IPsec Properties

Published: 2013-02-15

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

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:

- M Series
- T Series
- MX Series
- J 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 the CLI User Guide.

Documentation Conventions

Table 1 on page xiii 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.

Table 2 on page xiii 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`

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

Convention	Description	Examples
Italic text like this	• Introduces or emphasizes important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration 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)	Enclose 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)	Enclose a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identify 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.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web 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 J-Web 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 send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release 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 JNASC 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: https://www.juniper.net/alerts/

- 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

IPsec

IPsec Overview

The Juniper Networks Junos OS supports IPsec. This section discusses the following topics, which provide background information about configuring IPsec.

For a list of the IPsec and IKE standards supported by the Junos OS, see the *Junos OS Hierarchy and RFC Reference*.

IPsec

The IPsec architecture provides a security suite for the IP version 4 (IPv4) and IP version 6 (IPv6) network layers. The suite provides such functionality as authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. In addition to IPsec, the Junos OS also supports the Internet Key Exchange (IKE), which defines mechanisms for key generation and exchange, and manages security associations (SAs).

IPsec also defines a security association and key management framework that can be used with any network layer protocol. The SA specifies what protection policy to apply to traffic between two IP-layer entities. IPsec provides secure tunnels between two peers.

Security Associations

To use IPsec security services, you create SAs between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. There are two types of SAs:

- Manual SAs require no negotiation; 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 ends of the tunnel. Each peer must have the same configured options for communication to take place.

- Dynamic SAs require additional configuration. With dynamic SAs, you configure IKE first and then the SA. IKE creates dynamic security associations; it negotiates SAs for IPsec. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. This connection is then used to dynamically agree upon keys and other data used by the dynamic IPsec SA. The IKE SA is negotiated first and then used to protect the negotiations that determine the dynamic IPsec SAs.

IKE

IKE is a key management protocol that creates dynamic SAs; it negotiates SAs for IPsec. An IKE configuration defines the algorithms and keys used to establish a secure connection with a peer security gateway.

IKE performs the following tasks:

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

Two versions of the IKE protocol (IKEv1 and IKEv2) are supported now. IKE negotiates security attributes and establishes shared secrets to form the bidirectional IKE SA. In IKE, inbound and outbound IPsec SAs are established and the IKE SA secures the exchanges. Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. IKE also generates keying material, provides Perfect Forward Secrecy, and exchanges identities.

Comparison of IPsec Services and ES Interface Configuration

Table 3 on page 4 compares the top-level configuration of IPsec features on the ES PIC interfaces and on the AS or MultiServices PIC interfaces.

Table 3: Statement Equivalents for ES and AS Interfaces

ES PIC Configuration	AS and MultiServices PIC IPsec Configuration
[edit security ipsec] proposal {...}	[edit services ipsec-vpn ipsec] proposal {...}
[edit security ipsec] policy {...}	[edit services ipsec-vpn ipsec] policy {...}
[edit security ipsec] security-association sa-dynamic {...}	[edit services ipsec-vpn rule *rule-name*] term *term-name* match-conditions {...} then dynamic {...}]
[edit security ipsec] security-association sa-manual {...}	[edit services ipsec-vpn rule *rule-name*] term *term-name* match-conditions {...} then manual {...}]

Table 3: Statement Equivalents for ES and AS Interfaces *(continued)*

ES PIC Configuration	AS and MultiServices PIC IPsec Configuration
[edit security ike] proposal {...}	[edit services ipsec-vpn ike] proposal {...}
[edit security ike] policy {...}	[edit services ipsec-vpn ike] policy {...}
Not available	[edit services ipsec-vpn] rule-set {...}
Not available	[edit services ipsec-vpn] service-set {...}
[edit interfaces es-*fpc/pic/port*] tunnel source *address*	[edit services ipsec-vpn service-set *set-name* ipsec-vpn local-gateway *address*]
[edit interfaces es-*fpc/pic/port*] tunnel destination *address*	[edit services ipsec-vpn rule *rule-name*] remote-gateway *address*

For more information about configuring IPsec services on an AS or MultiServices PIC, see IPsec Properties. For more information about configuring encryption services on an ES PIC, see Configuring Encryption Interfaces.

NOTE: Although many of the same statements and properties are valid on both platforms, the configurations are not interchangeable. You must commit a complete configuration for the PIC type that is installed in your router.

PART 2

Configuration

CHAPTER 2

IPsec Services Configuration Guidelines

Minimum Security Association Configurations

The following sections show the minimum configurations necessary to set up security associations (SAs) for IPsec services:

Minimum Manual SA Configuration

To define a manual SA configuration, you must include at least the following statements at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then manual]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
direction (inbound | outbound | bidirectional) {
  authentication {
    algorithm (hmac-md5-96 | hmac-sha1-96);
    key (ascii-text key | hexadecimal key);
  }
  encryption {
    algorithm algorithm;
    key (ascii-text key | hexadecimal key);
  }
  protocol (ah | esp | bundle);
  spi spi-value;
}
```

Minimum Dynamic SA Configuration

To define a dynamic SA configuration, you must include at least the following statements at the **[edit services ipsec-vpn]** hierarchy level:

```
[edit services ipsec-vpn]
ike {
  proposal proposal-name {
    authentication-algorithm (md5 | sha1 | sha-256);
    authentication-method pre-shared-keys;
    dh-group (group1 | group2 | group5 |group14);
```

```
    encryption-algorithm algorithm;
  }
  policy policy-name {
    proposals [ ike-proposal-names ];
    pre-shared-key (ascii-text key | hexadecimal key);
    version (1 | 2);
    mode (aggressive | main);
  }
}
ipsec {
  policy policy-name {
    proposals [ ipsec-proposal-names ];
  }
  proposal proposal-name {
    authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
    encryption-algorithm algorithm;
    protocol (ah | esp | bundle);
  }
}
```

NOTE:

- Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. The **version** statement under the **[edit services ipsec-vpn ike policy *name*]** hierarchy allows you to configure the specific IKE version to be supported.
- The **mode** statement under the **[edit services ipsec-vpn ike policy *name*]** hierarchy is required only if the **version** option is set to 1.

You must also include the **ipsec-policy** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then dynamic]** hierarchy level.

CHAPTER 3

Configuration Tasks for IPsec Services

Configuring Security Associations

To use IPsec services, you create an SA between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. You can configure two types of SAs:

- Manual—Requires no negotiation; all values, including the keys, are static and specified in the configuration. As a result, each peer must have the same configured options for communication to take place. For information about how to configure a manual SA, see "Configuring Manual Security Associations" on page 12.
- Dynamic—Specifies proposals to be negotiated with the tunnel peer. The keys are generated as part of the negotiation and therefore do not need to be specified in the configuration. The dynamic SA includes one or more **proposal** statements, which allow you to prioritize a list of protocols and algorithms to be negotiated with the peer. For information about how to configure a dynamic SA, see "Configuring Dynamic Security Associations" on page 16.

This section includes the following topics:

NOTE: Both OSPFv2 and OSPFv3 support IPsec authentication. However, dynamic or tunnel mode IPsec SAs are not supported for OSPFv3. If you add SAs into OSPFv3 by including the **ipsec-sa** statement at the **[edit protocols ospf3 area *area-number* interface *interface-name*]** hierarchy level, your configuration fails to commit. For more information about OSPF authentication and other OSPF properties, see the Junos OS Routing Protocols Configuration Guide.

Configuring Manual Security Associations

Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. As a result, each peer must have the same configured options for communication to take place.

To configure a manual IPsec security association, include statements at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then manual]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
direction (inbound | outbound | bidirectional) {
  authentication {
    algorithm (hmac-md5-96 | hmac-sha1-96);
    key (ascii-text key | hexadecimal key);
  }
  auxiliary-spi auxiliary-spi-value;
  encryption {
    algorithm algorithm;
    key (ascii-text key | hexadecimal key);
  }
  protocol (ah | esp | bundle);
  spi spi-value;
}
```

To configure manual SA statements, do the following:

Configuring the Direction for IPsec Processing

The **direction** statement specifies inbound or outbound IPsec processing. If you want to define different algorithms, keys, or security parameter index (SPI) values for each direction, you configure the **inbound** and **outbound** options. If you want the same attributes in both directions, use the **bidirectional** option.

