ike-sa-algorithms>
                <ike-sa-traffic-statistics>
                    <ike-sa-input-bytes>3012</ike-sa-input-bytes>
                    <ike-sa-output-bytes>252</ike-sa-output-bytes>
                    <ike-sa-input-packets>3</ike-sa-input-packets>
                    <ike-sa-output-packets>3</ike-sa-output-packets>
                </ike-sa-traffic-statistics>
            </ike-security-associations>
            <gvpn-kek-security-association-statistics>
                <kek-security-association-statistics>    Push received
```



```
  :   3</kek-security-association-statistics>
                <kek-security-association-statistics>    Delete received
  :   0</kek-security-association-statistics>
            </gvpn-kek-security-association-statistics>
        </kek-security-associations-block>
    </gvpn-kek-security-associations-information>
    <cli>
        <banner></banner>
    </cli>
</rpc-reply>
```


show security group-vpn member policy

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn member policy <vpn *vpn-name*> <group-id *group-id*>

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Show Group VPN policies. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Display information for all groups.

vpn ***vpn-name***—(Optional) Display policy information for the specified group name.

group-id ***group-id***—(Optional) Display policy information for the specified group identifier.

Required Privilege Level view

Related Documentation

- Group VPNv2 Overview on page 665

List of Sample Output show security group-vpn member policy on page 1196

Output Fields Table 100 on page 1195 lists the output fields for the **show security group-vpn member policy** command. Output fields are listed in the approximate order in which they appear.

Table 100: show security group-vpn member policy Output Fields

Field Name	Field Description
Group VPN Name	Group name.
Group Id	Group identifier.
From-zone	From zone configured for the policy.
To-zone	To zone configured for the policy.
Tunnel-id	Tunnel identifier.
Policy type	Secure, fail-open, fail-close, or exclude.
Source	IP address, port, and protocol of the source traffic.
Destination	IP address, port, and protocol of the destination traffic.

Sample Output

show security group-vpn member policy

```
user@host> show security group-vpn member policy
Group VPN Name: group1000, Group Id: 1000
From-zone: trust_1, To-zone: untrust
  Tunnel-id: 63490, Policy type: Exclude
    Source      : IP <192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol
 <17>
    Destination : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <17>


  Tunnel-id: 49153, Policy type: Secure
    Source      : IP 192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol
<0>
    Destination : IP <192.0.2.0 - 192.0.2.255>, Port <0 - 65535>, Protocol <0>

  Tunnel-id: 49152, Policy type: Secure
    Source      : IP <192.0.2.0 - 192.0.2.255>, Port <0 - 65535>, Protocol <1>
    Destination : IP <192.0.2.0 - 192.0.2.255>, Port <0 - 65535>, Protocol <1>

  Tunnel-id: 63491, Policy type: Fail-open (Inactivated)
    Source      : IP 192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol
<17>
    Destination : IP <192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol
 <17>

  Tunnel-id: 63489, 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>
```


show security group-vpn server ike security-associations

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn server ike security-associations [brief | detail] [group *group-name* | group-id *group-id*] [index *sa-index*]

Release Information Command introduced in Junos OS Release 10.2.

Description Display IKE security associations (SAs). Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Display all IKE SAs for all groups.
- brief—(Optional) Display summary output.
- **detail**—(Optional) Display detailed level of output.
- **group**—(Optional) Display IKE SAs for the specified group.
- **group-id**—(Optional) Display IKE SAs for the specified group.

NOTE: An IKE SA can be used by a group member to register to multiple groups. When you specify the **group** or **group-id** options to list the IKE SAs for a specified group, all existing IKE SAs that could be used to register to the group are displayed.

- **index**—(Optional) Display information for a particular SA based on the index number of the SA. To obtain the index number for a particular SA, display the list of existing SAs by using the command with no options.

Required Privilege Level view

Related Documentation

- show security group-vpn member ike security-associations on page 1177
- Group VPNv2 Overview on page 665

List of Sample Output

show security group-vpn server ike security-associations on page 1199
show security group-vpn server ike security-associations detail on page 1200

Output Fields Table 101 on page 1198 lists the output fields for the **show security group-vpn server ike security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 101: show security group-vpn server ike security-associations Output Fields

Field Name	Field Description
Index	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
Remote Address	IP address of the destination peer with which the local peer communicates.
State	State of the IKE security associations: • **DOWN**—SA has not been negotiated with the peer. • **UP**—SA has been negotiated with the peer.
Initiator cookie	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
Responder cookie	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received. A cookie is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine the cookie's authenticity.
Mode	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • **main**—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • **aggressive**—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
IKE Peer	IP address of the destination peer with which the local peer communicates.
Exchange type	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • **main**—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • **aggressive**—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
Authentication method	Method the server uses to authenticate the source of IKE messages: • **pre-shared-keys**—Preshared key for encryption and decryption that both participants must have before beginning tunnel negotiations. **rsa-signatures**—Digital signature, a certificate that confirms the identity of the certificate holder.
Local	Address of the local peer.

Table 101: show security group-vpn server ike security-associations Output Fields *(continued)*

Field Name	Field Description
Remote	Address of the remote peer.
Lifetime	Number of seconds remaining until the IKE SA expires.

Algorithms	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process:
	• **Authentication**—Type of authentication algorithm used.
	  ◦ **sha-256**—Secure Hash Algorithm 256 authentication.
	  ◦ **sha-384**—Secure Hash Algorithm 384 authentication..
	• **Encryption**—Type of encryption algorithm used.
	  ◦ **aes-256-cbc**—Advanced Encryption Standard (AES) 256-bit encryption.
	  ◦ **aes-192-cbc**— AES192-bit encryption
	  ◦ **aes-128-cbc**—AES 128-bit encryption.

Traffic statistics	• **Input bytes—**Number of bytes received. • **Output bytes—**Number of bytes transmitted. • **Input packets—**Number of packets received. • **Output packets—**Number of packets transmitted.
IPSec security associations	• ***number* created:** The number of SAs created. • ***number* deleted**: The number of SAs deleted.

Phase 2 negotiations in progress	Number of Phase 2 IKE negotiations in progress and status information:
	• **Negotiation type**—Type of Phase 2 negotiation. Junos OS currently supports quick mode.
	• **Message ID**—Unique identifier for a Phase 2 negotiation.
	• **Local identity**—Identity of the local Phase 2 negotiation. The format is id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation)
	• **Remote identity**—Identity of the remote Phase 2 negotiation. The format is id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation)
	• **Flags**—Notification to the key management process of the status of the IKE negotiation:
	  ◦ **caller notification sent**—Caller program notified about the completion of the IKE negotiation.
	  ◦ **waiting for done**—Negotiation is done. The library is waiting for the remote end retransmission timers to expire.
	  ◦ **waiting for remove**—Negotiation has failed. The library is waiting for the remote end retransmission timers to expire before removing this negotiation.
	  ◦ **waiting for policy manager**—Negotiation is waiting for a response from the policy manager.

Sample Output

show security group-vpn server ike security-associations

`user@host>` **show security group-vpn server ike security-associations**

```
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address

738879  UP     0fa7c5fdcb74669f  8c21f5d1b533010c  Aggressive     192.168.1.120
```

Sample Output

show security group-vpn server ike security-associations detail

```
user@host> show security group-vpn server ike security-associations detail
IKE peer 192.168.1.120, Index 738879, Gateway Name: gvpn
  Role: Responder, State: UP
  Initiator cookie: 0fa7c5fdcb74669f, Responder cookie: 8c21f5d1b533010c
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 192.168.1.50:848, Remote: 192.168.1.120:848
  Lifetime: Expires in 3541 seconds
  Peer ike-id: test
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha-256-128
   Encryption            : aes-256-cbc
   Pseudo random function: hmac-sha-256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                  600
   Output bytes  :                  932
   Input  packets:                    4
   Output packets:                    3
  Flags: IKE SA is created
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 0

Flags: IKE SA is created
```

show security group-vpn server ipsec security-associations

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn server ipsec security-associations [brief | detail] [group *group-name* | group-id *group-id*]

Release Information Command introduced in Junos OS Release 10.2.

Description Display IPsec security associations (SAs). Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Display all IPsec SAs for all groups.
- **brief**—(Optional) Display summary output.
- **detail**—(Optional) Display detailed level of output.
- **group**—(Optional) Display IPsec SAs for the specified group.
- **group-id**—(Optional) Display IPsec SAs for the specified group.

Required Privilege Level view

Related Documentation

- show security group-vpn member ipsec security-associations on page 1184
- Group VPNv2 Overview on page 665

List of Sample Output

show security group-vpn server ipsec security-associations on page 1202
show security group-vpn server ipsec security-associations detail on page 1202

Output Fields Table 102 on page 1201 lists the output fields for the **show security group-vpn server ipsec security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 102: show security group-vpn server ipsec security-associations

Field Name	Field Description
Group	Group name.
Group ID	Group identifier.
Total IPsec SAs	The total number of IPsec SAs for each group is shown.
IPsec SA	Name of the SA.
Protocol	Protocol supported. Transport mode supports Encapsulation Security Protocol (ESP).

Table 102: show security group-vpn server ipsec security-associations *(continued)*

Field Name	Field Description
Algorithm	Cryptography used to secure exchanges between peers during the IKE Phase 2 negotiations includes • An authentication algorithm used to authenticate exchanges between the peers. Options are **sha-256** and **sha-384**. • An encryption algorithm used to encrypt data traffic. Options are **aes-128-cbc**, **aes-192-cbc**, or **aes-256-cbc**.
SPI	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI.
Lifetime	The lifetime of the SA, after which it expires, expressed in seconds.
Policy Name	Group policy associated with the IPsec SA. The source address, destination address, source port, destination port, and protocol defined for the policy are displayed.

Sample Output

show security group-vpn server ipsec security-associations

```
user@host> show security group-vpn server ipsec security-associations
  Group: group200, Group Id: 200
    Total IPsec SAs: 1
    IPsec SA          Algorithm             SPI              Lifetime
    sa1               ESP:aes-256/sha-256    55837dfe         17
    sa1               ESP:aes-256/sha1-256   760088d          137
```

Sample Output

show security group-vpn server ipsec security-associations detail

```
user@host> show security group-vpn server ipsec security-associations detail
Group: group1, Group Id: 1
Total IPsec SAs: 10
  IPsec SA: sa1
    Protocol: ESP, Authentication: sha-256, Encryption: aes-256
    Anti-replay: D3P enabled, window size 10 milliseconds
    SPI: e68c9525
    Lifetime: Expires in 66 seconds, Activated
    Policy Name: pol1
      Source: 192.168.1.0/24
      Destination: 192.168.1.0/24
      Source Port: 0
      Destination Port: 0
      Protocol: 0
  IPsec SA: sa1
    Protocol: ESP, Authentication: sha-256, Encryption: aes-256
    Anti-replay: D3P enabled, window size 10 milliseconds
    SPI: 7ee14902
    Lifetime: Expires in 276 seconds, Activated in 36 seconds
    Rekey in 186 seconds
    Policy Name: pol1
      Source: 192.168.1.0/24
      Destination: 192.168.1.0/24
```

```
Source Port: 0
Destination Port: 0
Protocol: 0
```


show security group-vpn server kek security-associations

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn server kek security-associations [brief | detail] [group *group-name* | group-id *group-id* | index *sa-index*]

Release Information Command introduced in Junos OS Release 10.2.

Description Display configured server-member communications. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Display server-member communications configured for all groups.
- **brief**—(Optional) Display summary output.
- **detail**—(Optional) Display detailed output.
- **group**—(Optional) Display server-member communications configured for the specified group.
- **group-id**—(Optional) Display server-member communications configured for the specified group.
- **index**—(Optional) Display information for a particular SA based on the index number of the SA. To obtain the index number for a particular SA, display the list of existing SAs by using the command with no options.

Required Privilege Level view

Related Documentation

- show security group-vpn member kek security-associations on page 1190
- Group VPNv2 Overview on page 665

List of Sample Output

show security group-vpn server kek security-associations on page 1206
show security group-vpn server kek security-associations detail on page 1206

Output Fields Table 103 on page 1204 lists the output fields for the **show security group-vpn server kek security-assocations** command. Output fields are listed in the approximate order in which they appear.

Table 103: show security group-vpn server kek security-associations Output Fields

Field Name	Field Description
Index	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.

Table 103: show security group-vpn server kek security-associations Output Fields *(continued)*

Field Name	Field Description
Remote Address	Identifier of the remote/peer. Because there could be multiple members, the remote address always contains the IP address 0.0.0.0.
State	State of the KEK security associations: • **DOWN**—SA is not active. • **UP**—SA is active.
Initiator cookie	Random number generated by the server. This is used when the server needs to push data to a member, or a member needs to reply to the server.
Responder cookie	Random number generated by the server. This is used when the server needs to push data to a member, or a member needs to reply to the server.
GroupId	Group identifier.
KEK Peer	IP address of the destination peer with which the local peer communicates. For KEK SAs, it always contains 0.0.0.0 which means any IP address.
Role	For the server, it is always initiator.
Authentication method	RSA is the supported authentication method.
Local	Address of the local peer.
Remote	Address of the remote peer.
Lifetime	Number of seconds remaining until the IKE SA expires.

Algorithms	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the Phase 2 process:
	• **Sig-hash**—Type of authentication algorithm used.
	  ◦ **sha-256**—Secure Hash Algorithm 256 authentication.
	  ◦ **sha-384**—Secure Hash Algorithm 384 authentication.
	• **Encryption**—Type of encryption algorithm used.
	  ◦ **aes-256-cbc**—Advanced Encryption Standard (AES) 256-bit encryption.
	  ◦ **aes-192-cbc**— AES192-bit encryption
	  ◦ **aes-128-cbc**—AES 128-bit encryption.

Traffic statistics	• **Input bytes—**Number of bytes received. • **Output bytes—**Number of bytes transmitted. • **Input packets—**Number of packets received. • **Output packets—**Number of packets transmitted.
Server Info Version	Identify the latest set of information maintained in the server.
The following fields are the configured **server-member-communication** options:	

Table 103: show security group-vpn server kek security-associations Output Fields *(continued)*

Field Name	Field Description
Server Replay Window	Antireplay time in milliseconds. This is 0 if antireplay is disabled.
Retransmission Period	Number of seconds between a rekey transmission and the first retransmission when there is no reply from the member.
Number of Retransmissions	For unicast communications, the number of times the server retransmits rekey messages to a member when there is no reply.
Lifetime Seconds	Configured lifetime, in seconds, for the KEK.
Group Key Push sequence number	Sequence number of the KEK SA groupkey-push message. This number is incremented with every groupkey-push message.

Sample Output

show security group-vpn server kek security-associations