To configure the direction of IPsec processing, include the **direction** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then manual]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
  direction (inbound | outbound | bidirectional) {
  ...
}
```

Example: Using Different Configuration for the Inbound and Outbound Directions

Define different algorithms, keys, and security parameter index values for each direction:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
direction inbound {
  protocol esp;
  spi 16384;
  encryption {
    algorithm 3des-cbc;
    key ascii-text 2345678901234567890234;
  }
}
direction outbound {
  protocol esp;
  spi 24576;
  encryption {
    algorithm 3des-cbc;
    key ascii-text 12345678901234567890abcd;
  }
}
```

Example: Using the Same Configuration for the Inbound and Outbound Directions

Define one set of algorithms, keys, and security parameter index values that is valid in both directions:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
direction bidirectional {
  protocol ah;
  spi 20001;
  authentication {
    algorithm hmac-md5-96;
    key ascii-text 123456789012abcd;
  }
}
```

Configuring the Protocol for a Manual IPsec SA

IPsec uses two protocols to protect IP traffic: Encapsulating Security Payload (ESP) and authentication header (AH). The AH protocol is used for strong authentication. A third option, **bundle**, uses AH authentication and ESP encryption; it does not use ESP authentication because AH provides stronger authentication of IP packets.

To configure the IPsec protocol, include the **protocol** statement and specify the **ah**, **esp**, or **bundle** option at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then manual direction *direction*]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
protocol (ah | bundle | esp);
```

Configuring the Security Parameter Index

An SPI is an arbitrary value that uniquely identifies which SA to use at the receiving host. The sending host uses the SPI to identify and select which SA to use to secure every packet. The receiving host uses the SPI to identify and select the encryption algorithm and key used to decrypt packets.

NOTE: Each manual SA must have a unique SPI and protocol combination. Use the auxiliary SPI when you configure the **protocol** statement to use the **bundle** option.

To configure the SPI, include the **spi** statement and specify a value (from 256 through 16,639) at the **[edit services ipsec-vpn rule** *rule-name* **term** *term-name* **then manual direction** *direction***]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
spi spi-value;
```

Configuring the Auxiliary Security Parameter Index

Use the auxiliary SPI when you configure the **protocol** statement to use the **bundle** option.

NOTE: Each manual SA must have a unique SPI and protocol combination.

To configure the auxiliary SPI, include the **auxiliary-spi** statement and specify a value (from 256 through 16,639) at the **[edit services ipsec-vpn rule** *rule-name* **term** *term-name* **then manual direction** *direction***]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
auxiliary-spi auxiliary-spi-value;
```

Configuring Authentication for a Manual IPsec SA

To configure an authentication algorithm, include the **authentication** statement and specify an authentication algorithm and a key at the **[edit services ipsec-vpn rule** *rule-name* **term** *term-name* **then manual direction** *direction***]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
  authentication {
  algorithm (hmac-md5-96 | hmac-sha1-96);
  key (ascii-text key | hexadecimal key);
}
```

The algorithm can be one of the following:

- **hmac-md5-96**—Hash algorithm that authenticates packet data. It produces a 128-bit authenticator value and a 96-bit digest.
- **hmac-sha1-96**—Hash algorithm that authenticates packet data. It produces a 160-bit authenticator value and a 96-bit digest.

The key can be one of the following:

- **ascii-text**—ASCII text key. With the **hmac-md5-96** option, the key contains 16 ASCII characters. With the **hmac-sha1-96** option, the key contains 20 ASCII characters.
- **hexadecimal**—Hexadecimal key. With the **hmac-md5-96** option, the key contains 32 hexadecimal characters. With the **hmac-sha1-96** option, the key contains 40 hexadecimal characters.

Configuring Encryption for a Manual IPsec SA

To configure IPsec encryption, include the **encryption** statement and specify an algorithm and key at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then manual direction *direction*]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
  encryption {
  algorithm algorithm;
  key (ascii-text key | hexadecimal key);
}
```

The algorithm can be one of the following:

- **des-cbc**—Encryption algorithm that has a block size of 8 bytes; its key size is 64 bits long.
- **3des-cbc**—Encryption algorithm that has a block size of 24 bytes; its key size is 192 bits long.
- **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.

NOTE: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409, *The Internet Key Exchange (IKE)*. The AES encryption algorithms use a software implementation that has much lower throughput, so DES remains the recommended option. For reference information on AES encryption, see RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*.

For **3des-cbc**, the first 8 bytes should differ from the second 8 bytes, and the second 8 bytes should be the same as the third 8 bytes.

If you configure an authentication proposal but do not include the **encryption** statement, the result is NULL encryption. Certain applications expect this result. If you configure no specific authentication or encryption values, the Junos OS uses the default values of **sha1** for the authentication and **3des-cbc** for the encryption.

The key can be one of the following:

- **ascii-text**—ASCII text key. With the **des-cbc** option, the key contains 8 ASCII characters. With the **3des-cbc** option, the key contains 24 ASCII characters.
- **hexadecimal**—Hexadecimal key. With the **des-cbc** option, the key contains 16 hexadecimal characters. With the **3des-cbc** option, the key contains 48 hexadecimal characters.

NOTE: You cannot configure encryption when you use the AH protocol.

Configuring Dynamic Security Associations

You configure dynamic SAs with a set of proposals that are negotiated by the security gateways. The keys are generated as part of the negotiation and therefore do not need to be specified in the configuration. The dynamic SA includes one or more proposals, which allow you to prioritize a list of protocols and algorithms to be negotiated with the peer.

To enable a dynamic SA, follow these steps:

1. Configure Internet Key Exchange (IKE) proposals and IKE policies associated with these proposals.
2. Configure IPsec proposals and an IPsec policy associated with these proposals.
3. Associate an SA with an IPsec policy by configuring the **dynamic** statement.

For more information about IKE policies and proposals, see "Configuring IKE Policies" on page 20 and "Configuring IKE Proposals" on page 17. For more information about IPsec policies and proposals, see "Configuring IPsec Policies" on page 28.

To configure a dynamic SA, include the **dynamic** statement and specify an IPsec policy name at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level. The **ike-policy** statement is optional unless you use the preshared key authentication method.

```
[edit services ipsec-vpn rule rule-name term term-name then]
dynamic {
   ike-policy policy-name;
   ipsec-policy policy-name;
}
```

NOTE: If you want to establish a dynamic SA, the attributes in at least one configured IPsec and IKE proposal must match those of its peer.

Clearing Security Associations

You can set up the router software to clear IKE or IPsec SAs automatically when the corresponding services PIC restarts or is taken offline. To configure this property, include

the **clear-ike-sas-on-pic-restart** or **clear-ipsec-sas-on-pic-restart** statement at the **[edit services ipsec-vpn]** hierarchy level:

```
[edit services ipsec-vpn]
clear-ike-sas-on-pic-restart;
clear-ipsec-sas-on-pic-restart;
```

After you add this statement to the configuration, all the IKE or IPsec SAs corresponding to the tunnels in the PIC will be cleared when the PIC restarts or goes offline.

Configuring IKE Proposals

Dynamic security associations (SAs) require IKE configuration. With dynamic SAs, you configure IKE first, and then the SA. IKE creates the dynamic SAs and negotiates them for IPsec. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway.

You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

NOTE: In IKEv1, the authentication method for SAs is negotiated with the remote peer based on the type of authentication method configured in the IKE proposal. In IKEv2, such a negotiation is not performed with the remote peer. Instead, each IKE peer uses the authentication method that is locally configured for them.

For SAs in IKEv2, the authentication method is either the default value as IKEv1 (if another authentication method is not configured in the IKE proposal) or all IKEv2 proposals in the IKE policy must be configured with the same authentication method value.