```
user@host> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
  739031  18995     7e17278bf0a65975  0616de443d1beb77  200
```

Sample Output

show security group-vpn server kek security-associations detail

```
user@host> 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
```

show security group-vpn server registered-members

Supported Platforms SRX Series, vSRX

Syntax show security group-vpn server registered-members <group *group-name*> <group-id *group-id*> <detail>

Release Information Command introduced in Junos OS Release 10.2.

Description Display currently registered group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options

- none—Display all group members for all groups.
- brief—(Optional) Display summary output.
- detail—(Optional) Display detailed output.
- **group**—(Optional) Display group members for the specified group.
- **group-id**—(Optional) Display group members for the specified group.

Required Privilege Level view

Related Documentation

- clear security group-vpn server on page 1133
- Group VPNv2 Overview on page 665

List of Sample Output show security group-vpn server registered-members on page 1208
show security group-vpn server registered-members detail on page 1208

Output Fields Table 104 on page 1207 lists the output fields for the **show security group-vpn server registered-members**command. Output fields are listed in the approximate order in which they appear.

Table 104: show security group—vpn server registered-members Output Fields

Field Name	Field Description
Group	Group name.
Group Id	Group identifier.
Member Gateway	IP address of the gateway for the group member.
Member IP	IP address of the group member.
Last Update	The last time that members registered or sent acknowledgements to the server.

Table 104: show security group—vpn server registered-members Output Fields *(continued)*

Field Name	Field Description
Vsys	The root system.

Sample Output

show security group-vpn server registered-members

```
user@host> show security group-vpn server registered-members
 Group: group200, Group Id: 200
    Total number of registered members: 1
    Member Gateway                   Member IP        Last Update
Vsys
    gvpn_simpleman                   192.168.1.100    Fri Dec 20 2013 07:27:33
root
```

Sample Output

show security group-vpn server registered-members detail

```
user@host> show security group-vpn server registered-members detail
Group: group1, Group Id: 1
      Total number of registered members: 1

  Member gateway: gateway_group1_1, Member IP: 192.168.1.2, Vsys: root
  Last Update: Fri May 16 2014 03:37:17
  Stats:
      Pull Succeeded                : 321
      Pull Failed                   : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

show security group-vpn server server-cluster

Supported Platforms SRX Series, vSRX

Syntax **show security group-vpn server server-cluster <brief> <detail> <group *group-name*> <group-id *group-id*> <peer-gateway *gateway-name*>**

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Show information about servers in the Group VPNv2 server cluster. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Display Group VPNv2 server cluster information for all groups.

brief—(Optional) Display summary output.

detail—(Optional) Display detailed output, including information about exchanges with peer servers in the cluster.

group *group-name*—(Optional) Display Group VPNv2 server cluster information for the specified group name.

group-id *group-id*—(Optional) Display Group VPNv2 server cluster information for the specified group identifier.

peer-gateway *gateway-name*—(Optional) Display Group VPNv2 server cluster information for the specified peer.

Required Privilege Level view

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Server Clusters on page 714

List of Sample Output

show security group-vpn server server-cluster on page 1210
show security group-vpn server server-cluster detail on page 1210

Output Fields Table 105 on page 1209 lists the output fields for the **show security group-vpn server server-cluster** command. Output fields are listed in the approximate order in which they appear.

Table 105: show security group-vpn server server-cluster Output Fields

Field Name	Field Description
Group	Group name.

Table 105: show security group-vpn server server-cluster Output Fields *(continued)*

Field Name	Field Description
Group Id	Group identifier.
Role	Role of this server in the Group VPNv2 server cluster.
Version Number	32-bit version number included in **cluster-update** exchanges and DPD probes to support anti-replay. The first **cluster-update** message sent from the root-server has version number 1. Subsequent **cluster-update** messages increment the version number by one. (Retransmit messages do not increment the version number.) Upon receipt of a **cluster-update** message, the sub-server validates the received version number. The received version number must be greater than the version number in the last received message, otherwise the message is discarded. The sub-server responds to a **cluster-update** message with an ACK message that contains the same version number as the received message. Upon receipt of the ACK message, the root-server checks that the version number is the same as in the message it sent. If the version number is valid, the exchange is considered successful. If the version number is not valid, the original message is retransmitted or the exchange is considered failed.
Peer Gateway	Name of the peer server in the Group VPNv2 server cluster.
Peer IP	IP address of the remote peer server in the Group VPNv2 server cluster.
Role	Role of the peer server in the Group VPNv2 server cluster.
Status	Status of the peer server in the Group VPNv2 server cluster.

Sample Output

show security group-vpn server server-cluster

```
user@host> show security group-vpn server server-cluster
Group: group200, Group Id: 200
Role: Root-server, Version Number: 1,
  Peer Gateway                     Peer IP          Role
Status
  sub_server1                      192.168.1.112    Sub-server
Active
  sub_server2                      192.168.1.113    Sub-server
Active
```

show security group-vpn server server-cluster detail

```
user@host> show security group-vpn server server-cluster detail
GGroup: group200, Group Id: 200
Role: Root-server, Version Number: 1,

Peer gateway: sub_server1,
  Peer IP: 192.168.1.112, Local IP: 192.168.1.111, VR: vr1,
  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:              1
  CLUSTER-UPDATE recv:              0
  CLUSTER-UPDATE success:           1
  CLUSTER-UPDATE fail:              0
  CLUSTER-UPDATE abort:             0
  CLUSTER-UPDATE timeout:           0
  CLUSTER-UPDATE pending:           0
  CLUSTER-UPDATE max retry reached: 0
  DPD send:                         5
  DPD send fail:                    0
  DPD ACK recv:                     5
  DPD ACK invalid seqno:            0
  IPsec SA policy mismatch:         0
  IPsec SA proposal mismatch:       0
  KEK SA proposal mismatch:         0

Peer gateway: sub_server2,
  Peer IP: 192.168.1.113, Local IP: 192.168.1.111, 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:              1
  CLUSTER-UPDATE recv:              0
  CLUSTER-UPDATE success:           1
  CLUSTER-UPDATE fail:              0
  CLUSTER-UPDATE abort:             0
  CLUSTER-UPDATE timeout:           0
  CLUSTER-UPDATE pending:           0
  CLUSTER-UPDATE max retry reached: 0
  DPD send:                         6
  DPD send fail:                    0
  DPD ACK recv:                     6
  DPD ACK invalid seqno:            0
  IPsec SA policy mismatch:         0
  IPsec SA proposal mismatch:       0
  KEK SA proposal mismatch:         0
```

show security group-vpn server statistics

Supported Platforms SRX Series, vSRX

Syntax **show security group-vpn server statistics <group *group-name*> <group-id *group-id*>**

Release Information Command introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Show Group VPNv2 server statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, and SRX4200 devices and vSRX instances.

Options **none**—Display Group VPNv2 server statistics for all groups.

group *group-name*—(Optional) Display Group VPNv2 server statistics for the specified group name.

group-id *group-id*—(Optional) Display Group VPNv2 server statistics for the specified group identifier.

Required Privilege Level view

Related Documentation

- Group VPNv2 Overview on page 665
- Understanding Group VPNv2 Server Clusters on page 714

List of Sample Output show security group-vpn server statistics on page 1212

Output Fields Table 106 on page 1212 lists the output fields for the **show security group-vpn server statistics** command. Output fields are listed in the approximate order in which they appear.

Table 106: show security group-vpn server statistics Output Fields

Field Name	Field Description
Group	Group name.
Group Id	Group identifier.
Stats	Server events and number of occurrences.

Sample Output

show security group-vpn server statistics

```
user@host> show security group-vpn server statistics
```

```
Group: group1, Group Id: 1
  Stats:
      Pull Succeeded               : 321
      Pull Failed                  : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                    : 0
      Push Acknowledged            : 0
      Push Unacknowledged          : 0
```

show security ike active-peer

Supported Platforms SRX Series, vSRX

Syntax show security ike active-peer

Release Information Command introduced in Junos OS Release 10.4. Support to display dead peer detection (DPD) statistics added in Junos OS Release 12.3X48-D10.

Description Display the list of connected active users with details about the peer addresses and ports they are using.

Required Privilege Level view

Related Documentation

- show security ike security-associations on page 1217
- show security ipsec security-associations on page 1236

List of Sample Output

show security ike active-peer on page 1214
show security ike active-peer detail on page 1214

Sample Output

show security ike active-peer

```
user@host> show security ike active-peer

Remote Address     Port      Peer IKE-ID        AAA username      Assigned IP
192.168.6.136      8034      user1tac@650a      user1             192.168.80.225
```

show security ike active-peer detail

```
user@host> show security ike active-peer detail
Peer address: 192.168.0.6, Port: 500,
Peer IKE-ID: C=US, ST=California, L=Sunnyvale, O=example, OU=engineering,
CN=SPOKE9061
AAA username: not available
Assigned network attributes:
IP Address: 0.0.0.0 ,    netmask          : 0.0.0.0
DNS Address      : 0.0.0.0 ,    DNS2 Address     : 0.0.0.0
WINS Address     : 0.0.0.0 ,    WINS2 Address    : 0.0.0.0

Previous Peer address    : 0.0.0.0, Port               : 0
Active IKE SA indexes    : 75203629
IKE SA negotiated        : 1
IPSec tunnels active     : 1, IPSec Tunnel IDs    : 68157442

DPD Config Info      :  Mode: always-send    Interval: 60   Threshold: 5
p1sa_index:75203629
DPD Statistics       : DPD-flags: REMOTE_ACCESS
DPD Statistics       : DPD TTL                  :     0       DPD seq-no
```

```
          :       0
DPD Statistics      : DPD Req Sent              :     0       DPD Resp Rcvd
          :       0
```

show security ike pre-shared-key

Supported Platforms SRX Series, vSRX

Syntax

show security ike pre-shared key
<master-key *master-key* >
<user-id *user-id* >

Release Information Command introduced in Junos OS Release 8.5.

Description Display the Internet Key Exchange (IKE) preshared key used by the Virtual Private network (VPN) gateway to authenticate the remote access user.

Options

- **master-key *master-key*** —(Optional) Master preshared key.
- **user-id *user-id*** —(Optional) IKE user ID value.

Required Privilege Level view

Related Documentation

- pre-shared-key (Security IKE Policy) on page 1053

List of Sample Output show security ike pre-shared-key on page 1216

Sample Output

show security ike pre-shared-key

```
user@host> show security ike pre-shared-key user-id a@example.net master-key example
Preshared Key:3b33ec3631a561ec5a710f5d02f208033b108bb4
```


show security ike security-associations

Supported Platforms SRX Series, vSRX

Syntax

show security ike security-associations
peer-address
brief | detail
family (inet | inet6)
fpc *slot-number*
index *SA-index-number*
kmd-instance (all | *kmd-instance-name*)
pic *slot-number*
sa-type shortcut <detail>

Release Information Command introduced in Junos OS Release 8.5 . Support for the **fpc**, **pic**, and **kmd-instance** options added in Junos OS Release 9.3. Support for the **family** option added in Junos OS Release 11.1. Support for Auto Discovery VPN added in Junos OS Release 12.3X48-D10. Support for IKEv2 reauthentication added in Junos OS Release 15.1X49-D60. Support for IKEv2 fragmentation added in Junos OS Release 15.1X49-D80.

Description Display information about Internet Key Exchange security associations (IKE SAs).

Options

- none—Display standard information about existing IKE SAs, including index numbers.
- ***peer-address***—(Optional) Display details about a particular SA based on the IPv4 or IPv6 address of the destination peer. This option and **index** provide the same level of output.
- **brief**—(Optional) Display standard information about all existing IKE SAs. (Default)
- **detail**—(Optional) Display detailed information about all existing IKE SAs.
- **family**—(Optional) Display IKE SAs by family. This option is used to filter the output.
  - **inet**—IPv4 address family.
  - **inet6**—IPv6 address family.
- **fpc *slot-number***—(Optional) Display information about existing IKE SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.
- **index *SA-index-number***—(Optional) Display information for a particular SA based on the index number of the SA. For a particular SA, display the list of existing SAs by using the command with no options. This option and ***peer-address*** provide the same level of output.
- **kmd-instance** —(Optional) Display information about existing IKE SAs in the key management process (in this case, it is KMD) identified by FPC *slot-number* and PIC *slot-number*. This option is used to filter the output.
  - **all**—All KMD instances running on the Services Processing Unit (SPU).

- *kmd-instance-name*—Name of the KMD instance running on the SPU.

- pic *slot-number* —(Optional) Display information about existing IKE SAs in this PIC slot. This option is used to filter the output.
- sa-type—(Optional for ADVPN) Type of SA. **shortcut** is the only option for this release.

Required Privilege Level view

Related Documentation

- *Example: Configuring a Route-Based VPN Tunnel in a User Logical System*

List of Sample Output

Output Fields Table 107 on page 1218 lists the output fields for the **show security ike security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 107: show security ike security-associations Output Fields

Field Name	Field Description
IKE Peer or Remote Address	IP address of the destination peer with which the local peer communicates.
Index	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
Gateway Name	Name of the IKE gateway.