To configure an IKE proposal, include the **proposal** statement and specify a name at the **[edit services ipsec-vpn ike]** hierarchy level:

```
[edit services ipsec-vpn ike]
proposal proposal-name {
   authentication-algorithm (md5 | sha1 | sha-256);
   authentication-method (dsa-signatures | pre-shared-key | rsa-signatures);
   dh-group (group1 | group2 | group5 |group14);
   encryption-algorithm algorithm;
   lifetime-seconds seconds;
}
```

This section includes the following topics:

- Configuring the Lifetime for an IKE SA on page 20
- Example: Configuring an IKE Proposal on page 20

Configuring the Authentication Algorithm for an IKE Proposal

To configure the authentication algorithm for an IKE proposal, include the **authentication-algorithm** statement at the **[edit services ipsec-vpn ike proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
authentication-algorithm (md5 | sha1 | sha-256);
```

The authentication algorithm can be one of the following:

- **md5**—Produces a 128-bit digest.
- **sha1**—Produces a 160-bit digest.
- **sha-256**—Produces a 256-bit digest.

NOTE: For reference information on Secure Hash Algorithms (SHAs), see Internet draft **draft-eastlake-sha2-02.txt**, *Secure Hash Algorithms (SHA and HMAC-SHA)* (expires July 2006).

Configuring the Authentication Method for an IKE Proposal

To configure the authentication method for an IKE proposal, include the **authentication-method** statement at the **[edit services ipsec-vpn ike proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
```

The authentication method can be one of the following:

- **dsa-signatures**—Digital Signature Algorithm
- **pre-shared-keys**—A key derived from an out-of-band mechanism; the key authenticates the exchanges
- **rsa-signatures**—Public key algorithm (supports encryption and digital signatures)

Configuring the Diffie-Hellman Group for an IKE Proposal

Diffie-Hellman is a public-key cryptography scheme that allows two parties to establish a shared secret over an insecure communications channel. It is also used within IKE to establish session keys.

To configure the Diffie-Hellman group for an IKE proposal, include the **dh-group** statement at the **[edit services ipsec-vpn ike proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
dh-group (group1 | group2 | group5 |group14);
```

The group can be one of the following:

- **group1**—Specifies that IKE use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group2**—Specifies that IKE use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group5**—Specifies that IKE use the 1536-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group14**—Specifies that IKE use the 2048-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

Using a Diffie-Hellman group based on a greater number of bits results a more secure IKE tunnel than using a group based on fewer bits. However, this additional security entails additional processing time.

Configuring the Encryption Algorithm for an IKE Proposal

To configure the encryption algorithm for an IKE proposal, include the **encryption-algorithm** statement at the **[edit services ipsec-vpn ike proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
encryption-algorithm algorithm;
```

The encryption algorithm can be one of the following:

- **3des-cbc**—Cipher block chaining encryption algorithm with a key size of 24 bytes; its key size is 192 bits long.
- **des-cbc**—Cipher block chaining encryption algorithm with a key size of 8 bytes; its key size is 56 bits long.
- **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.

NOTE: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409, *The Internet Key Exchange (IKE)*. The AES encryption algorithms use a software implementation that has much lower throughput, so DES remains the recommended option.

For **3des-cbc**, the first 8 bytes should differ from the second 8 bytes, and the second 8 bytes should be the same as the third 8 bytes.

If you configure an authentication proposal but do not include the **encryption** statement, the result is NULL encryption. Certain applications expect this result. If you configure no specific authentication or encryption values, the Junos OS uses the default values of **sha1** for the authentication and **3des-cbc** for the encryption.

Configuring the Lifetime for an IKE SA

The **lifetime-seconds** statement sets the lifetime of an IKE SA. When the IKE SA expires, it is replaced by a new SA (and SPI) or the IPsec connection is terminated.

To configure the lifetime for an IKE SA, include the **lifetime-seconds** statement at the **[edit services ipsec-vpn ike proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
lifetime-seconds seconds;
```

By default, the IKE SA lifetime is 3600 seconds. The range is from 180 through 86,400 seconds.

NOTE: In IKEv1, the lifetime for SAs is negotiated with the remote peer based on the type of lifetime configured in the IKE proposal. In IKEv2, such a negotiation is not performed with the remote peer. Instead, each IKE peer uses the lifetime that is locally configured for them.

For SAs in IKEv2, the lifetime is either the default value as IKEv1 (if another lifetime is not configured in the IKE proposal) or all IKEv2 proposals in the IKE policy must be configured with the same lifetime value.

NOTE: For IKE proposals, there is only one SA lifetime value, specified by the Junos OS. IPsec proposals use a different mechanism; for more information, see "Configuring the Lifetime for an IPsec SA" on page 27.

Example: Configuring an IKE Proposal

Configure an IKE proposal:

```
[edit services ipsec-vpn ike]
proposal ike-proposal {
   authentication-method pre-shared-keys;
   dh-group group1;
   authentication-algorithm sha1;
   encryption-algorithm 3des-cbc;
}
```

Configuring IKE Policies

An IKE policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address and the proposals needed for that connection. Depending on which authentication method is used, it defines the preshared key for the given peer or the local certificate. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used. The configured preshared key must also match its peer.

Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. You can configure the specific IKE phase to be supported for the negotiation. However, if only IKEv1 is supported, the Junos OS rejects IKEv2 negotiations. Similarly, if only IKEv2 is supported, the Junos OS rejects all IKEv1 negotiations.

The key management process (kmd) daemon determines which version of IKE is used in a negotiation. If kmd is the IKE initiator, it uses IKEv1 by default and retains the configured version for negotiations. If kmd is the IKE responder, it accepts connections from both IKEv1 and IKEv2.

You can create multiple, prioritized proposals at each peer to ensure that at least one proposal matches a remote peer's proposal.

First, you configure one or more IKE proposals; then you associate these proposals with an IKE policy. You can also prioritize a list of proposals used by IKE in the **policy** statement by listing the proposals you want to use, from first to last.

To configure an IKE policy, include the **policy** statement and specify a policy name at the **[edit services ipsec-vpn ike]** hierarchy level:

```
[edit services ipsec-vpn ike]
policy policy-name {
    description description;
    local-certificate identifier;
    local-id (ipv4_addr ipv4-address | ipv6-addr ipv6-address | key-id identifier);
    version (1 | 2);
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposals [ proposal-names ];
    remote-id {
        any-remote-id;
        ipv4_addr [ values ];
        ipv6_addr [ values ];
        key_id [ values ];
    }
}
```

This section includes the following topics:

For an example of an IKE policy configuration, see "Example: Configuring an IKE Policy" on page 25.

Configuring the IKE Phase

Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. You can configure the specific IKE phase to be supported for the negotiation. However, if only IKEv1 is supported, the Junos OS rejects IKEv2 negotiations. Similarly, if only IKEv2 is supported, the Junos OS rejects all IKEv1 negotiations.

To configure the IKE phase used, include the **version** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
version (1 | 2);
```

Configuring the Mode for an IKE Policy

IKE policy has two modes: aggressive and main. By default, main mode is enabled. Main mode uses six messages, in three exchanges, to establish the IKE SA. (These three steps are IKE SA negotiation, a Diffie-Hellman exchange, and authentication of the peer.) Main mode also allows a peer to hide its identity.

Aggressive mode also establishes an authenticated IKE SA and keys. However, aggressive mode uses half the number of messages, has less negotiation power, and does not provide identity protection. The peer can use the aggressive or main mode to start IKE negotiation; the remote peer accepts the mode sent by the peer.

NOTE: **The mode configuration is required only if the version option is set to 1.**

To configure the mode for an IKE policy, include the **mode** statement and specify **aggressive** or **main** at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
mode (aggressive | main);
```

Configuring the Proposals in an IKE Policy

The IKE policy includes a list of one or more proposals associated with an IKE policy.

To configure the proposals in an IKE policy, include the **proposals** statement and specify one or more proposal names at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
proposals [ proposal-names ];
```

Configuring the Preshared Key for an IKE Policy

When you include the **authentication-method pre-shared-keys** statement at the **[edit services ipsec-vpn ike proposal *proposal-name*]** hierarchy level, IKE policy preshared keys authenticate peers; for more information, see "Configuring the Authentication Method for an IKE Proposal" on page 18. You must manually configure a preshared key, which must match that of its peer. The preshared key can be an ASCII text (alphanumeric) key or a hexadecimal key.