Table 107: show security ike security-associations Output Fields *(continued)*

Field Name	Field Description
Location	• **FPC**—Flexible PIC Concentrator (FPC) slot number. • **PIC**—PIC slot number. • **KMD-Instance**—The name of the KMD instance running on the SPU, identified by FPC *slot-number* and PIC *slot-number*. Currently, 4 KMD instances are running on each SPU, and any particular IKE negotiation is carried out by a single KMD instance.
Role	Part played in the IKE session. The device triggering the IKE negotiation is the initiator, and the device accepting the first IKE exchange packets is the responder.
State	State of the IKE SAs: • **DOWN**—SA has not been negotiated with the peer. • **UP**—SA has been negotiated with the peer.
Initiator cookie	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
Responder cookie	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received. A cookie is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine the cookie's authenticity.
Exchange type	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between one another. Each exchange type or mode determines the number of messages and the payload types that are contained in each message. The modes are: • **main**—The exchange is done with six messages. This mode encrypts the payload, protecting the identity of the neighbor. • **aggressive**—The exchange is done with three messages. This mode does not encrypt the payload, leaving the identity of the neighbor unprotected. NOTE: IKEv2 protocol does not use the mode configuration for negotiation. Therefore, the mode displays the version number of the security association.
Authentication method	Method used to authenticate the source of IKE messages, which can be either preshared keys or digital certificates (**DSA-signatures**, **ECDSA-signatures-256**, **ECDSA-signatures-384**, or **RSA-signatures**).
Local	Address of the local peer.
Remote	Address of the remote peer.
Lifetime	Number of seconds remaining until the IKE SA expires.
Reauth Lifetime	When enabled, number of seconds remaining until reauthentication triggers a new IKEv2 SA negotiation.

Table 107: show security ike security-associations Output Fields *(continued)*

Field Name	Field Description
IKE Fragmentation	**Enabled** means that both the IKEv2 initiator and responder support message fragmentation and have negotiated the support during the IKE_SA_INIT message exchange. **Size** shows the maximum size of an IKEv2 message before it is fragmented.

Algorithms	IKE algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process:
	
	• **Authentication**—Type of authentication algorithm used:
	  ◦ **sha1**—Secure Hash Algorithm 1 authentication.
	  ◦ **md5**—MD5 authentication.
	• **Encryption**—Type of encryption algorithm used:
	  ◦ **aes-256-cbc**—Advanced Encryption Standard (AES) 256-bit encryption.
	  ◦ **aes-192-cbc**— AES192-bit encryption.
	  ◦ **aes-128-cbc**—AES 128-bit encryption.
	  ◦ **3des-cbc**—3 Data Encryption Standard (DES) encryption.
	  ◦ **des-cbc**—DES encryption.

Diffie-Hellman group	Specifies the IKE Diffie-Hellman group.
Traffic statistics	• **Input bytes—**Number of bytes received. • **Output bytes—**Number of bytes transmitted. • **Input packets—**Number of packets received. • **Output packets—**Number of packets transmitted. • **Input fragmented packets**—Number of IKEv2 fragmented packets received. • **Output fragmented packets**—Number of IKEv2 fragmented packets transmitted.
Flags	Notification to the key management process of the status of the IKE negotiation: • **caller notification sent**—Caller program notified about the completion of the IKE negotiation. • **waiting for done**—Negotiation is done. The library is waiting for the remote end retransmission timers to expire. • **waiting for remove**—Negotiation has failed. The library is waiting for the remote end retransmission timers to expire before removing this negotiation. • **waiting for policy manager**—Negotiation is waiting for a response from the policy manager.
IPSec security associations	• ***number* created**: The number of SAs created. • ***number* deleted**: The number of SAs deleted.

Table 107: show security ike security-associations Output Fields *(continued)*

Field Name	Field Description

Phase 2 negotiations in progress	Number of Phase 2 IKE negotiations in progress and status information:
	• **Negotiation type**—Type of Phase 2 negotiation. Junos OS currently supports quick mode.
	• **Message ID**—Unique identifier for a Phase 2 negotiation.
	• **Local identity**—Identity of the local Phase 2 negotiation. The format is *id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation)*.
	• **Remote identity**—Identity of the remote Phase 2 negotiation. The format is *id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation)*.
	• **Flags**—Notification to the key management process of the status of the IKE negotiation:
	  ◦ **caller notification sent**—Caller program notified about the completion of the IKE negotiation.
	  ◦ **waiting for done**—Negotiation is done. The library is waiting for the remote end retransmission timers to expire.
	  ◦ **waiting for remove**—Negotiation has failed. The library is waiting for the remote end retransmission timers to expire before removing this negotiation.
	  ◦ **waiting for policy manager**—Negotiation is waiting for a response from the policy manager.

Sample Output

show security ike security-associations (IPv4)

```
user@host> show security ike security-associations
Index  Remote Address  State  Initiator cookie        Responder cookie Mode
8  192.168.1.2  UP  3a895f8a9f620198 9040753e66d700bb Main
Index  Remote Address  State  fInitiator cookie Responder cookie Mode
9  192.168.1.3   UP  5ba96hfa9f65067  70890755b65b80b  Main
```

show security ike security-associations (IPv6)

```
user@host> show security ike security-associations
Index    State  Initiator cookie  Responder cookie  Mode          Remote Address

5       UP     e48efd6a444853cf  0d09c59aafb720be  Aggressive     2001:db8::1112
```

show security ike security-associations detail (SRX300, SRX320, SRX340, SRX345, and SRX550HM Devices)

```
user@host> show security ike security-associations detail
IKE peer 192.168.134.245, Index 2577565, Gateway Name: tropic
  Role: Initiator, State: UP
  Initiator cookie: b869b3424513340a, Responder cookie: 4cb3488cb19397c3
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 192.168.134.241:500, Remote: 192.168.134.245:500
  Lifetime: Expires in 169 seconds
  Peer ike-id: 192.168.134.245
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha1
  Diffie-Hellman group   : DH-group-5
```



```
  Traffic statistics:
   Input  bytes  :                 1012
   Output bytes  :                 1196
   Input  packets:                    4
   Output packets:                    5
  Flags: IKE SA is created
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 192.168.134.241:500, Remote: 192.168.134.245:500
    Local identity: 192.168.134.241
    Remote identity: 192.168.134.245
    Flags: IKE SA is created
```

show security ike security-associations detail (SRX5400, SRX5600, and SRX5800 Devices)

```
user@host> show security ike security-associations detail
IKE peer 192.168.2, Index 914039858, Gateway Name: tropic
  Location: FPC 3, PIC 1, KMD-Instance 3
  Role: Initiator, State: UP
  Initiator cookie: 219a697652bdde37, Responder cookie: b49c30b229d36bcd
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 192.168.1.1:500, Remote: 192.168.1.2:500
  Lifetime: Expires in 26297 seconds
  Peer ike-id: 192.168.1.2
  AAA user-name: not available
  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-5
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 1
```

show security ike security-associations family inet6

```
user@host> show security ike security-associations family inet6
  IKE peer 2001:db8:1212::1112, Index 5, Gateway Name: tropic
  Role: Initiator, State: UP
  Initiator cookie: e48efd6a444853cf, Responder cookie: 0d09c59aafb720be
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 2001:db8:1212::1111:500, Remote: 2001:db8:1212::1112:500
  Lifetime: Expires in 19518 seconds
  Peer ike-id: not valid
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  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:db8:1212::1111:500, Remote: 2001:db8:1212::1112:500
    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
```

show security ike security-associations index 8 detail

```
user@host> show security ike security-associations index 8 detail
IKE peer 192.168.1.2, Index 8, Gateway Name: tropic
 Role: Responder, State:UP
 Initiator cookie: 3a895f8a9f620198, Responder cookie: 9040753e66d700bb
 Exchange type; main, Authentication method: Pre-shared-keys
 Local: 192.168.1.1:500, Remote: 192.168.1.2:500
 Lifetime: Expired in 381 seconds
 Algorithms:
  Authentication:       md5
  Encryption:      3des-cbc
  Pseudo random function      hmac-md5
     Diffie-Hellman group  : DH-group-5
    Traffic statistics:
  Input bytes:       11268
  Output bytes:        6940
  Input packets:        57
  Output packets:        57
 Flags: Caller notification sent
 IPsec security associations: 0 created, 0 deleted
 Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Responder, Message ID: 1765792815
  Local: 192.168.1.1:500, Remote: 192.168.1.2:500
  Local identity: No Id
  Remote identity: No Id
  Flags: Caller notification sent, Waiting for remove
```

show security ike security-associations 192.168.1.2

```
user@host> show security ike security-associations 192.168.1.2
Index     State  Initiator cookie  Responder cookie  Mode Remote Address
   8        UP     3a895f8a9f620198  9040753e66d700bb  Main 192.168.1.2
```

show security ike security-associations fpc 6 pic 1 kmd-instance all (SRX Series Devices)

```
user@host> show security ike security-associations fpc 6 pic 1 kmd-instance all
Index      Remote Address  State  Initiator cookie  Responder cookie  Mode

1728053250 192.168.1.2     UP     fc959afd1070d10b  bdeb7e8c1ea99483  Main
```

show security ike security-associations detail (ADVPN Suggester, Static Tunnel)

```
user@host> show security ike security-associations detail
IKE peer 192.168.0.105, Index 13563297, Gateway Name: zth_hub_gw
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
```

```
 Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent            :  12
     Suggestion response accepted:  12
     Suggestion response declined:   0
  Role: Responder, State: UP
  Initiator cookie: 4d3f4e4b2e75d727, Responder cookie: 81ab914e13cecd21
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.0.154:500, Remote: 192.168.0.105:500
  Lifetime: Expires in 26429 seconds
  Peer ike-id: DC=example, CN=host02, L=Sunnyvale, ST=CA, C=US
```

show security ike security-associations detail (ADVPN Partner, Static Tunnel)

```
user@host> show security ike security-associations detail
IKE peer 192.168.0.154, Index 4980720, Gateway Name: zth_spoke_gw
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
 Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:  12
     Suggestions accepted:  12
     Suggestions declined:   0
  Role: Initiator, State: UP
  Initiator cookie: 4d3f4e4b2e75d727, Responder cookie: 81ab914e13cecd21
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.0.105:500, Remote: 192.168.0.154:500
  Lifetime: Expires in 26252 seconds
  Peer ike-id: DC=example, CN=host01, OU=SBU, O=example, L=Sunnyvale, ST=CA, C=US
```

show security ike security-associations detail (ADVPN Partner, Shortcut)

```
user@host> show security ike security-associations detail
IKE peer 192.168.0.106, Index 4980737, Gateway Name:
GW-ADVPN-GT-ADVPN-zth_spoke_vpn-268173323
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
   Type: Shortcut, Local Capability: Partner, Peer Capability: Partner
  Role: Responder, State: UP
  Initiator cookie: e1ed0c655929debc, Responder cookie: 437de6ed784ba63e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.0.105:500, Remote: 192.168.0.106:500
  Lifetime: Expires in 28796 seconds
  Peer ike-id: DC=example, CN=paulyd, L=Sunnyvale, ST=CA, C=US
```

show security ike security-associations sa-type shortcut (ADVPN)

```
user@host> show security ike security-associations sa-type shortcut
Index    State  Initiator cookie  Responder cookie  Mode          Remote Address

4980742 UP     vb56fbe694eaee5b6 064dbccbfa3b2aab  IKEv2          192.168.0.106
```

show security ike security-associations sa-type shortcut detail (ADVPN)

```
user@host> show security ike security-associations sa-type shortcut detail
IKE peer 192.168.0.106, Index 4980742, Gateway Name:
GW-ADVPN-GT-ADVPN-zth_spoke_vpn-268173327
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
```



```
  Type: Shortcut, Local Role: Partner, Peer Role: Partner
 Role: Responder, State: UP
```

show security ike security-associations detail (IKEv2 Reauthentication)

```
user@host> show security ike security-associations detail
IKE peer 10.1.2.11, Index 6009224, Gateway Name: GW
  Role: Responder, State: UP
  Initiator cookie: 2c74d14c798a9d70, Responder cookie: 83cbb49bfbcb80cb
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 10.1.1.11:500, Remote: 10.1.2.11:500
  Lifetime: Expires in 173 seconds
  Reauth Lifetime: Expires in 600 seconds
  Peer ike-id: vsrx@example.net
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                 1782
   Output bytes  :                 1743
   Input  packets:                    2
```

show security ike security-associations detail (IKEv2 Fragmentation)

```
user@host> show security ike security-associations detail
IKE peer 172.24.23.157, Index 11883008, Gateway Name: routebased_s2s_gw-552_1
  Role: Responder, State: UP
  Initiator cookie: f3255e720f162e3a, Responder cookie: 17555e3ff7451841
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 192.168.254.1:500, Remote: 172.24.23.157:500
  Lifetime: Expires in 530 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Peer ike-id: 172.24.23.157
  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-5
  Traffic statistics:
   Input  bytes  :                 1004
   Output bytes  :                  756
   Input  packets:                    6
   Output packets:                    4
   Input  fragmented packets:  3
   Output fragmented packets: 3
  IPSec security associations: 1 created, 1 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.254.1:500, Remote: 172.24.23.157:500
    Local identity: 192.168.254.1
    Remote identity: 172.24.23.157
    Flags: IKE SA is created
```


show security ike tunnel-map

Supported Platforms SRX Series, vSRX

Syntax

```
show security ike tunnel-map
brief | summary
fpc slot-number
kmd-instance (all | kmd-instance-name)
pic slot-number
```

Release Information Command introduced in Junos OS Release 12.1X44-D10.

Description Display the tunnel mapping on different Services Processing Units (SPUs) for site-to-site and manual VPNs. You can insert an SPC on a device in a chassis cluster without disrupting traffic on the existing VPN tunnels. After inserting the SPC, you can view the tunnel mapping using this command. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices and vSRX instances.

Options

brief—Display standard information about all existing IKE SAs. This is the default.

fpc *slot-number*—Display information about existing IKE SAs in the specified Flexible PIC Concentrator (FPC) slot.

kmd-instance (all | *kmd-instance-name*)—(Optional) Display information about existing IKE SAs in the key management process (KMD) identified by FPC *slot-number* and PIC *slot-number*. This option is used to filter the output. You can specify one of the following options:

- all—All KMD instances running on the Services Processing Unit (SPU).
- *kmd-instance-name*—Name of the KMD instance running on the SPU.

pic *slot-number*—Display information about existing IKE SAs in the specified PIC slot.

summary—Display the tunnel-mapping load on each SPU. The load is the number of times an SPU has been chosen as an anchor SPU. For site-to-site VPNs, the load should be equal to the number of gateways mapped to an SPU.

Required Privilege Level view

Related Documentation

- Understanding VPN Support for Inserting Services Processing Cards on page 34

List of Sample Output

show security ike tunnel-map on page 1227
show security ike tunnel-map brief on page 1227
show security ike tunnel-map fpc 1 pic 0 on page 1227
show security ike tunnel-map kmd-instance kmd1 on page 1228
show security ike tunnel-map kmd-instance all on page 1228

show security ike tunnel-map summary on page 1228

Output Fields Table 108 on page 1227 lists the output fields for the **show security ike tunnel-map** command. Output fields are listed in the approximate order in which they appear.

Table 108: show security ike tunnel-map Output Fields

Field Name	Field Descripton
Gateway ID	Gateway identifier. This is a nondeterministic number that is constant as long as the configuration is present. This number does not appear in any other outputs.
Gateway Name	Name of the IKE gateway.
FPC	FPC slot number.
PIC	PIC slot number.
IKED Instance	IKE process instance identifier.
SPU Load	Number of times an SPU has been chosen as an anchor SPU.

Sample Output

show security ike tunnel-map

```
user@host> show security ike tunnel-map
Gateway ID   Gateway Name FPC  PIC  IKED Instance
      2        ike_gw1      4    0        1
      3        ike_gw2      7    0        1
      4        ike_gw3      7    0        2
      5        ike_gw4      4    0        2
```

show security ike tunnel-map brief

```
user@host> show security ike tunnel-map brief
Gateway ID   Gateway Name FPC  PIC  IKED Instance
      2        gw-01        1    0        1
      3        LAN_1        1    0        2
      4        LAN_2        1    0        1
      5        LAN_3        1    0        2
      6        LAN_4        1    0        1
```

show security ike tunnel-map fpc 1 pic 0

```
user@host> run show security ike tunnel-map fpc 1 pic 0
Gateway ID   Gateway Name FPC  PIC  IKED Instance
      2        gw-01        1    0        1
      3        LAN_1        1    0        2
      4        LAN_2        1    0        1
      5        LAN_3        1    0        2
      6        LAN_4        1    0        1
```


show security ike tunnel-map kmd-instance kmd1

```
user@host> show security ike tunnel-map kmd-instance kmd1
Gateway ID   Gateway Name FPC  PIC  IKED Instance
    2        gw-01         1    0        1
    4        LAN_2         1    0        1
    6        LAN_4         1    0        1
```

show security ike tunnel-map kmd-instance all

```
user@host> show security ike tunnel-map kmd-instance all
Gateway ID   Gateway Name FPC  PIC  IKED Instance
    2        gw-01         1    0        1
    3        LAN_1         1    0        2
    4        LAN_2         1    0        1
    5        LAN_3         1    0        2
    6        LAN_4         1    0        1
```

show security ike tunnel-map summary

```
user@host> show security ike tunnel-map summary
FPC  PIC  SPU Load
1    0      5
```

show security ipsec control-plane-security-associations

Supported Platforms SRX Series, vSRX

Syntax

show security ipsec control-plane-security-associations
<brief | detail>
<sa-name *sa-name*>

Release Information Command introduced in Junos OS Release 12.1X46-D20.

Description Display information about manual IPsec security associations (SAs) applied to OSPF or OSPFv3 interfaces or virtual links.

Options

- **brief | detail**—(Optional) Display the specified level of output.
- **sa-name** ***sa-name***—Name of the manual SA.

Required Privilege Level view

Related Documentation

- Understanding OSPF and OSPFv3 Authentication on SRX Series Devices on page 37

List of Sample Output

show security ipsec control-plane-security-associations on page 1230
show security ipsec control-plane-security-associations sa-name on page 1230
show security ipsec control-plane-security-associations detail on page 1230

Output Fields Table 109 on page 1229 lists the output fields for the **show security ipsec control-plane-security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 109: show security ipsec control-plane-security-associations Output Fields

Field Name	Field Description
Name	Name of the SA.
Algorithm	IPsec protocol followed by encryption algorithm and authentication algorithm.
SPI	SPI value.
Total active security-associations	Total number of active manual SAs for application to OSPF or OSPFv3 interfaces or virtual links.

Sample Output

show security ipsec control-plane-security-associations

```
user@host> show security ipsec control-plane-security-associations
Name      Algorithm       SPI
test_sa   ESP:3des/md5   3e8
test_sa   ESP:3des/md5   3e8
test_sa2  ESP:3des/sha1  7d1
test_sa2  ESP:3des/sha1  7d1
Total active security-associations: 2
```

show security ipsec control-plane-security-associations sa-name

```
user@host> show security ipsec control-plane-security-associations sa-name test_sa
Name      Algorithm       SPI
test_sa   ESP:3des/md5    3e8
test_sa   ESP:3des/md5    3e8
Total active security-associations: 1
```

show security ipsec control-plane-security-associations detail

```
user@host> show security ipsec control-plane-security-associations detail
Direction: inbound, SA Name: test_sa,
Protocol: ESP:, Authentication: md5
SPI: 3e8, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 1,

Direction: outbound, SA Name: test_sa,
Protocol: ESP:, Authentication: md5
SPI: 3e8, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 2,

Direction: inbound, SA Name: test_sa2,
Protocol: ESP:, Authentication: sha1
SPI: 7d1, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 3,

Direction: outbound, SA Name: test_sa2,
Protocol: ESP:, Authentication: sha1
SPI: 7d1, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 4,
```

show security ipsec inactive-tunnels

Supported Platforms SRX Series, vSRX

Syntax

show security ipsec inactive-tunnels
brief | detail
family (inet | inet6)
fpc *slot-number*
index *index-number*
kmd-instance (all | *kmd-instance-name*)
pic *slot-number*
sa-type shortcut
vpn-name *vpn-name*

Release Information Command introduced in Junos OS Release 11.4R3. Support for Auto Discovery VPN added in Junos OS Release 12.3X48-D10.

Description Display security information about the inactive tunnel.

Options

- none—Display information about all inactive tunnels.
- **brief** | **detail**—(Optional) Display the specified level of output.
- **family**—(Optional) Display the inactive tunnel by family. This option is used to filter the output.
  - **inet**—IPv4 address family.
  - **inet6**—IPv6 address family.
- **fpc *slot-number***—(Optional) Display information about inactive tunnels in the Flexible PIC Concentrator (FPC) slot.
- **index *index-number***—(Optional) Display detailed information about the specified inactive tunnel identified by this index number. For a list of all inactive tunnels with their index numbers, use the command with no options.
- **kmd-instance** —(Optional) Display information about inactive tunnels in the key management process (in this case, it is KMD) identified by FPC *slot-number* and PIC *slot-number*.
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.
- **pic *slot-number***—Display information about inactive tunnels in the PIC slot.
- **sa-type**—(Optional for ADVPN) Type of SA. **shortcut** is the only option for this release.
- **vpn-name *vpn-name***—(Optional) Name of the VPN.

NOTE: The fpc *slot-number*, kmd-instance (all | *kmd-instance-name*), and pic *slot-number* parameters apply to SRX5600 and SRX5800 devices only.

Required Privilege Level view

Related Documentation

- show security ipsec security-associations on page 1236

List of Sample Output

show security ipsec inactive-tunnels on page 1233
show security ipsec inactive-tunnels index 131073 on page 1233
show security ipsec inactive-tunnels sa-type shortcut on page 1233

Output Fields Table 110 on page 1232 lists the output fields for the **show security ipsec inactive-tunnels** command. Output fields are listed in the approximate order in which they appear.

Table 110: show security ipsec inactive-tunnels Output Fields

Field Name	Field Description
Total inactive tunnels	Total number of inactive IPsec tunnels.
Total inactive tunnels which establish immediately	Total number of inactive IPsec tunnels that can establish a session immediately.
ID	Identification number of the inactive tunnel. You can use this number to get more information about the inactive tunnel.
Gateway	IP address of the remote gateway.
Port	If Network Address Translation (NAT) is used, this value is 4500. Otherwise, it is the standard IKE port, 500.
Def-Del#	Number of deferred deletions of a dial-up IPsec VPN.
Virtual system	Virtual system to which the VPN belongs.
VPN name	Name of the IPsec VPN.
Local gateway	Gateway address of the local system.
Remote gateway	Gateway address of the remote system.
Local identity	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IP address, fully qualified domain name, e-mail address, or distinguished name (DN).
Remote identity	IP address of the destination peer gateway.
Version	Version of IKE.
DF-bit	State of the don't fragment bit: **set** or **clear**.
Bind-interface	The tunnel interface to which the route-based VPN is bound.

Table 110: show security ipsec inactive-tunnels Output Fields *(continued)*

Field Name	Field Description
Policy-name	Name of the applicable policy.
Tunnel Down Reason	Reason for which the tunnel is inactive.
Tunnel events	Tunnel event and the number of times the event has occurred. See "Tunnel Events" on page 917 for descriptions of tunnel events and the action you can take.

Sample Output

show security ipsec inactive-tunnels

```
user@host> show security ipsec inactive-tunnels
Total inactive tunnels: 1
  Total inactive tunnels with establish immediately: 0
  ID     Gateway     Port  Tunnel down reason
  131073 192.168.1.2  500   Phase1 proposal mismatch detected
```

show security ipsec inactive-tunnels index 131073

```
user@host> show security ipsec inactive-tunnels index 131073
ID: 131073 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 192.168.1.100, Remote Gateway: 192.168.1.2
  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: st0.0
  Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 600a29
  Tunnel events:
    Wed Jul 16 2014 06:18:02 +0800: User cleared IPSec SA from CLI (1 times)
    Wed Jul 16 2014 06:17:58 +0800: IPSec SA negotiation successfully completed
(1 times)
    Wed Jul 16 2014 06:17:54 +0800: User cleared IPSec SA from CLI (1 times)
    Wed Jul 16 2014 06:16:58 +0800: IPSec SA negotiation successfully completed
(1 times)
   Wed Jul 16 2014 06:16:58 +0800: Bind interface's address received. Information
 updated (1 times)
    Wed Jul 16 2014 06:16:58 +0800: Tunnel is ready. Waiting for trigger event
or peer to trigger negotiation (1 times)
    Wed Jul 16 2014 06:16:58 +0800: External interface's address received.
Information updated (1 times)
    Wed Jul 16 2014 06:16:58 +0800: Bind interface's zone received. Information
updated (1 times)
    Wed Jul 16 2014 06:16:58 +0800: IKE SA negotiation successfully completed (1
 times)
```

show security ipsec inactive-tunnels sa-type shortcut

```
user@host> show security ipsec inactive-tunnels sa-type shortcut
  Total inactive tunnels: 1
  Total inactive tunnels with establish immediately: 0
  ID      Port  Nego#  Fail#  Flag      Gateway          Tunnel Down Reason
  268173322 500 0      0      40608aa9  192.168.0.105       Cleared via CLI
```


show security ipsec next-hop-tunnels

Supported Platforms SRX Series, vSRX

Syntax **show security ipsec next-hop-tunnels**
<interface-name *interface-name*>

Release Information Command introduced in Junos OS Release 8.5.

Description Display security information about the secure tunnel interface.

Options

- none—Display information about all secure tunnel interface.
- **interface-name *interface-name***—(Optional) Name of the secure tunnel logical interface.

Required Privilege Level view

Related Documentation

- show security ipsec security-associations on page 1236

List of Sample Output show security ipsec next-hop-tunnels on page 1235

Output Fields Table 111 on page 1235 lists the output fields for the **show security ipsec next-hop-tunnels** command. Output fields are listed in the approximate order in which they appear.

Table 111: show security ipsec next-hop-tunnels Output Fields

Field Name	Field Description
Next-hop gateway	IP address of the next gateway.
Interface	Name of the secure tunnel logical interface.
IPsec VPN name	Name of the IPsec VPN tunnel.
Flag	• **Static**—IP address manually configured. • **Auto**—IP address obtained from the remote peer automatically.