To configure the preshared key in an IKE policy, include the **pre-shared-key** statement and a key at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
pre-shared-key (ascii-text key | hexadecimal key);
```

The key can be one of the following:

- **ascii-text**—ASCII text key. With the **des-cbc** option, the key contains 8 ASCII characters. With the **3des-cbc** option, the key contains 24 ASCII characters.
- **hexadecimal**—Hexadecimal key. With the **des-cbc** option, the key contains 16 hexadecimal characters. With the **3des-cbc** option, the key contains 48 hexadecimal characters.

Configuring the Local Certificate for an IKE Policy

When you include the **authentication-method rsa-signatures** statement at the **[edit services ipsec-vpn ike proposal *proposal-name*]** hierarchy level, public key infrastructure (PKI) digital certificates authenticate peers; for more information, see "Configuring the Authentication Method for an IKE Proposal" on page 18. You must identify a local certificate that is sent to the peer during the IKE authentication phase.

To configure the local certificate for an IKE policy, include the **local-certificate** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
local-certificate identifier;
```

The **local-certificate** statement specifies the identifier used to obtain the end entity's certificate from the certification authority. Configuring it in an IKE policy allows you the flexibility of using a separate certificate with each remote peer if that is needed. You must also specify the identity of the certification authority by configuring the **ca-profile** statement at the **[edit security pki]** hierarchy level; for more information, see the Junos OS System Basics Configuration Guide. For complete examples of digital certificate configuration, see the Junos OS Feature Guides.

You can use the configured profiles to establish a set of trusted certification authorities for use with a particular service set. This enables you to configure separate service sets for individual clients to whom you are providing IP services; the distinct service sets provide logical separation of one set of IKE sessions from another, using different local gateway addresses, or *virtualization*. To configure the set of trusted certification authorities, include

the **trusted-ca** statement at the **[edit services service-set *service-set-name* ipsec-vpn-options]** hierarchy level:

```
[edit services service-set service-set-name ipsec-vpn-options]
trusted-ca ca-profile;
```

For more information, see Configuring IPsec Service Sets.

Configuring a Certificate Revocation List

A certificate revocation list (CRL) contains a list of digital certificates that have been cancelled before their expiration date. When a participating peer uses a digital certificate, it checks the certificate signature and validity. It also acquires the most recently issued CRL and checks that the certificate serial number is not on that CRL.

NOTE: By default, certificate revocation list verification is enabled. You can disable CRL verification by including the **disable** statement at the **[edit security pki ca-profile *ca-profile-name* revocation-check]** hierarchy level.

By default, if the router either cannot access the Lightweight Directory Access Protocol (LDAP) URL or retrieve a valid certificate revocation list, certificate verification fails and the IPsec tunnel is not established. To override this behavior and permit the authentication of the IPsec peer when the CRL is not downloaded, include the **disable on-download-failure** statement at the **[edit security pki ca-profile *ca-profile-name* revocation-check crl]** hierarchy level.

To use the CA certificate revocation list, you include statements at the **[edit security pki ca-profile *ca-profile-name* revocation-check]** hierarchy level. For details, see the Junos OS System Basics Configuration Guide.

Configuring the Description for an IKE Policy

To specify an optional text description for an IKE policy, include the **description** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
description description;
```

Configuring Local and Remote IDs for IKE Phase 1 Negotiation

You can optionally specify local identifiers for use in IKE phase 1 negotiation. If the **local-id** statement is omitted, the local gateway address is used.

To specify one or more local IDs, include the **local-id** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
local-id (ipv4_addr ipv4-address | ipv6-addr ipv6-address | key-id identifier);
```

You can also specify remote gateway identifiers for which the IKE policy is used. The remote gateway address in which this policy is defined is added by default.

To specify one or more remote IDs, include the **remote-id** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
  remote-id {
  any-remote-id;
  ipv4_addr [ values ];
  ipv6_addr [ values ];
  key_id [ values ];
}
```

The **any-remote-id** option allows any remote address to connect. This option is supported only in dynamic endpoints configurations and cannot be configured along with specific values. For more information about dynamic endpoint configurations, see "Configuring Dynamic Endpoints for IPsec Tunnels" on page 38.

Example: Configuring an IKE Policy

Define two IKE policies: **policy 10.1.1.2** and **policy 10.1.1.1**. Each policy is associated with **proposal-1** and **proposal-2**. The following configuration uses only IKEv1 for negotiation.

```
[edit services ipsec-vpn]
ike {
  proposal proposal-1 {
    authentication-method pre-shared-keys;
    dh-group group1;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 1000;
  }
  proposal proposal-2 {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm md5;
    encryption-algorithm des-cbc;
    lifetime-seconds 10000;
  }
  proposal proposal-3 {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm md5;
    encryption-algorithm des-cbc;
    lifetime-seconds 10000;
  }
  policy 10.1.1.2 {
    mode main;
    proposals [ proposal-1 proposal-2 ];
    pre-shared-key ascii-text example-pre-shared-key;
  }
  policy 10.1.1.1 {
    local-certificate certificate-file-name;
    local-key-pair private-public-key-file;
    mode aggressive;
    proposals [ proposal-2 proposal-3 ]
    pre-shared-key hexadecimal 0102030abbcd;
  }
```

}

NOTE: Updates to the current IKE proposal and policy configuration are not applied to the current IKE SA; updates are applied to new IKE SAs.

If you want the new updates to take immediate effect, you must clear the existing IKE security associations so that they will be reestablished with the changed configuration. For information about how to clear the current IKE security association, see the Junos OS Operational Mode Commands.

Configuring IPsec Proposals

An IPsec proposal lists protocols and algorithms (security services) to be negotiated with the remote IPsec peer.

To configure an IPsec proposal, include the **proposal** statement and specify an IPsec proposal name at the **[edit services ipsec-vpn ipsec]** hierarchy level:

```
[edit services ipsec-vpn ipsec]
proposal proposal-name {
  authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
  description description;
  encryption-algorithm algorithm;
  lifetime-seconds seconds;
  protocol (ah | esp | bundle);
}
```

This section discusses the following topics:

Configuring the Authentication Algorithm for an IPsec Proposal

To configure the authentication algorithm for an IPsec proposal, include the **authentication-algorithm** statement at the **[edit services ipsec-vpn ipsec proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
```

The authentication algorithm can be one of the following:

- **hmac-md5-96**—Hash algorithm that authenticates packet data. It produces a 128-bit digest. Only 96 bits are used for authentication.
- **hmac-sha1-96**—Hash algorithm that authenticates packet data. It produces a 160-bit digest. Only 96 bits are used for authentication.

Configuring the Description for an IPsec Proposal

To specify an optional text description for an IPsec proposal, include the **description** statement at the **[edit services ipsec-vpn ipsec proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
description description;
```

Configuring the Encryption Algorithm for an IPsec Proposal

To configure encryption algorithm for an IPsec proposal, include the **encryption-algorithm** statement at the **[edit services ipsec-vpn ipsec proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
encryption-algorithm algorithm;
```

The encryption algorithm can be one of the following:

- **3des-cbc**—Encryption algorithm that has a block size of 24 bytes; its key size is 192 bits long.
- **des-cbc**—Encryption algorithm that has a block size of 8 bytes; its key size is 48 bits long.
- **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.

NOTE: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409, *The Internet Key Exchange (IKE)*. The AES encryption algorithms use a software implementation that has much lower throughput, so DES remains the recommended option.

For **3des-cbc**, the first 8 bytes should differ from the second 8 bytes, and the second 8 bytes should be the same as the third 8 bytes.

If you configure an authentication proposal but do not include the **encryption** statement, the result is NULL encryption. Certain applications expect this result. If you configure no specific authentication or encryption values, the Junos OS uses the default values of **sha1** for the authentication and **3des-cbc** for the encryption.

Configuring the Lifetime for an IPsec SA

When a dynamic IPsec SA is created, two types of lifetimes are used: hard and soft. The hard lifetime specifies the lifetime of the SA. The soft lifetime, which is derived from the hard lifetime, informs the IPsec key management system that the SA is about to expire. This allows the key management system to negotiate a new SA before the hard lifetime expires.

NOTE: In IKEv1, the lifetime for SAs is negotiated with the remote peer based on the type of lifetime configured in the IPsec proposal. In IKEv2, such a negotiation is not performed with the remote peer. Instead, each IKE peer uses the lifetime that is locally configured for them.

For SAs in IKEv2, the lifetime is either the default value as IKEv1 (if another lifetime is not configured in the IPsec proposal) or all IKEv2 proposals in the IPsec policy must be configured with the same lifetime value.