Sample Output

show security ipsec next-hop-tunnels

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway    interface   IPsec VPN name                  Flag
192.168.1.2         st0.0       autokey                         Static
192.168.1.3         st0.0       pbd-4-6                         Auto
```

show security ipsec security-associations

Supported Platforms SRX Series, vSRX

Syntax

show security ipsec security-associations
brief | detail
family (inet | inet6)
fpc *slot-number*
index *SA-index-number*
kmd-instance (all | *kmd-instance-name*)
pic *slot-number*>
sa-type shortcut
vpn-name *vpn-name* <traffic-selector *traffic-selector-name*>

Release Information Command introduced in Junos OS Release 8.5. Support for the **fpc**, **pic**, and **kmd-instance** options added in Junos OS Release 9.3. Support for the **family** option added in Junos OS Release 11.1. Support for the **vpn-name** option added in Junos OS Release 11.4R3. Support for the **traffic-selector** option and traffic selector field added in Junos OS Release 12.1X46-D10. Support for Auto Discovery VPN (ADVPN) added in Junos OS Release 12.3X48-D10. Support for IPsec datapath verification added in Junos OS Release 15.1X49-D70.

Description Display information about the IPsec security associations (SAs).

Options

- none—Display information about all SAs.
- **brief** | **detail**—(Optional) Display the specified level of output.
- **family**—(Optional) Display SAs by family. This option is used to filter the output.
  - **inet**—IPv4 address family.
  - **inet6**—IPv6 address family.
- **fpc** ***slot-number***—(Optional) Display information about existing IPsec SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.
- **index** ***SA-index-number***—(Optional) Display detailed information about the specified SA identified by this index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.
- **kmd-instance**—(Optional) Display information about existing IPsec SAs in the key management process (in this case, it is KMD) identified by the FPC *slot-number* and PIC *slot-number*. This option is used to filter the output.
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.
- **pic** ***slot-number***—(Optional) Display information about existing IPsec SAs in this PIC slot. This option is used to filter the output.
- **sa-type**—(Optional for ADVPN) Type of SA. **shortcut** is the only option for this release.

- **vpn-name** ***vpn-name***—Name of the VPN. If configured, **traffic-selector** ***traffic-selector-name*** can optionally be specified.

Required Privilege Level

view

Related Documentation

- clear security ipsec security-associations on page 1139
- *Example: Configuring a Route-Based VPN Tunnel in a User Logical System*

List of Sample Output

show security ipsec security-associations (IPv4) on page 1240
show security ipsec security-associations (IPv6) on page 1240
show security ipsec security-associations index 131073 on page 1240
show security ipsec security-associations brief on page 1241
show security ipsec security-associations detail on page 1241
show security ipsec security-associations family inet6 on page 1242
show security ipsec security-associations fpc 6 pic 1 kmd-instance all (SRX Series Devices) on page 1242
show security ipsec security-associations detail (ADVPN Suggester, Static Tunnel) on page 1243
show security ike sa index 222075191 detail on page 1243
show security ipsec security-associations detail (ADVPN Partner, Static Tunnel) on page 1244
show security ike sa index 788674 detail on page 1245
show security ipsec security-associations sa-type shortcut (ADVPN) on page 1245
show security ipsec security-associations sa-type shortcut detail (ADVPN) on page 1246
show security ipsec security-associations family inet detail on page 1246

Output Fields

Table 112 on page 1237 lists the output fields for the **show security ipsec security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 112: show security ipsec security-associations

Field Name	Field Description
Total active tunnels	Total number of active IPsec tunnels.
ID	Index number of the SA. You can use this number to get additional information about the SA.
VPN name	IPsec name for VPN.
Gateway	IP address of the remote gateway.
Port	If Network Address Translation (NAT) is used, this value is 4500. Otherwise, it is the standard IKE port, 500.

Table 112: show security ipsec security-associations *(continued)*

Field Name	Field Description
Algorithm	Cryptography used to secure exchanges between peers during the IKE Phase 2 negotiations includes: • An authentication algorithm used to authenticate exchanges between the peers. Options are **hmac-md5-95**, **hmac-sha1-96,** or **ESP**. • An encryption algorithm used to encrypt data traffic. Options are **3des-cbc, aes-128-cbc, aes-192-cbc**, **aes-256-cbc**, or **des-cbc.**
SPI	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI. Each entry includes the name of the VPN, the remote gateway address, the SPIs for each direction, the encryption and authentication algorithms, and keys. The peer gateways each have two SAs, one resulting from each of the two phases of negotiation: Phase 1 and Phase 2.
Life: sec/kb	The lifetime of the SA, after which it expires, expressed either in seconds or kilobytes.
Sta	State has two options, **Installed** and **Not Installed**. • **Installed**—The SA is installed in the SA database. • **Not Installed**—The SA is not installed in the SA database. For transport mode, the value of State is always **Installed**.
Mon	The Mon field refers to VPN monitoring status. If VPN monitoring is enabled, then this field displays **U** (up) or **D** (down). A hyphen (-) means VPN monitoring is not enabled for this SA. A **V** means that IPsec datapath verification is in progress.
vsys or Virtual-system	The root system.
Tunnel index	Numeric identifier of the specific IPsec tunnel for the SA.
Local gateway	Gateway address of the local system.
Remote gateway	Gateway address of the remote system.
Traffic selector	Name of the traffic selector.
Local identity	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IP address, fully qualified domain name, e-mail address, or distinguished name (DN).
Remote identity	IP address of the destination peer gateway.
DF-bit	State of the don't fragment bit: **set** or **cleared**.
Policy-name	Name of the applicable policy.

Table 112: show security ipsec security-associations *(continued)*

Field Name	Field Description
Location	**FPC**—Flexible PIC Concentrator (FPC) slot number. **PIC**—PIC slot number. **KMD-Instance**—The name of the KMD instance running on the SPU, identified by FPC *slot-number* and PIC *slot-number*. Currently, 4 KMD instances running on each SPU, and any particular IPsec negotiation is carried out by a single KMD instance.
Tunnel events	Tunnel event and the number of times the event has occurred. See "Tunnel Events" on page 917 for descriptions of tunnel events and the action you can take.
Direction	Direction of the SA; it can be inbound or outbound.
AUX-SPI	Value of the auxiliary security parameter index(SPI). • When the value is **AH** or **ESP**, **AUX-SPI** is always 0. • When the value is **AH+ESP**, **AUX-SPI** is always a positive integer.
Mode	Mode of the SA: • **transport**—Protects host-to-host connections. • **tunnel—**Protects connections between security gateways.
Type	Type of the SA: • **manual**—Security parameters require no negotiation. They are static and are configured by the user. • **dynamic**—Security parameters are negotiated by the IKE protocol. Dynamic SAs are not supported in transport mode.
State	State of the SA: • **Installed**—The SA is installed in the SA database. • **Not Installed**—The SA is not installed in the SA database. For transport mode, the value of State is always **Installed**.

Protocol	Protocol supported.
	
	• Transport mode supports Encapsulation Security Protocol (ESP) and Authentication Header (AH).
	• Tunnel mode supports ESP and AH.
	  ◦ **Authentication**—Type of authentication used.
	  ◦ **Encryption**—Type of encryption used.

Soft lifetime	The soft lifetime informs the IPsec key management system that the SA is about to expire. Each lifetime of an SA has two display options, hard and soft, one of which must be present for a dynamic SA. This allows the key management system to negotiate a new SA before the hard lifetime expires. • **Expires in seconds**—Number of seconds left until the SA expires.

Table 112: show security ipsec security-associations *(continued)*

Field Name	Field Description
Hard lifetime	The hard lifetime specifies the lifetime of the SA. • **Expires in seconds**—Number of seconds left until the SA expires.
Lifesize Remaining	The lifesize remaining specifies the usage limits in kilobytes. If there is no lifesize specified, it shows unlimited. • **Expires in kilobytes**—Number of kilobytes left until the SA expires.
Anti-replay service	State of the service that prevents packets from being replayed. It can be **Enabled** or **Disabled**.
Replay window size	Configured size of the antireplay service window. It can be 32 or 64 packets. If the replay window size is 0, the antireplay service is disabled. The antireplay window size protects the receiver against replay attacks by rejecting old or duplicate packets.
Bind-interface	The tunnel interface to which the route-based VPN is bound.
Copy-Outer-DSCP	Indicates if copying outer IP header DSCP and ECN to inner IP header is enabled or disabled.

Sample Output

show security ipsec security-associations (IPv4)

```
user@host> show security ipsec security-associations
Total active tunnels: 1
ID     Gateway              Port  Algorithm       SPI      Life:sec/kb  Mon vsys
  131075 192.168.28.241     500   ESP:3des/sha1   86758ff0 6918/ unlim   -   0
  131075 192.168.28.241     500   ESP:3des/sha1   3183ff26 6918/ unlim   -   0
```

show security ipsec security-associations (IPv6)

```
user@host> show security ipsec security-associations
Total active tunnels: 1
ID      Algorithm     SPI      Life:sec/kb  Mon  vsys Port  Gateway
131074 ESP:3des/sha1 14caf1d9 3597/ unlim   -    root 500   2001:db8::1112
131074 ESP:3des/sha1 9a4db486 3597/ unlim   -    root 500   2001:db8::1112
```

show security ipsec security-associations index 131073

```
user@host> show security ipsec security-associations index 131073
  ID: 131073 Virtual-system: root, VPN Name: ike-vpn-chicago
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  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: IKEv1
  DF-bit: clear
  , Copy-Outer-DSCP Enabled
  Bind-interface: st0.99
```



```
  Port: 500, Nego#: 116, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Tunnel events:
  Fri Oct 30 2015 15:47:21 -0700: IPSec SA rekey successfully completed (115
times)
  Fri Oct 30 2015 11:38:35 -0700: IKE SA negotiation successfully completed (12
times)
  Mon Oct 26 2015 16:41:07 -0700: IPSec SA negotiation successfully completed (1
 times)
  Mon Oct 26 2015 16:40:56 -0700: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Mon Oct 26 2015 16:40:56 -0700: External interface's address received.
Information updated (1 times)
  Location: FPC 0, PIC 1, KMD-Instance 1
  Direction: inbound, SPI: 81b9fc17, AUX-SPI: 0
  Hard lifetime: Expires in 1774 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 1151 seconds
  Mode: Tunnel(0 0), 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
  Location: FPC 0, PIC 1, KMD-Instance 1
  Direction: outbound, SPI: 727f629d, AUX-SPI: 0
  Hard lifetime: Expires in 1774 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 1151 seconds
  Mode: Tunnel(0 0), 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
```

show security ipsec security-associations brief

```
user@host> show security ipsec security-associations brief
Total active tunnels: 2
ID     Gateway     Port Algorithm     SPI      Life:sec/kb Mon vsys
<16384 192.168.1.1 500  ESP:3des/sha1 af88baa  28795/unlim D   0
>16384 192.168.1.1 500  ESP:3des/sha1 f4e3e5f4 28795/unlim D   0
```

show security ipsec security-associations detail

```
user@host> show security ipsec security-associations detail
ID: 131073 Virtual-system: root, VPN Name: ike-vpn-chicago
Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
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: IKEv1
DF-bit: clear
, Copy-Outer-DSCP Enabled
Bind-interface: st0.99

Port: 500, Nego#: 8, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
Tunnel events:
Mon Oct 26 2015 22:27:50 -0700: IPSec SA rekey successfully completed (7 times)
Mon Oct 26 2015 16:41:07 -0700: IPSec SA negotiation successfully completed (1
times)
Mon Oct 26 2015 16:41:07 -0700: IKE SA negotiation successfully completed (1
times)
```

```
Mon Oct 26 2015 16:40:56 -0700: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
Mon Oct 26 2015 16:40:56 -0700: External interface's address received. Information
 updated (1 times)
Location: FPC 0, PIC 1, KMD-Instance 1
Direction: inbound, SPI: 81ed9998, AUX-SPI: 0
Hard lifetime: Expires in 2296 seconds
Lifesize Remaining:  Unlimited
Soft lifetime: Expires in 1688 seconds
Mode: Tunnel(0 0), 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
Location: FPC 0, PIC 1, KMD-Instance 1
Direction: outbound, SPI: 80565248, AUX-SPI: 0
Hard lifetime: Expires in 2296 seconds
Lifesize Remaining:  Unlimited
Soft lifetime: Expires in 1688 seconds
Mode: Tunnel(0 0), 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
```

show security ipsec security-associations family inet6

```
user@host> show security ipsec security-associations family inet6
  Virtual-system: root
  Local Gateway: 2001:db8:1212::1111, Remote Gateway: 2001:db8:1212::1112
  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: 3des-cbc
    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: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations fpc 6 pic 1 kmd-instance all (SRX Series Devices)

```
user@host> show security ipsec security-associations fpc 6 pic 1 kmd-instance all
  Total active tunnels: 1

ID    Gateway          Port  Algorithm       SPI      Life:sec/kb  Mon vsys

<2    192.168.1.2      500   ESP:3des/sha1   67a7d25d 28280/unlim   -    0
```

```
>2    192.168.1.2      500   ESP:3des/sha1   a23cbcdc 28280/unlim   -   0
```

show security ipsec security-associations detail (ADVPN Suggester, Static Tunnel)

```
user@host> show security ipsec security-associations detail
ID: 70516737 Virtual-system: root, VPN Name: ZTH_HUB_VPN
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  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#: 5, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
  Tue Nov 03 2015 01:24:27 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:24:27 -0800: IKE SA negotiation successfully completed (4
times)
  Tue Nov 03 2015 01:23:38 -0800: User cleared IPSec SA from CLI (1 times)
  Tue Nov 03 2015 01:21:32 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:21:31 -0800: IPSec SA delete payload received from peer,
corresponding IPSec SAs cleared (1 times)
  Tue Nov 03 2015 01:21:27 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:21:13 -0800: Tunnel configuration changed. Corresponding
IKE/IPSec SAs are deleted (1 times)
  Tue Nov 03 2015 01:19:27 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:19:27 -0800: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Location: FPC 0, PIC 3, KMD-Instance 2
  Direction: inbound, SPI: 43de5d65, AUX-SPI: 0
  Hard lifetime: Expires in 1335 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 996 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 0, PIC 3, KMD-Instance 2
  Direction: outbound, SPI: 5b6e157c, AUX-SPI: 0
  Hard lifetime: Expires in 1335 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 996 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
```

show security ike sa index 222075191 detail

```
user@host> show security ike sa index 222075191 detail
node0:
--------------------------------------------------------------------------
IKE peer 192.168.1.2, Index 222075191, Gateway Name: ZTH_HUB_GW
```

```
  Location: FPC 0, PIC 3, KMD-Instance 2
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    2
     Suggestions accepted:    4
     Suggestions declined:    1
  Role: Responder, State: UP
  Initiator cookie: 7b996b4c310d2424, Responder cookie: 5724c5882a212157
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.1.1:500, Remote: 192.168.1.2:500
  Lifetime: Expires in 828 seconds
  Peer ike-id: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering,
CN=cssvk36-d
  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  :               20474
   Output bytes  :               21091
   Input  packets:                 237
   Output packets:                 237
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.1.1:500, Remote: 192.168.1.2:500
    Local identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host1
    Remote identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host2
    Flags: IKE SA is created
```

show security ipsec security-associations detail (ADVPN Partner, Static Tunnel)

```
user@host> show security ipsec security-associations detail
ID: 67108872 Virtual-system: root, VPN Name: ZTH_SPOKE_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.1
  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 Nov 03 2015 01:24:26 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:24:26 -0800: IKE SA negotiation successfully completed (4
times)
  Tue Nov 03 2015 01:23:37 -0800: IPSec SA delete payload received from peer,
corresponding IPSec SAs cleared (1 times)
  Tue Nov 03 2015 01:21:31 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:21:31 -0800: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Tue Nov 03 2015 01:18:26 -0800: Key pair not found for configured local
certificate. Negotiation failed (1 times)
  Tue Nov 03 2015 01:18:13 -0800: CA certificate for configured local certificate
```



```
 not found. Negotiation not initiated/successful (1 times)
  Direction: inbound, SPI: 5b6e157c, AUX-SPI: 0
  Hard lifetime: Expires in 941 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 556 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
  Direction: outbound, SPI: 43de5d65, AUX-SPI: 0
  Hard lifetime: Expires in 941 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 556 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
```

show security ike sa index 788674 detail

```
user@host> show security ike sa index 788674 detail
IKE peer 192.168.1.1, Index 788674, Gateway Name: ZTH_SPOKE_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    2
     Suggestions accepted:    2
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: 7b996b4c310d2424, Responder cookie: 5724c5882a212157
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.1.2:500, Remote: 192.168.1.1:500
  Lifetime: Expires in 734 seconds
  Peer ike-id: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering,
CN=test
  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  :                22535
   Output bytes  :                21918
   Input  packets:                  256
   Output packets:                  256
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 192.168.1.2:500, Remote: 192.168.1.1:500
    Local identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host1
    Remote identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host2
    Flags: IKE SA is created
```

show security ipsec security-associations sa-type shortcut (ADVPN)

```
user@host> show security ipsec security-associations sa-type shortcut
```



```
Total active tunnels: 1
ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
<268173318 ESP:aes-cbc-256/sha1 6f164ee0 3580/ unlim - root 500 192.168.0.111

>268173318 ESP:aes-cbc-256/sha1 e6f29cb0 3580/ unlim - root 500 192.168.0.111
```

show security ipsec security-associations sa-type shortcut detail (ADVPN)

```
user@host> show security ipsec security-associations sa-type shortcut detail
node0:
--------------------------------------------------------------------------

ID: 67108874 Virtual-system: root, VPN Name: ZTH_SPOKE_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.2
  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: 4500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x40608a29
  Tunnel events:
    Tue Nov 03 2015 01:47:26 -0800: IPSec SA negotiation successfully completed
(1 times)
    Tue Nov 03 2015 01:47:26 -0800: Tunnel is ready. Waiting for trigger event
or peer to trigger negotiation (1 times)
    Tue Nov 03 2015 01:47:26 -0800: IKE SA negotiation successfully completed (1
 times)
  Direction: inbound, SPI: b7a5518, AUX-SPI: 0
    Hard lifetime: Expires in 1766 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1381 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
  Direction: outbound, SPI: b7e0268, AUX-SPI: 0
    Hard lifetime: Expires in 1766 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1381 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
```

show security ipsec security-associations family inet detail

```
user@host> show security ipsec security-associations family inet detail
ID: 131073 Virtual-system: root, VPN Name: ike-vpn-chicago
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  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: IKEv1
  DF-bit: clear
  , Copy-Outer-DSCP Enabled
  Bind-interface: st0.99

  Port: 500, Nego#: 116, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Tunnel events:
  Fri Oct 30 2015 15:47:21 -0700: IPSec SA rekey successfully completed (115
times)
  Fri Oct 30 2015 11:38:35 -0700: IKE SA negotiation successfully completed (12
```

```
times)
  Mon Oct 26 2015 16:41:07 -0700: IPSec SA negotiation successfully completed (1
 times)
  Mon Oct 26 2015 16:40:56 -0700: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Mon Oct 26 2015 16:40:56 -0700: External interface's address received.
Information updated (1 times)
  Location: FPC 0, PIC 1, KMD-Instance 1
  Direction: inbound, SPI: 81b9fc17, AUX-SPI: 0
  Hard lifetime: Expires in 1713 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 1090 seconds
  Mode: Tunnel(0 0), 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
  Location: FPC 0, PIC 1, KMD-Instance 1
  Direction: outbound, SPI: 727f629d, AUX-SPI: 0
  Hard lifetime: Expires in 1713 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 1090 seconds
  Mode: Tunnel(0 0), 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
```

show security ipsec statistics

Supported Platforms SRX Series, vSRX

Syntax

show security ipsec statistics
<fpc *slot-number*>
<index *SA-index-number*>
<kmd-instance *kmd-instance-name*>
<pic *slot-number*>

Release Information Command introduced in Junos OS Release 8.5. **fpc** and **pic** options added in Junos OS Release 9.3. **kmd-instance** option added in Junos OS Release 10.4.

Description Display standard IPsec statistics.

Options

- none—Display statistics about all IPsec security associations (SAs).
- **fpc** ***slot-number***—Specific to SRX Series devices. Display statistics about existing IPsec SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.
- **index** ***SA-index-number***—(Optional) Display statistics for the SA with this index number.
- **kmd-instance** ***kmd-instance-name***—Specific to SRX Series devices. Display information about existing IKE SAs in the key management process (the daemon, which in this case is KMD) identified by FPC *slot-number* and PIC *slot-number*. This option is used to filter the output.
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.
- **pic** ***slot-number***—Specific to SRX Series devices. Display statistics about existing IPsec SAs in this PIC slot. This option is used to filter the output.

Required Privilege Level view

Related Documentation

- clear security ipsec statistics on page 1140

List of Sample Output

show security ipsec statistics on page 1249
show security ipsec statistics index 5 on page 1250
show security ipsec statistics fpc 6 pic 1 (SRX Series devices) on page 1250

Output Fields Table 113 on page 1249 lists the output fields for the **show security ipsec statistics** command. Output fields are listed in the approximate order in which they appear.

Table 113: show security ipsec statistics Output Fields

Field Name	Field Description
Virtual-system	The root system.
ESP Statistics	• **Encrypted bytes**—Total number of bytes encrypted by the local system across the IPsec tunnel. • **Decrypted bytes**—Total number of bytes decrypted by the local system across the IPsec tunnel. • **Encrypted packets**—Total number of packets encrypted by the local system across the IPsec tunnel. • **Decrypted packets**—Total number of packets decrypted by the local system across the IPsec tunnel.
AH Statistics	• **Input bytes**—Total number of bytes received by the local system across the IPsec tunnel. • **Output bytes**—Total number of bytes transmitted by the local system across the IPsec tunnel. • **Input packets**—Total number of packets received by the local system across the IPsec tunnel. • **Output packets**—Total number of packets transmitted by the local system across the IPsec tunnel.
Errors	• **AH authentication failures**—Total number of authentication header (AH) failures. An AH failure occurs when there is a mismatch of the authentication header in a packet transmitted across an IPsec tunnel. • **Replay errors**—Total number of replay errors. A replay error is generated when a duplicate packet is received within the replay window. • **ESP authentication failures**—Total number of Encapsulation Security Payload (ESP) failures. An ESP failure occurs when there is an authentication mismatch in ESP packets. • **ESP decryption failures**—total number of ESP decryption errors. • **Bad headers**—Total number of invalid headers detected. • **Bad trailers**—Total number of invalid trailers detected.

Sample Output

show security ipsec statistics

```
user@host> show security ipsec statistics
Virtual-system: Root
ESP Statistics:
  Encrypted bytes:              0
  Decrypted bytes:              0
  Encrypted packets:            0
  Decrypted packets:            0
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
```

Sample Output

show security ipsec statistics index 5

```
user@host> show security ipsec statistics index 5
Virtual-system: Root
SA index: 5
ESP Statistics:
  Encrypted bytes:              0
  Decrypted bytes:              0
  Encrypted packets:            0
  Decrypted packets:            0
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
```

Sample Output

show security ipsec statistics fpc 6 pic 1 (SRX Series devices)

```
user@host> show security ipsec statistics fpc 6 pic 1
ESP Statistics:
Encrypted bytes:           536408
Decrypted bytes:           696696
Encrypted packets:           1246
Decrypted packets:            888
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
```

show security ipsec traffic-selector

Supported Platforms SRX Series, vSRX

Syntax

show security ipsec traffic-selector interface-name ***interface-name***
<brief | detail>
<destination-address ***address*****>**
<fpc ***slot-number*****>**
<kmd-instance (all | ***kmd-instance-name*****)>**
<pic ***slot-number*****>**
<source-address ***address*****>**

Release Information Command introduced in Junos OS Release 12.3X48-D10.

Description Display information about the traffic selectors that have been negotiated between the initiator and responder.

Options

interface-name ***interface-name***—Name of the secure tunnel logical interface.

brief | detail —(Optional) Display the specified level of output.

destination-address ***address***—(Optional) Destination IP address.

fpc ***slot-number***—(Optional) Display information about existing IKE SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.

kmd-instance—(Optional) Display information about existing traffic selectors in the key management process (in this case, it is KMD) identified by FPC slot-number and PIC slot-number. This option is used to filter the output.

- **all**—All KMD instances running on the Services Processing Unit (SPU).
- ***kmd-instance-name***—Name of the KMD instance running on the SPU.

pic ***slot-number***—(Optional) Display information about existing traffic selectors in this PIC slot. This option is used to filter the output.

source-address ***address***—(Optional) Source IP address.

Required Privilege Level view

Related Documentation

- IPsec VPN Overview on page 3

List of Sample Output

show security ipsec traffic-selector interface-name st0.1 on page 1252
show security ipsec traffic-selector interface-name st0.1 detail on page 1252

Output Fields Table 114 on page 1252 lists the output fields for the **show security ipsec traffic-selector** command. Output fields are listed in the approximate order in which they appear.

Table 114: show security ipsec traffic-selector Output Fields

Field Name	Field Description
Source IP	Source IP address for the negotiated traffic selector.
Destination IP	Destination IP address for the negotiated traffic selector.
Interface	Secure tunnel (st0) interface for the traffic selector.
Tunnel-id	Tunnel ID.
IKE-ID	Peer IKE ID for the negotiated traffic selector.

Sample Output

show security ipsec traffic-selector interface-name st0.1

```
user@host> show security ipsec traffic-selector interface-name st0.1
 Source IP                        Destination IP                   Interface
Tunnel-id    IKE-ID
 192.0.2.0-192.0.2.255            198.51.100.0-198.51.100.255      st0.1
69206018     DC=Common_component, CN=enodeA, OU=Dept, O=Company, L=City, ST=CA,
C=US
 192.0.2.0-192.0.2.255            203.0.113.0-203.0.113.255        st0.1
77594626     DC=Common_component, CN=enodeB, OU=Det, O=Company, L=City, ST=CA,
C=US
```

show security ipsec traffic-selector interface-name st0.1 detail

```
user@host> show security ipsec traffic-selector interface-name st0.1 detail
 Source IP                        Destination IP                   Interface
Tunnel-id    IKE-ID
 192.168.0.0-192.168.0.255        10.0.0.0-10.0.0.255              st0.1
208666625    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia1
 192.168.1.0-192.168.1.255        10.0.1.0-10.0.1.255              st0.1
213909505    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia2
 192.168.2.0-192.168.2.255        10.0.2.0-10.0.2.255              st0.1
214958081    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia3
 192.168.3.0-192.168.3.255        10.0.3.0-10.0.3.255              st0.1
216006657    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia4
 192.168.4.0-192.168.4.255        10.0.4.0-10.0.4.255              st0.1
217055233    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia5
 192.168.5.0-192.168.255          10.0.5.0-10.0.5.255              st0.1
218103809    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia6
 192.168.6.0-192.168.6.255        10.0.6.0-10.0.6.255              st0.1
219152385    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia7
 192.168.7.0-192.168.7.255        10.0.7.0-10.0.7.255              st0.1
220200961    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia8
 192.168.8.0-192.168.8.255        10.0.8.0-10.0.8.255              st0.1
221249537    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia9
 192.168.9.0-192.168.9.255        10.0.9.0-10.0.9.255              st0.1
222298113    C=US, ST=CA, L=City, O=Ixia, OU=IxLoad, CN=ixia10
```


show security ipsec tunnel-events-statistics

Supported Platforms SRX Series, vSRX

Syntax **show security ipsec tunnel-events-statistics**

Release Information Command introduced in Junos OS Release 12.3X48-D10.

Description Show tunnel event statistics.

Required Privilege Level view

Related Documentation
- *clear security ipsec tunnel-events-statistics*

List of Sample Output show security ipsec tunnel-events statistics on page 1253

Sample Output

show security ipsec tunnel-events statistics

```
user@host> show security ipsec tunnel-events statistics
IPSec SA delete payload received from peer                              : 153
Configuration change triggered clearing of IPSec SA                     : 1
Peer's remote IKE-ID validation failed during negotiation               : 2
Phase1 proposal mismatch detected                                       : 2
Phase2 proposal mismatch detected                                       : 2
Peer proposed traffic-selectors are not in configured range            : 8576
Negotiation failed as peer did not respond                              : 4
IKE SA negotiation successfully completed                               : 19
IPSec SA negotiation successfully completed                             : 154
Tunnel is ready. Waiting for trigger event or peer to trigger negotiation  : 1
```


show security pki ca-certificate (View)

Supported Platforms MX Series, SRX Series, vSRX

Syntax

show security pki ca-certificate
<brief | detail>
<ca-profile *ca-profile-name* >

Release Information Command modified in Junos OS Release 8.5. Subject string output field added in Junos OS Release 12.1X44-D10. Policy identifier output field added in Junos OS Release 12.3X48-D10.