To configure the hard lifetime value, include the **lifetime-seconds** statement and specify the number of seconds at the **[edit services ipsec-vpn ipsec proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
lifetime-seconds seconds;
```

The default lifetime is 28,800 seconds. The range is from 180 through 86,400 seconds.

The soft lifetime values are as follows:

- Initiator: Soft lifetime = Hard lifetime – 135 seconds.
- Responder: Soft lifetime = Hard lifetime – 90 seconds.

Configuring the Protocol for a Dynamic SA

The **protocol** statement sets the protocol for a dynamic SA. IPsec uses two protocols to protect IP traffic: ESP and AH. The ESP protocol can support authentication, encryption, or both. The AH protocol is used for strong authentication. AH also authenticates the IP packet. The **bundle** option uses AH authentication and ESP encryption; it does not use ESP authentication because AH provides stronger authentication of IP packets.

To configure the protocol for a dynamic SA, include the **protocol** statement and specify the **ah**, **esp**, or **bundle** option at the **[edit services ipsec-vpn ipsec proposal *proposal-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
protocol (ah | esp | bundle);
```

Configuring IPsec Policies

An IPsec policy defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines Perfect Forward Secrecy (PFS) and the proposals needed for the connection. During the IPsec negotiation, IPsec looks for a proposal that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used.

You can create multiple, prioritized IPsec proposals at each peer to ensure that at least one proposal matches a remote peer's proposal.

First, you configure one or more IPsec proposals; then you associate these proposals with an IPsec policy. You can prioritize a list of proposals used by IPsec in the **policy** statement by listing the proposals you want to use, from first to last.

To configure an IPsec policy, include the **policy** statement, and specify the policy name and one or more proposals to associate with the policy, at the **[edit services ipsec-vpn ipsec]** hierarchy level:

```
[edit services ipsec-vpn ipsec]
policy policy-name {
  description description;
  perfect-forward-secrecy {
    keys (group1 | group2 | group5 | group14);
  }
  proposals [ proposal-names ];
}
```

This section includes the following topics related to configuring an IPsec policy:

Configuring the Description for an IPsec Policy

To specify an optional text description for an IPsec policy, include the **description** statement at the **[edit services ipsec-vpn ipsec policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec policy policy-name]
description description;
```

Configuring Perfect Forward Secrecy

PFS provides additional security by means of a Diffie-Hellman shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys. This statement is optional.

To configure PFS, include the **perfect-forward-secrecy** statement and specify a Diffie-Hellman group at the **[edit services ipsec-vpn ipsec policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec policy policy-name]
perfect-forward-secrecy {
  keys (group1 | group2 | group5 | group14);
}
```

The key can be one of the following:

- **group1**—Specifies that IKE use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group2**—Specifies that IKE use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group5**—Specifies that IKE use the 1536-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group14**—Specifies that IKE use the 2048-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

The higher numbered groups provide more security than the lowered numbered groups,, but require more processing time.

Configuring the Proposals in an IPsec Policy

The IPsec policy includes a list of one or more proposals associated with an IPsec policy.

To configure the proposals in an IPsec policy, include the **proposals** statement and specify one or more proposal names at the **[edit services ipsec-vpn ipsec policy *policy-name*]** hierarchy level:

```
[edit services ipsec-vpn ipsec policy policy-name]
proposals [ proposal-names ];
```

Example: Configuring an IPsec Policy

Define an IPsec policy, **dynamic policy-1**, that is associated with two proposals (**dynamic-1** and **dynamic-2**):

```
[edit services ipsec-vpn ipsec]
proposal dynamic-1 {
   protocol esp;
   authentication-algorithm hmac-md5-96;
   encryption-algorithm 3des-cbc;
   lifetime-seconds 6000;
}
proposal dynamic-2 {
   protocol esp;
   authentication-algorithm hmac-sha1-96;
   encryption-algorithm 3des-cbc;
   lifetime-seconds 6000;
}
policy dynamic-policy-1 {
   perfect-forward-secrecy {
      keys group1;
   }
   proposals [ dynamic-1 dynamic-2 ];
}
```

NOTE: Updates to the current IPsec proposal and policy configuration are not applied to the current IPsec SA; updates are applied to new IPsec SAs.

If you want the new updates to take immediate effect, you must clear the existing IPsec security associations so that they will be reestablished with the changed configuration. For information about how to clear the current IPsec security association, see the Junos OS Operational Mode Commands.

Configuring IPsec Rules

To configure an IPsec rule, include the **rule** statement and specify a rule name at the **[edit services ipsec-vpn]** hierarchy level:

```
[edit services ipsec-vpn]
rule rule-name {
  match-direction (input | output);
  term term-name {
    from {
      destination-address address;
      ipsec-inside-interface interface-name;
      source-address address;
    }
    then {
      anti-replay-window-size bits;
      backup-remote-gateway address;
      clear-dont-fragment-bit;
      dynamic {
        ike-policy policy-name;
        ipsec-policy policy-name;
      }
      initiate-dead-peer-detection;
      manual {
        direction (inbound | outbound | bidirectional) {
          authentication {
            algorithm (hmac-md5-96 | hmac-sha1-96);
            key (ascii-text key | hexadecimal key);
          }
          auxiliary-spi spi-value;
          encryption {
            algorithm algorithm;
            key (ascii-text key | hexadecimal key);
          }
          protocol (ah | bundle | esp);
          spi spi-value;
        }
      }
      no-anti-replay;
      remote-gateway address;
      syslog;
      tunnel-mtu bytes;
    }
  }
```

```
}
```

Each IPsec rule consists of a set of terms, similar to a firewall filter. A term consists of the following:

- **from** statement—Specifies the match conditions and applications that are included and excluded.
- **then** statement—Specifies the actions and action modifiers to be performed by the router software.

The following sections explain how to configure the components of IPsec rules:

- Configuring Match Direction for IPsec Rules on page 32
- Configuring Match Conditions in IPsec Rules on page 32
- Configuring Actions in IPsec Rules on page 34

Configuring Match Direction for IPsec Rules

Each rule must include a **match-direction** statement that specifies whether the match is applied on the input or output side of the interface. To configure where the match is applied, include the **match-direction (input | output)** statement at the **[edit services ipsec-vpn rule *rule-name*]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name]
match-direction (input | output);
```

The match direction is used with respect to the traffic flow through the AS or Multiservices PIC. When a packet is sent to the PIC, direction information is carried along with it.

With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.

With a next-hop service set, packet direction is determined by the interface used to route the packet to the AS or Multiservices PIC. If the inside interface is used to route the packet, the packet direction is input. If the outside interface is used to direct the packet to the PIC, the packet direction is output. For more information on inside and outside interfaces, see Configuring Service Sets to be Applied to Services Interfaces.

On the AS or Multiservices PIC, a flow lookup is performed. If no flow is found, rule processing is performed. All rules in the service set are considered. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that match the packet direction are considered.

Configuring Match Conditions in IPsec Rules

To configure the match conditions in an IPsec rule, include the **from** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name*]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name]
from {
  destination-address address;
  ipsec-inside-interface interface-name;
  source-address address;
```

```
}
```

You can use either the source address or the destination address as a match condition, in the same way that you would configure a firewall filter; for more information, see the Routing Policy Configuration Guide.

IPsec services support both IPv4 and IPv6 address formats. If you do not specifically configure either the source address or destination address, the default value **0.0.0.0/0** (IPv4 ANY) is used. To use IPv6 ANY (**0::0/128**) as either source or destination address, you must configure it explicitly.

For next-hop-style service sets only, the **ipsec-inside-interface** statement allows you to assign a logical interface to the tunnels established as a result of this match condition. The **inside-service-interface** statement that you can configure at the **[edit services service-set *name* next-hop-service]** hierarchy level allows you to specify **.1** and **.2** as inside and outside interfaces. However, you can configure multiple adaptive services logical interfaces with the **service-domain inside** statement and use one of them to configure the **ipsec-inside-interface** statement. For more information, see Configuring Service Sets to be Applied to Services Interfaces and Interface Properties.

The Junos OS evaluates the criteria you configure in the **from** statement. If multiple link-type tunnels are configured within the same next-hop-style service set, the **ipsec-inside-interface** value enables the rule lookup module to distinguish a particular tunnel from other tunnels in case the source and destination addresses for all of them are **0.0.0.0/0** (ANY-ANY).

NOTE: When you configure the ipsec-inside-interface statement, interface-style service sets are not supported.

A special situation is provided by a term containing an “any-any” match condition (usually because the **from** statement is omitted). If there is an any-any match in a tunnel, a flow is not needed, because all flows within this tunnel use the same security association (SA) and packet selectors do not play a significant role. As a result, these tunnels will use packet-based IPsec. This strategy saves some flow resources on the PIC, which can be used for other tunnels that need a flow-based service.