Description Display information about the certificate authority (CA) public key infrastructure (PKI) digital certificates configured on the device.

NOTE: The FIPS image does not permit the use of MD5 fingerprints. Therefore, MD5 fingerprints are not included when a certificate is displayed using this command. The SHA-1 fingerprint that is currently displayed is retained in the FIPS image. The Simple Certificate Enrollment Protocol (SCEP) is disabled in the FIPS image.

Options

- none—Display basic information about all configured CA certificates.
- **brief | detail**—(Optional) Display the specified level of output.
- **ca-profile** ***ca-profile-name***- (Optional) Display information about only the specified CA certificate.

Required Privilege Level view

Related Documentation

- ca-profile (Security PKI)
- request security pki ca-certificate verify (Security)

List of Sample Output

show security pki ca-certificate ca-profile RootCA brief on page 1256
show security pki ca-certificate ca-profile RootCA detail on page 1256
show security pki ca-certificate ca-profile ca-tmp detail on page 1256

Output Fields Table 115 on page 1255 lists the output fields for the **show security pki ca-certificate** command. Output fields are listed in the approximate order in which they appear.

Table 115: show security pki ca-certificate Output Fields

<table>
<tr><th>Field Name</th><th>Field Description</th></tr>
<tr><td>Certificate identifier</td><td>Name of the digital certificate.</td></tr>
<tr><td>Certificate version</td><td>Revision number of the digital certificate.</td></tr>
<tr><td>Serial number</td><td>Unique serial number of the digital certificate.</td></tr>
<tr><td>Issuer</td><td>Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are:
Organization—Organization of origin.
Organizational unit—Department within an organization.
Country—Country of origin.
Locality—Locality of origin.
Common name—Name of the authority.</td></tr>
<tr><td>Subject</td><td>Details of the digital certificate holder organized using the distinguished name format. Possible subfields are:
Organization—Organization of origin.
Organizational unit—Department within an organization.
Country—Country of origin.
Locality—Locality of origin.
Common name—Name of the authority.
If the certificate contains multiple subfield entries, all entries are displayed.</td></tr>
<tr><td>Subject string</td><td>Subject field as it appears in the certificate.</td></tr>
<tr><td>Validity</td><td>Time period when the digital certificate is valid. Values are:
Not before—Start time when the digital certificate becomes valid.
Not after—End time when the digital certificate becomes invalid.</td></tr>
<tr><td>Public key algorithm</td><td>Encryption algorithm used with the private key, such as rsaEncryption(1024 bits).</td></tr>
<tr><td>Signature algorithm</td><td>Encryption algorithm that the CA used to sign the digital certificate, such as sha1WithRSAEncryption.</td></tr>
<tr><td>Certificate Policy</td><td>Policy Identifier—One or more policy object identifiers (OIDs).</td></tr>
<tr><td>Use for key</td><td>Use of the public key, such as Certificate signing, CRL signing, Digital signature, or Data encipherment.</td></tr>
<tr><td>Fingerprint</td><td>Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate.</td></tr>
<tr><td>Distribution CRL</td><td>Distinguished name information and the URL for the certificate revocation list (CRL) server.</td></tr>
</table>

Sample Output

show security pki ca-certificate ca-profile RootCA brief

```
user@host> show security pki ca-certificate ca-profile RootCA brief
Certificate identifier: RootCA
  Issued to: RootCA, Issued by: C = US, O = example, CN = RootCA
  Validity:
    Not before: 05- 3-2012 07:15
    Not after: 05- 2-2017 07:15
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki ca-certificate ca-profile RootCA detail

```
user@host> show security pki ca-certificate ca-profile RootCA detail
Certificate identifier: RootCA
  Certificate version: 3
  Serial number: 0712dc31
  Issuer:
    Organization: example, Country: US, Common name: RootCA
  Subject:
    Organization: example, Country: US, Common name: RootCA
  Subject string:
    C=US, O=example, CN=RootCA
  Validity:
    Not before: 05- 3-2012 07:15
    Not after: 05- 2-2017 07:15
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:ac:b0:c0:11:ac:0c:34:37:04:97:65:c2:b1
    ae:7e:68:e0:fa:37:23:a1:f0:eb:4d:eb:03:89:c9:d9:0d:34:f3:66
    91:97:8c:e9:9c:d4:b5:55:8d:c1:e2:8b:95:08:9d:29:f8:ab:ac:ff
    ae:af:f7:bc:4b:33:f2:eb:b9:e6:13:6d:18:d7:64:a7:85:78:99:41
    4e:b4:fa:bc:3e:1b:5c:26:25:89:03:af:e9:c6:e9:9e:7b:74:1a:1a
    5b:b4:2a:48:78:57:68:e2:5c:0b:71:71:78:ac:a2:23:5f:ca:d2:4a
    38:4c:35:5a:20:cc:44:39:96:26:20:43:bd:75:fd:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Use for key: CRL signing, Certificate signing, Key encipherment,
  Digital signature
  Fingerprint:
    eb:2a:2a:eb:d3:c7:cb:62:65:2e:6a:76:56:b8:af:88:51:8a:30:c9 (sha1)
    cd:43:ae:a4:b2:11:9e:cf:1a:47:fd:7f:0c:ce:d9:fd (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Sample Output

show security pki ca-certificate ca-profile ca-tmp detail

```
user@host> show security pki ca-certificate ca-profile ca-tmp detail
Certificate identifier: ca-tmp
  Certificate version: 3
  Serial number: 00000047
  Issuer:
    Organization: Example,
    Organizational unit: DoD, Organizational unit: Testing, Country: US,
    Common name: Trust Anchor
```



```
Subject:
  Organization: Example,
  Organizational unit: Dod, Organizational unit: Testing, Country: US,
  Common name: CA1-PP.01.03
Subject string:
  C=US, O=Example, OU=Example, 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)
```


show security pki certificate-request (View)

Supported Platforms SRX Series, vSRX

Syntax

show security pki certificate-request
<**brief** | **detail**>
<certificate-id *certificate-id-name* >

Release Information Command modified in Junos OS Release 8.5.

Description Display information about manually generated local digital certificate requests that are stored on the device.

Options

- none—Display basic information about all local digital certificate requests.
- **brief** | **detail**—(Optional) Display the specified level of output.
- **certificate-id** ***certificate-id-name*** —(Optional) Display information about only the specified local digital certificate requests.

Required Privilege Level view

Related Documentation

- clear security pki key-pair (Local Certificate) on page 1142

List of Sample Output

show security pki certificate-request certificate-id user brief on page 1259
show security pki certificate-request certificate-id user detail on page 1259

Output Fields Table 116 on page 1258 lists the output fields for the **show security pki certificate-request** command. Output fields are listed in the approximate order in which they appear.

Table 116: show security pki certificate-request Output Fields

Field Name	Field Description
Certificate identifier	Name of the digital certificate.
Certificate version	Revision number of the digital certificate.
Issued to	Device that was issued the digital certificate.

Table 116: show security pki certificate-request Output Fields *(continued)*

Field Name	Field Description
Subject	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **Country**—Country of origin. • **Locality**—Locality of origin. • **Common name**—Name of the authority.
Alternate subject	Domain name or IP address of the device related to the digital certificate.
Public key algorithm	Encryption algorithm used with the private key, such as **rsaEncryption**(**1024 bits**).
Public key verification status	Public key verification status: **Failed** or **Passed**. The **detail** output also provides the verification hash.
Fingerprint	Secure Hash Algorithm (**SHA1**) and Message Digest 5 (**MD5**) hashes used to identify the digital certificate.
Use for key	Use of the public key, such as **Certificate signing**, **CRL signing**, **Digital signature**, or **Data encipherment**.

Sample Output

show security pki certificate-request certificate-id user brief

```
user@host> show security pki certificate-request certificate-id hassan brief
Certificate identifier: user
      Issued to: user@example.net
      Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki certificate-request certificate-id user detail

```
user@host> show security pki certificate-request certificate-id hassan detail
    Certificate identifier: user
      Certificate version: 3
      Subject:
        Organization: example, Organizational unit: example, Country: IN,
                        Common name: user1
      Alternate subject: 192.168.72.124
      Public key algorithm: rsaEncryption(1024 bits)
      Public key verification status: Passed
        c7:a4:fb:e7:8c:4f:31:e7:eb:01:d8:32:65:21:f2:eb:6f:7d:49:1a:c3:9b
        63:47:e2:4f:f6:db:f6:c8:75:dd:e6:ec:0b:35:0a:62:32:45:6b:35:1f:65
        c9:66:b7:40:b2:f9:2a:ab:5b:60:f7:c7:73:36:da:68:25:fc:40:4b:12:3c
        d5:c8:c6:66:f6:10:1e:86:67:a8:95:9b:7f:1c:ae:a7:55:b0:28:95:a7:9a
        a2:24:28:e4:5a:b2:a9:06:7a:69:37:20:15:e1:b6:66:eb:22:b5:b6:77:f6
        65:88:b0:94:2b:91:4b:99:78:4a:e3:56:cc:14:45:d7:97:fd
      Fingerprint:
```



```
  8f:22:1a:f2:9f:27:b0:21:6c:da:46:64:31:34:1f:68:42:5a:39:e0 (sha1)
  09:15:11:aa:ea:f9:5a:b5:70:d7:0b:8e:be:a6:d3:cb (md5)
Use for key: Digital signature
```

show security pki crl (View)

Supported Platforms SRX Series, vSRX

Syntax

```
show security pki crl
< brief | detail>
<ca-profile ca-profile-name >
```

Release Information Command modified in Junos OS Release 8.5.

Description Display information about the certificate revocation lists (CRLs) configured on the device.

Options

- none—Display basic information about all CRLs.
- **brief | detail**—(Optional) Display the specified level of output.
- **ca-profile** ***ca-profile-name***- (Optional) Display information about only the specified CA profile.

Required Privilege Level view

Related Documentation

- crl (Security) on page 963

List of Sample Output

show security pki crl ca-profile ca2 on page 1262
show security pki crl ca-profile ca2 brief on page 1262
show security pki crl ca-profile ca2 detail on page 1263

Output Fields Table 117 on page 1261 lists the output fields for the **show security pki crl** command. Output fields are listed in the approximate order in which they appear.

Table 117: show security pki crl Output Fields

Field Name	Field Description
CA profile	Name of the configured CA profile.
CRL version	Revision number of the certificate revocation list.

Table 117: show security pki crl Output Fields *(continued)*

Field Name	Field Description
CRL issuer	Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are: • **emailAddress**—Mail address of the issuing authority. • **C**—Country of origin. • **ST**—State of origin. • **L**—Locality of origin. • **O**—Organization of origin. • **OU**—Department within an organization. • **CN**—Name of the authority.
Effective date	Date and time the certificate revocation list becomes valid.
Next update	Date and time the routing platform will download the latest version of the certificate revocation list.
Revocation List	List of digital certificates that have been revoked before their expiration date. Values are: • **Serial number**—Unique serial number of the digital certificate. • **Revocation date**—Date and time that the digital certificate was revoked.

Sample Output

show security pki crl ca-profile ca2

```
user@host> show security pki crl ca-profile ca2
CA profile: ca2
  CRL version: V00000001
  CRL issuer: emailAddress = user@example.net, C = US, ST = ca, L = sunnyvale, O
 = , OU = SPG QA, CN = 2000-spg-example-net
  Effective date: 04-26-2007 18:47
  Next update: 05- 4-2007 07:07
```

Sample Output

show security pki crl ca-profile ca2 brief

```
user@host> show security pki crl ca-profile ca2 brief
CA profile: ca2
  CRL version: V00000001
  CRL issuer: emailAddress = user@example.net, C = US, ST = ca, L = sunnyvale, O
 = example networks, OU = SPG QA, CN = 2000-spg-example-net
  Effective date: 04-26-2007 18:47
  Next update: 05- 4-2007 07:07
```


Sample Output

show security pki crl ca-profile ca2 detail