The following configuration example shows an any-any tunnel configuration with no **from** statement in **term-1**. Missing selectors in the **from** clause result in a packet-based IPsec service.

```
services {
   ipsec-vpn {
      rule rule-1 {
         term term-1 {
            then {
               remote-gateway 10.1.0.1;
               dynamic {
                  ike-policy ike_policy;
                  ipsec-policy ipsec_policy;
               }
            }
```

```
        }
        match-direction input;
      }
      .....
    }
```

Flowless IPsec service is provided to link-type tunnels with an any-any matching, as well as to dynamic tunnels with any-any matching in both dedicated and shared mode.

For link-type tunnels, a mixture of flowless and flow-based IPsec is supported within a service set. If a service set includes some terms with any-any matching and some terms with selectors in the **from** clause, packet-based service is provided for the any-any tunnels and flow-based service is provided for the other tunnels with selectors.

For non link-type tunnels, if a service set contains both any-any terms and selector-based terms, flow-based service is provided to all the tunnels.

Configuring Actions in IPsec Rules

To configure actions in an IPsec rule, include the **then** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name*]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name]
then {
  anti-replay-window-size bits;
  backup-remote-gateway address;
  clear-dont-fragment-bit;
  dynamic {
    ike-policy policy-name;
    ipsec-policy policy-name;
  }
  initiate-dead-peer-detection;
  manual {
    direction (inbound | outbound | bidirectional) {
      authentication {
        algorithm (hmac-md5-96 | hmac-sha1-96);
        key (ascii-text key | hexadecimal key);
      }
      auxiliary-spi spi-value;
      encryption {
        algorithm algorithm;
        key (ascii-text key | hexadecimal key);
      }
      protocol (ah | bundle | esp);
      spi spi-value;
    }
  }
  no-anti-replay;
  remote-gateway address;
  syslog;
  tunnel-mtu bytes;
}
```

The principal IPsec actions are to configure a dynamic or manual SA:

- You configure a dynamic SA by including the **dynamic** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level and referencing policies you have configured at the **[edit services ipsec-vpn ipsec]** and **[edit services ipsec-vpn ike]** hierarchy levels; for more information, see "Configuring Dynamic Security Associations" on page 16.
- You configure a manual SA by including the **manual** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level; for more information, see "Configuring Manual Security Associations" on page 12.

You can configure the following additional properties:

Enabling IPsec Packet Fragmentation

To enable fragmentation of IP version 4 (IPv4) packets in IPsec tunnels, include the **clear-dont-fragment-bit** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
clear-dont-fragment-bit;
```

Setting the **clear-dont-fragment-bit** statement clears the Don't Fragment (DF) bit in the packet header, regardless of the packet size. If the packet size exceeds the tunnel maximum transmission unit (MTU) value, the packet is fragmented before encapsulation. For IPsec tunnels, the default MTU value is 1500 regardless of the interface MTU setting.

Configuring Destination Addresses for Dead Peer Detection

To specify the remote address to which the IPsec traffic is directed, include the **remote-gateway** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
remote-gateway address;
```

To specify a backup remote address, include the **backup-remote-gateway** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
backup-remote-gateway address;
```

These two statements support both IPv4 and IPv6 address formats.

Configuring the **backup-remote-gateway** statement enables the dead peer detection (DPD) protocol, which monitors the tunnel state and remote peer availability. When the primary tunnel defined by the **remote-gateway** statement is active, the backup tunnel is

in standby mode. If the DPD protocol determines that the primary remote gateway address is no longer reachable, a new tunnel is established to the backup address.

If there is no incoming traffic from a peer during a defined interval of 10 seconds, the router detects a tunnel as inactive. A global timer polls all tunnels every 10 seconds and the Adaptive Services (AS) or Multiservices Physical Interface Card (PIC) sends a message listing any inactive tunnels. If a tunnel becomes inactive, the router takes the following steps to failover to the backup address:

1. The adaptive services message triggers the DPD protocol to send a hello message to the peer.
2. If no acknowledgment is received, two retries are sent at 2-second intervals, and then the tunnel is declared dead.
3. Failover takes place if the tunnel is declared dead or there is an IPsec Phase 1 negotiation timeout. The primary tunnel is put in standby mode and the backup becomes active.
4. If the negotiation to the backup tunnel times out, the router switches back to the primary tunnel. If both peers are down, it tries the failover six times. It then stops failing over and reverts to the original configuration, with the primary tunnel active and the backup in standby mode.

You can also enable triggering of DPD Hello messages without configuring a backup remote gateway by including the **initiate-dead-peer-detection** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
initiate-dead-peer-detection;
```

The monitoring behavior is the same as described for the **backup-remote-gateway** statement. This configuration enables the router to initiate DPD Hellos when a backup IPsec gateway does not exist and clean up the IKE and IPsec SAs in case the IKE peer is not reachable.

If the DPD protocol determines that the primary remote gateway address is no longer reachable, a new tunnel is established to the backup address. However, when you configure **initiate-dead-peer-detection** without a backup remote gateway address and the DPD protocol determines that the primary remote gateway address is no longer reachable, the tunnel is declared dead and IKE and IPsec SAs are cleaned up.

For more information on the DPD protocol, see RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*.

Configuring or Disabling IPsec Anti-Replay

To configure the size of the IPsec antireplay window, include the **anti-replay-window-size** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
anti-replay-window-size bits;
```

anti-replay-window-size can take values in the range from 64 through 4096 bits. The default value is 64 bits for AS PICs and 128 bits for Multiservices PICs and DPCs. AS PICs can support a maximum replay window size of 1024 bits, whereas Multiservices PICs and DPCs can support a maximum replay window size of 4096 bits. When the software is committing an IPsec configuration , the key management process (kmd) is unable to differentiate between the service interface types. As a result, if the maximum antireplay window size exceeds 1024 for AS PICs, the commit succeeds and no error message is produced. However, the software internally sets the antireplay window size for AS PICs to 1024 bits even if the configured value of the **anti-replay-window-size** is larger.

To disable the IPsec antireplay feature, include the **no-anti-replay** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
no-anti-replay;
```

By default, antireplay service is enabled. Occasionally this can cause interoperability issues with other vendors' equipment.

Enabling System Log Messages

To record an alert in the system logging facility, include the **syslog** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
syslog;
```

Specifying the MTU for IPsec Tunnels

To configure a specific maximum transmission unit (MTU) value for IPsec tunnels, include the **tunnel-mtu** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
tunnel-mtu bytes;
```

NOTE: The **tunnel-mtu** setting is the only place you need to configure an MTU value for IPsec tunnels. Inclusion of an **mtu** setting at the **[edit interfaces sp-*fpc/pic/port* unit *logical-unit-number* family inet]** hierarchy level is not supported.

Configuring IPsec Rule Sets

The **rule-set** statement defines a collection of IPsec rules that determine what actions the router software performs on packets in the data stream. You define each rule by specifying a rule name and configuring terms. Then, you specify the order of the rules by including the **rule-set** statement at the **[edit services ipsec-vpn]** hierarchy level with a **rule** statement for each rule:

```
[edit services ipsec-vpn]
rule-set rule-set-name {
  rule rule-name;
```

```
}
```

The router software processes the rules in the order in which you specify them in the configuration. If a term in a rule matches the packet, the router performs the corresponding action and the rule processing stops. If no term in a rule matches the packet, processing continues to the next rule in the rule set. If none of the rules matches the packet, the packet is dropped by default.

Configuring Dynamic Endpoints for IPsec Tunnels

IPsec tunnels can also be established using *dynamic peer* security gateways, in which the remote ends of tunnels do not have a statically assigned IP address. Since the remote address is not known and might be pulled from an address pool each time the remote host reboots, establishment of the tunnel relies on using IKE **main** mode with either preshared global keys or digital certificates that accept any remote identification value. For more information on IKE policy modes, see "Configuring the Mode for an IKE Policy" on page 22. Both policy-based and link-type tunnels are supported:

- Policy-based tunnels used shared mode.
- Link-type or routed tunnels use dedicated mode. Each tunnel allocates a service interface from a pool of interfaces configured for the dynamic peers. Routing protocols can be configured to run on these service interfaces to learn routes over the IPsec tunnel that is used as a link in this scenario.

This section includes the following topics:

Authentication Process

The remote (dynamic peer) initiates the negotiations with the local (Juniper Networks) router. The local router uses the default IKE and IPsec policies to match the proposals sent by the remote peer to negotiate the security association (SA) values. Implicit proposals contain a list of all the supported transforms that the local router expects from all the dynamic peers.

If preshared key authentication is used, the preshared key is global for a service set. When seeking the preshared key for the peer, the local router matches the peer's source address against any explicitly configured preshared keys in that service set. If a match is not found, the local router uses the global preshared key for authentication. This key is the one configured in the IKE access profile referenced by the service set.

Phase 2 of the authentication matches the *proxy identities* of the protected hosts and networks sent by the peer against a list of configured proxy identities. The accepted proxy identity is used to create the dynamic rules for encrypting the traffic. You can configure proxy identities by including the **allowed-proxy-pair** statement in the IKE access profile. If no entry matches, the negotiation is rejected.

If you do not configure the **allowed-proxy-pair** statement, the default value **ANY(0.0.0.0/0)-ANY** is applied, and the local router accepts any proxy identities sent by the peer. Both IPv4 and IPv6 addresses are accepted, but you must configure all IPv6 addresses manually.

Once the phase 2 negotiation completes successfully, the router builds the dynamic rules and inserts the reverse route into the routing table using the accepted proxy identity.

Implicit Dynamic Rules

After successful negotiation with the dynamic peer, the key management process (kmd) creates a dynamic rule for the accepted phase 2 proxy and applies it on the local AS or Multiservices PIC. The source and destination addresses are specified by the accepted proxy. This rule is used to encrypt traffic directed to one of the end hosts in the phase 2 proxy identity.

The dynamic rule includes an **ipsec-inside-interface** value, which is the interface name assigned to the dynamic tunnel. The **source-address** and **destination-address** values are accepted from the proxy ID. The **match-direction** value is **input** for next-hop-style service sets.

NOTE: You do not configure this rule; it is created by the key management process (kmd).

Rule lookup for static tunnels is unaffected by the presence of a dynamic rule; it is performed in the order configured. When a packet is received for a service set, static rules are always matched first.

Dynamic rules are matched after the rule match for static rules has failed.

Response to dead peer detection (DPD) hello messages takes place the same way with dynamic peers as with static peers. Initiating DPD hello messages from dynamic peers is not supported. For more information on DPD, see "Configuring Destination Addresses for Dead Peer Detection" on page 35.

Reverse Route Insertion

Static routes are automatically inserted into the route table for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created based on the remote proxy network and mask sent by the peer and is inserted in the relevant route table after successful phase 1 and phase 2 negotiations.

The route preference for each static reverse route is 1. This value is necessary to avoid conflict with similar routes that might be added by the routing protocol process (rpd).

No routes are added if the accepted remote proxy address is the default (**0.0.0.0/0**). In this case you can run routing protocols over the IPsec tunnel to learn routes and add static routes for the traffic you want to be protected over this tunnel.

For next-hop-style service sets, the reverse routes include next hops pointing to the locations specified by the **inside-service-interface** statement.

The route table in which to insert these routes depends on where the **inside-service-interface** location is listed. If these interfaces are present in a VPN routing and forwarding (VRF) instance, then routes are added to the corresponding VRF table; otherwise, the routes are added to **inet.0**.

NOTE: Reverse route insertion takes place only for tunnels to dynamic peers. These routes are added only for next-hop-style service sets.

Configuring an IKE Access Profile

You can configure only one tunnel profile per service set for all dynamic peers. The configured preshared key in the profile is used for IKE authentication of all dynamic peers terminating in that service set. Alternatively, you can include the **ike-policy** statement to reference an IKE policy you define with either specific identification values or a wildcard (the **any-remote-id** option). You configure the IKE policy at the **[edit services ipsec-vpn ike]** hierarchy level; for more information, see "Configuring IKE Policies" on page 20.

The IKE tunnel profile specifies all the information needed to complete the IKE negotiation. Each protocol has its own statement hierarchy within the client statement to configure protocol-specific attribute value pairs, but only one client configuration is allowed for each profile. The following is the configuration at the **[edit access]** hierarchy level; for more information on access profiles, see the Junos OS System Basics Configuration Guide.

```
[edit access]
profile profile-name {
   client * {
      ike {
         allowed-proxy-pair {
            remote remote-proxy-address local local-proxy-address;
         }
         pre-shared-key (ascii-text key-string | hexadecimal key-string);
         ike-policy policy-name;
         interface-id <string-value>;
         ipsec-policy ipsec-policy;
      }
   }
}
```

NOTE: For dynamic peers, the Junos OS supports the IKE main mode with either the preshared key method of authentication or an IKE access profile that uses a local digital certificate.

- In preshared key mode, the IP address is used to identify a tunnel peer to get the preshared key information. The client value * (wildcard) means that configuration within this profile is valid for all dynamic peers terminating within the service set accessing this profile.
- In digital certificate mode, the IKE policy defines which remote identification values are allowed; for more information, see "Configuring IKE Policies" on page 20.

The following statements make up the IKE profile:

- **allowed-proxy-pair**—During phase 2 IKE negotiation, the remote peer supplies its network address (**remote**) and its peer's network address (**local**). Since multiple dynamic tunnels are authenticated through the same mechanism, this statement must include the list of possible combinations. If the dynamic peer does not present a valid combination, the phase 2 IKE negotiation fails.

  By default, **remote 0.0.0.0/0 local 0.0.0.0/0** is used if no values are configured. Both IPv4 and IPv6 address formats are supported in this configuration, but there are no default IPv6 addresses. You must specify even **0::0/0**.

- **pre-shared-key**—Key used to authenticate the dynamic peer during IKE phase 1 negotiation. This key is known to both ends through an out-of-band secure mechanism. You can configure the value either in **hexadecimal** or **ascii-text** format. It is a mandatory value.
- **ike-policy**—Policy that defines the remote identification values corresponding to the allowed dynamic peers; can contain a wildcard value **any-remote-id** for use in dynamic endpoint configurations only.
- **interface-id**—Interface identifier, a mandatory attribute used to derive the logical service interface information for the session.
- **ipsec-policy**—Name of the IPsec policy that defines the IPsec policy information for the session. You define the IPsec policy at the **[edit services ipsec-vpn ipsec policy *policy-name*]** hierarchy level. If no policy is set, any policy proposed by the dynamic peer is accepted.

Referencing the IKE Access Profile in a Service Set

To complete the configuration, you need to reference the IKE access profile configured at the **[edit access]** hierarchy level. To do this, include the **ike-access-profile** statement at the **[edit services service-set *name* ipsec-vpn-options]** hierarchy level:

```
[edit services service-set name]
ipsec-vpn-options {
   local-gateway address;
   ike-access-profile profile-name;
```

```
}
next-hop-service {
  inside-service-interface interface-name;
  outside-service-interface interface-name;
}
```

The **ike-access-profile** statement must reference the same name as the **profile** statement you configured for IKE access at the **[edit access]** hierarchy level. You can reference only one access profile in each service set. This profile is used to negotiate IKE and IPsec security associations with dynamic peers only.

NOTE: If you configure an IKE access profile in a service set, no other service set can share the same **local-gateway** address.

Also, you must configure a separate service set for each VRF instance. All interfaces referenced by the **ipsec-inside-interface** statement within a service set must belong to the same VRF instance.

Configuring the Interface Identifier

You can configure an interface identifier for a group of dynamic peers, which specifies which adaptive services logical interface(s) take part in the dynamic IPsec negotiation. By assigning the same interface identifier to multiple logical interfaces, you can create a pool of interfaces for this purpose. To configure an interface identifier, include the **ipsec-interface-id** statement and the **dedicated** or **shared** statement at the **[edit interfaces** *interface-name* **unit** *logical-unit-number* **dial-options]** hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number dial-options]
ipsec-interface-id identifier;
(dedicated | shared);
```

Specifying the interface identifier in the **dial-options** statement makes this logical interface part of the pool identified by the **ipsec-interface-id** statement.

NOTE: Only one interface identifier can be specified at a time. You can include the **ipsec-interface-id** statement or the **l2tp-interface-id** statement, but not both.