```
user@host> show security pki crl ca-profile ca2 detail
CA profile: ca2
  CRL version: V00000001
  CRL issuer: emailAddress = user@example.net, C = US, ST = ca, L = sunnyvale, O
 = example, OU = SPG QA, CN = 2000-spg-example-net
  Effective date: 04-26-2007 18:47
  Next update: 05- 4-2007 07:07
  Revocation List:
    Serial number              Revocation date
    174e6399000000000506       03-16-2007 23:09
    174ef3f3000000000507       03-16-2007 23:09
    17529cd6000000000508       03-16-2007 23:09
    1763ac26000000000509       03-16-2007 23:09
    21904e5700000000050a       03-16-2007 23:09
    2191cf7900000000050b       03-16-2007 23:09
    21f10eb600000000050c       03-16-2007 23:09
    2253ca2a00000000050f       03-16-2007 23:09
    2478939b000000000515       03-16-2007 23:09
    24f35004000000000516       03-16-2007 23:09
    277ddfa8000000000517       03-16-2007 23:09
    277e97bd000000000518       03-16-2007 23:09
    27846a76000000000519       03-16-2007 23:09
    2785176f00000000051a       03-16-2007 23:09
```

show security pki local-certificate (View)

Supported Platforms SRX Series, vSRX

Syntax

show security pki local-certificate
< **brief** | **detail** >
< certificate-id ***certificate-id-name*** >
<system-generated>

Release Information Command modified in Junos OS Release 9.1. Subject string output field added in Junos OS Release 12.1X44-D10.

Description Display information about the local digital certificates, corresponding public keys, and the automatically generated self-signed certificate configured on the device.

Options

- none—Display basic information about all configured local digital certificates, corresponding public keys, and the automatically generated self-signed certificate.
- **brief** | **detail**—(Optional) Display the specified level of output.
- certificate-id ***certificate-id-name*** —(Optional) Display information about only the specified local digital certificates and corresponding public keys.
- **system-generated**—Display information about the automatically generated self-signed certificate.

Required Privilege Level view

Related Documentation

- clear security pki local-certificate (Device) on page 1143
- request security pki local-certificate generate-self-signed (Security) on page 1160

List of Sample Output

show security pki local-certificate certificate-id hello on page 1266
show security pki local-certificate certificate-id hello detail on page 1266
show security pki local-certificate system-generated on page 1267
show security pki local-certificate system-generated detail on page 1267
show security pki local-certificate certificate-id mycert - (local certificate enrolled online using SCEP) on page 1268
show security pki local-certificate certificate-id mycert detail - (local certificate enrolled online using SCEP) on page 1268

Output Fields Table 118 on page 1265 lists the output fields for the **show security pki local-certificate** command. Output fields are listed in the approximate order in which they appear.

Table 118: show security pki local-certificate Output Fields

Field Name	Field Description
Certificate identifier	Name of the digital certificate.
Certificate version	Revision number of the digital certificate.
Serial number	Unique serial number of the digital certificate.
Issued to	Device that was issued the digital certificate.
Issued by	Authority that issued the digital certificate.
Issuer	Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are: • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **Country**—Country of origin. • **Locality**—Locality of origin. • **Common name**—Name of the authority.
Subject	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **Country**—Country of origin. • **Locality**—Locality of origin. • **Common name**—Name of the authority. • **Serial number**—Serial number of the device. If the certificate contains multiple subfield entries, all entries are displayed.
Subject string	Subject field as it appears in the certificate.
Alternate subject	Domain name or IP address of the device related to the digital certificate.
Validity	Time period when the digital certificate is valid. Values are: • **Not before**—Start time when the digital certificate becomes valid. • **Not after**—End time when the digital certificate becomes invalid.
Public key algorithm	Encryption algorithm used with the private key, such as **rsaEncryption**(**1024 bits**).
Public key verification status	Public key verification status: **Failed** or **Passed**. The **detail** output also provides the verification hash.
Signature algorithm	Encryption algorithm that the CA used to sign the digital certificate, such as **sha1WithRSAEncryption**.

Table 118: show security pki local-certificate Output Fields *(continued)*

Field Name	Field Description
Fingerprint	Secure Hash Algorithm (**SHA1**) and Message Digest 5 (**MD5**) hashes used to identify the digital certificate.
Distribution CRL	Distinguished name information and URL for the certificate revocation list (**CRL**) server.
Use for key	Use of the public key, such as **Certificate signing**, **CRL signing**, **Digital signature**, or **Data encipherment**.

Sample Output

show security pki local-certificate certificate-id hello

```
user@host> show security pki local-certificate certificate-id hello
Certificate identifier: hello
  Issued to: cn1, Issued by: DC = local, DC = demo, CN = domain-example-WIN-CA
  Validity:
    Not before: 08- 8-2012 17:02
    Not after: 08- 8-2014 17:02
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki local-certificate certificate-id hello detail

```
user@host> show security pki local-certificate certificate-id hello detail
Certificate identifier: hello
  Certificate version: 3
  Serial number: 61ba9da000000000d72e
  Issuer:
    Common name: Example-CA,
    Domain component: local, Domain component: demo
  Subject:
    Organization: o1, Organization: o2,
    Organizational unit: ou1, Organizational unit: ou2, Country: US, State: CA,
    Locality: Sunnyvale, Common name: cn1, Common name: cn2,
    Domain component: dc1, Domain component: dc2
  Subject string:
    C=Example, DC=dc1, DC=dc2, ST=CA, L=Sunnyvale, O=o1, O=o2, OU=ou1, OU=ou2,
CN=cn1, CN=cn2
  Alternate subject: "user@example.net", user.example.net, 192.0.2.1
  Validity:
    Not before: 08- 8-2012 17:02
    Not after: 08- 8-2014 17:02
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b4:14:01:d5:4f:79:87:d5:bb:e6:5e:c1:14
    97:da:b4:40:ad:1a:77:3e:ec:2e:68:8e:e4:93:a3:fe:7c:0b:58:af
    e1:20:27:82:ca:8d:6f:f0:97:d1:ad:fe:df:6c:cb:3c:b0:4f:cc:dd
    ac:d8:69:3f:3c:59:b5:2a:c6:83:e8:b3:94:5e:0a:2d:cd:e2:b0:15
    3e:97:a7:8a:4e:fb:59:f7:20:4c:ba:a8:80:3e:ba:be:69:ef:2b:32
    e4:1a:1c:24:53:1b:d5:c3:aa:d4:25:73:96:76:ea:49:d4:da:7e:3e
    0c:c6:6b:22:43:cb:04:84:0d:25:33:07:6b:49:41:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
```



```
    ldap:///Example-CA,CN=cn-win,CN=CDP,CN=Public%20Key
%20Services,CN=Services,CN=Configuration,DC=demo,DC=local?certificateRevocationList?base?
objectClass=cRLDistributionPoint
    http://example.example.net/CertEnroll/Example-CA.crl
  Use for key: Key encipherment, Digital signature, 1.3.6.1.5.5.8.2.2,
  1.3.6.1.5.5.8.2.2
  Fingerprint:
    76:a8:5f:65:b4:bf:bd:10:d8:56:82:65:ff:0d:04:3a:a5:e9:41:dd (sha1)
    8f:99:a4:15:98:10:4b:b6:1a:3d:81:13:93:2a:ac:e7 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Sample Output

show security pki local-certificate system-generated

```
user@host> show security pki local-certificate system-generated
Certificate identifier: system-generated
  Issued to: JN10B9390AGB, Issued by: CN = JN10B9390AGB, CN = system generated,
CN = self-signed
  Validity:
    Not before: 10-30-2009 23:02
    Not after: 10-29-2014 23:02
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki local-certificate system-generated detail

```
user@host> show security pki local-certificate system-generated detail
Certificate identifier: system-generated
  Certificate version: 3
  Serial number: e90d42ebd14ef954b3e48c2eed5b30fb
  Issuer:
    Common name: JN10B9390AGB, Common name: system generated, Common name:
self-signed
  Subject:
    Common name: JN10B9390AGB, Common name: system generated, Common name:
self-signed
  Subject string:
    CN=JN10B9390AGB, CN=system generated, CN=self-signed
  Validity:
    Not before: 10-30-2009 23:02
    Not after: 10-29-2014 23:02
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:cb:c8:3f:e6:d3:e5:ca:9d:dc:2d:e9:ca:c7
    5f:b1:f5:3a:f0:1c:a7:55:43:0f:ef:fd:1c:fe:29:09:d5:37:d0:fa
    d6:ee:bc:b8:3f:58:d4:31:fb:96:4f:4f:cc:a9:1a:8f:2e:1b:50:6f
    2b:88:34:74:b2:6d:ad:94:b5:dd:3d:80:87:56:d0:42:50:4d:ac:d7
    8c:21:06:2d:07:1e:f4:d0:c7:85:2e:25:60:ad:1b:b5:b2:d2:1d:c8
    79:67:8c:56:06:04:75:6e:be:4e:99:b8:07:e6:9a:11:fe:b5:ec:c0
    1e:68:da:47:99:1b:b2:c8:07:ab:cd:6e:fe:c1:fd:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    be:1f:21:13:71:cd:9d:de:7a:41:d7:4c:52:8d:3e:d6:ba:db:75:96 (sha1)
    ba:fc:90:4b:5f:a8:66:a3:b9:64:89:9f:e2:45:b5:84 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```


Sample Output

show security pki local-certificate certificate-id mycert - (local certificate enrolled online using SCEP)

```
user@host> show security pki local-certificate certificate-id mycert
Certificate identifier: mycert
  Issued to: bubba, Issued by: DC = local, DC = demo, CN = domain-example-WIN-CA

  Validity:
    Not before: 11-15-2012 18:58
    Not after: 11-15-2014 18:58
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki local-certificate certificate-id mycert detail - (local certificate enrolled online using SCEP)

```
user@host> show security pki local-certificate certificate-id mycert detail
Certificate identifier: mycert
  Certificate version: 3
  Serial number: 1f00b50a000000013ad2
  Issuer:
    Common name: Example-CA,
    Domain component: local, Domain component: demo
  Subject:
    Organization: example, Organizational unit: SSD, Country: US,
    Common name: host1, Serial number: SRX240-11152012
  Subject string:
    serialNumber=SRX240-11152012, C=US, O=example, OU=SSD, CN=host1
  Alternate subject: "user@example.net", user.example.net, 192.0.2.1
  Validity:
    Not before: 11-15-2012 18:58
    Not after: 11-15-2014 18:58
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:e3:e5:ae:c0:82:af:db:94:01:2f:56:46:50
    7d:3d:0b:0c:f0:1f:1d:7d:c3:aa:d4:4c:a0:cd:23:8b:3f:47:05:ee
    7b:65:42:a0:dc:c4:ac:a7:b6:a6:9f:5c:ea:d8:22:b0:bf:03:75:09
    be:fa:77:cb:d6:67:19:e6:80:fa:a5:7c:93:af:96:66:9f:cc:45:d5
    eb:ab:c1:f0:32:a6:d9:27:1b:80:bb:57:ec:31:a2:e0:2b:e1:42:c0
    92:8a:9b:ed:a6:d2:ec:7c:84:5a:8a:d9:96:a7:7e:40:c3:80:0e:f4
    d6:a2:5d:78:93:3b:7d:d5:8a:f5:de:fb:bc:0d:6d:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    ldap:///Example-CA,CN=cn-win,CN=CDP,CN=Public%20Key%20Services,
CN=Services,CN=Configuration,DC=demo,DC=local?certificateRevocationList?
base?objectClass=cRLDistributionPoint
    http://example.example.net/CertEnroll/Example-CA.crl
  Use for key: Key encipherment, Digital signature, 1.3.6.1.5.5.8.2.2,
  1.3.6.1.5.5.8.2.2
  Fingerprint:
    1f:2f:a9:22:a8:d5:a9:36:cc:c4:bd:81:59:9d:9c:58:bb:40:15:72 (sha1)
    51:27:e4:d5:29:90:f7:85:9e:67:84:a1:75:d1:5b:16 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```


show security tcp-encap connection

Supported Platforms SRX Series, vSRX

Syntax

show security tcp-encap connection
<brief | detail>
<session-id *session-id*>

Release Information Command introduced in Junos OS Release 15.1X49-D80. This command is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices.

Description Display information about TCP encapsulation sessions.

Options

none——Display information about TCP encapsulation sessions.

brief | detail—(Optional) Display the specified level of output.

session-id ***session-id***—(Optional) Display information for the specified session identifier.

Required Privilege Level view

Related Documentation

- tcp-encap on page 1098

List of Sample Output

show security tcp-encap connection on page 1270
show security tcp-encap connection detail on page 1270
show security tcp-encap connection session-id 644 on page 1270

Output Fields Table 119 on page 1269 lists the output fields for the **show security tcp-encap connection** command. Output fields are listed in the approximate order in which they appear.

Table 119: show security tcp-encap connection Output Fields

Field Name	Field Description
Session-Id	Session identifier.
Client	Name of the remote access client.
Gateway	IP address of the remote gateway.
Local Gateway	IP address of the local gateway.
Remote Gateway	IP address of the remote gateway.
Started	Date and time the connection started.
Anchor spu	Services Processing Unit (SPU) on which the connection is anchored.

Sample Output

show security tcp-encap connection

```
user@host> show security tcp-encap connection
Session-Id    Client            Gateway
   34         NCP-1              10.4.0.1
   644        NCP-1              10.5.0.1
```

show security tcp-encap connection detail

```
user@host> show security tcp-encap connection detail
Session id: 34
    Local Gateway: 10.4.0.2:500 , Remote Gateway: 10.4.0.1:9500
    Client: NCP-1
    Started: Sun Jan 08 2017 21:32:58
    Anchor spu: 1

Session id: 644
    Local Gateway: 10.4.0.2:443 , Remote Gateway: 10.5.0.1:9500
    Client: NCP-1
    Started: Sun Jan 08 2017 21:32:58
    Anchor spu: 1
```

show security tcp-encap connection session-id 644

```
user@host> show security tcp-encap connection session-id 644
Session id: 644
    Local Gateway: 10.4.0.2:443 , Remote Gateway: 10.5.0.1:9500
    Client: NCP-1
    Started: Sun Jan 08 2017 21:32:58
    Anchor spu: 1
```


show security tcp-encap statistics

Supported Platforms SRX Series, vSRX

Syntax show security tcp-encap statistics

Release Information Command introduced in Junos OS Release 15.1X49-D80. This command is supported on all SRX Series devices except for SRX5400, SRX5600, and SRX5800 devices.

Description Display TCP encapsulation statistics.

Required Privilege Level view

Related Documentation

- clear security tcp-encap statistics on page 1144

List of Sample Output show security tcp-encap statistics on page 1271

Output Fields Table 120 on page 1271 lists the output fields for the **show security tcp-encap statistics** command. Output fields are listed in the approximate order in which they appear.

Table 120: show security tcp-encap statistics Output Fields

Field Name	Field Description
Policy Matched	Number of policies matched.
TCP sessions	Number of TCP sessions.

Sample Output

show security tcp-encap statistics

```
user@host> show security tcp-encap statistics
TCP encapsulation statistics:
  Policy Matched:                    16
  TCP sessions:                      16
```