If you configure **shared** mode, it enables one logical interface to be shared across multiple tunnels. The **dedicated** statement specifies that the logical interface is used in a dedicated mode, which is necessary when you are configuring an IPsec link-type tunnel. You must include the **dedicated** statement when you specify an **ipsec-interface-id** value.

Default IKE and IPsec Proposals

The software includes implicit default IKE and IPsec proposals to match the proposals sent by the dynamic peers. The values are shown in Table 4 on page 43; if more than one value is shown, the first value is the default. For more information on IKE proposals, see

"Configuring IKE Proposals" on page 17; for more information on IPsec proposals, see "Configuring IPsec Proposals" on page 26.

NOTE: RSA certificates are not supported with dynamic endpoint configuration.

Table 4: Default IKE and IPsec Proposals for Dynamic Negotiations

Statement Name	Values
Implicit IKE Proposal	
authentication-method	pre-shared keys
dh-group	group1, group2, group5, group14
authentication-algorithm	sha1, md5, sha-256
encryption-algorithm	3des-cbc, des-cbc, aes-128, aes-192, aes-256
lifetime-seconds	3600 seconds
Implicit IPsec Proposal	
protocol	esp, ah, bundle
authentication-algorithm	hmac-sha1-96, hmac-md5-96
encryption-algorithm	3des-cbc, des-cbc, aes-128, aes-192, aes-256
lifetime-seconds	28,800 seconds (8 hours)

Tracing IPsec Operations

Trace operations track IPsec events and record them in a log file in the **/var/log** directory. By default, this file is named **/var/log/kmd**.

To trace IPsec operations, include the **traceoptions** statement at the **[edit services ipsec-vpn]** hierarchy level:

```
[edit services ipsec-vpn]
traceoptions {
    file <filename> <files number> <match regular-expression> <size bytes> <world-readable |
        no-world-readable>;
    flag flag;
    level level;
    no-remote-trace;
}
```

You can specify the following IPsec tracing flags:

- **all**—Trace everything.
- **certificates**—Trace certificates 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.
- **snmp**—Trace SNMP operations.
- **timer**—Trace internal timer events.

The **level** statement sets the key management process (kmd) tracing level. The following values are supported:

- **all**—Match all levels.
- **error**—Match error conditions.
- **info**–Match informational messages.
- **notice**—Match conditions that should be handled specially.
- **verbose**—Match verbose messages.
- **warning**—Match warning messages.

Disabling IPsec Tunnel Endpoint in Traceroute

If you include the **no-ipsec-tunnel-in-traceroute** statement at the **[edit services ipsec-vpn]** hierarchy level, the IPsec tunnel is not treated as a next hop and TTL is not decremented. Also, if the TTL reaches zero, an ICMP time exceeded message is not generated.

```
[edit services ipsec-vpn]
no-ipsec-tunnel-in-traceroute;
```

NOTE: This functionality is also provided by the **passive-mode-tunneling** statement described in Configuring IPsec Service Sets. You can use the **no-ipsec-tunnel-in-traceroute** statement in specific scenarios in which the IPsec tunnel should not be treated as a next hop and passive mode is not desired.

Tracing IPsec PKI Operations

Trace operations track IPsec PKI events and record them in a log file in the **/var/log** directory. By default, this file is named **/var/log/pkid**.

To trace IPsec PKI operations, include the **traceoptions** statement at the **[edit security pki]** hierarchy level:

```
[edit security pki]
traceoptions {
  file filename <files number> <match regular-expression> <size maximum-file-size>
    <world-readable | no-world-readable>;
  flag flag (all | certificate-verification | enrollment | online-crl-check);
}
```

You can specify the following PKI tracing flags:

- **all**—Trace everything.
- **certificates**—Trace certificates 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.
- **snmp**—Trace SNMP operations.
- **timer**—Trace internal timer events.

Configuring IPSec on the Services SDK

Starting with Junos OS Release 11.4, IPSec is supported by the Services SDK. IPSec on the Services SDK is supported on all M Series, T Series and MX Series routers with Multiservices 100, Multiservices 400 PICs, and Multiservices DPCs.

IPSec on the Services SDK has the following limitations:

- IPSec on the Services SDK supports only policies negotiated between dynamic peer security gateways in which the remote ends of tunnels do not have a statically assigned IP address (Dynamic Endpoints).
- Encapsulating Security Payload (ESP) is the only protocol that is supported for protecting IP traffic.
- IPSec on the Services SDK does not support IPv6.

To enable IPSec for the Services SDK on the adaptive services interface, configure the **object-cache-size**, **policy-db-size**, and **package** statements at the **[edit chassis fpc *slot-number* pic *pic-number* adaptive-services service-package extension-provider]** hierarchy level. For the IPSec plugin on the Services SDK, ***package-name*** in the **package *package-name*** statement is **jservices-ipsec**.

For more information about the Services SDK, see the *SDK Applications Configuration Guide and Command Reference*.

The following example shows how to enable IPSec for the Services SDK on the adaptive services interface:

```
chassis fpc 1 {
  pic 2 {
    adaptive-services {
      service-package {
        extension-provider {
          control-cores 1;
          data-cores 7;
          object-cache-size 1280;
          policy-db-size 64;
          package jservices-crypto-base;
          package jservices-ipsec;
        }
      }
    }
  }
}
```

Configure the inside and outside interfaces for next-hop-style service sets:

```
service-set abc {
  next-hop-service {
    inside-service-interface ms-0/2/0.1; # Name and logical unit number of the service
      interface associated with the service set applied inside the network.
    outside-service-interface ms-0/2/0.2; # Name and logical unit number of the service
      interface associated with the service set applied outside the network.
  }
}
```

CHAPTER 4

Examples

Example: Configuring Statically Assigned Tunnels

Following is the configuration of the provider edge (PE) router, demonstrating the usage of next-hop service sets and dynamic SA configuration:

```
[edit interfaces]
so-0/0/0 {
    no-keepalives;
    encapsulation cisco-hdlc;
    unit 0 {
        family inet {
            address 10.6.6.6/32;
        }
    }
}
so-2/2/0 {
    description "teller so-0/2/0";
    no-keepalives;
    encapsulation cisco-hdlc;
    unit 0 {
        family inet {
            address 10.21.1.1/16;
        }
    }
}
sp-3/1/0 {
    unit 0 {
        family inet {
            address 10.7.7.7/32;
        }
    }
    unit 1 {
        family inet;
        service-domain inside;
    }
    unit 2 {
        family inet;
        service-domain outside;
```

```
    }
}
[edit policy-options]
policy-statement vpn-export {
    then {
        community add vpn-comm;
        accept;
    }
}
policy-statement vpn-import {
    term a {
        from community vpn-comm;
        then accept;
    }
}
community vpn-comm members target:100:20;
[edit routing-instances]
vrf {
    instance-type vrf;
    interface sp-3/1/0.1; # Inside sp interface
    interface so-0/0/0.0;
    route-distinguisher 192.168.0.1:1;
    vrf-import vpn-import;
    vrf-export vpn-export;
    routing-options {
        static {
            route 10.0.0.0/0 next-hop so-0/0/0.0;
            route 10.11.11.1/32 next-hop so-0/0/0.0;
            route 10.8.8.1/32 next-hop sp-3/1/0.1;
        }
    }
}
[edit services]
ipsec-vpn {
    rule rule-1 {
        term term-1 {
            then {
                remote-gateway 10.21.2.1;
                dynamic {
                    ike-policy ike-policy;
                }
            }
        }
        match-direction input;
    }
    ike {
        policy ike-policy {
            pre-shared-key ascii-text "$9$ExmcSeMWxdVYBI";
        }
    }
}
service-set service-set-1 {
    ipsec-vpn {
        local-gateway 10.21.1.1;
    }
    ipsec-vpn-rules rule-1;
```

```
    next-hop-service {
      inside-service-interface sp-3/1/0.1;
      outside-service-interface sp-3/1/0.2;
    }
}
```

Following is an example for configuring multiple link-type tunnels to static peers using a single next-hop style service set:

```
services ipsec-vpn {
  rule demo-rule {
    term term-0 {
      from {
        ipsec-inside-interface sp-0/0/0.1;
      }
      then {
        remote-gateway 10.2.2.2;
        dynamic {
          ike-policy demo-ike-policy;
        }
      }
    }
    term term-1 {
      from {
        ipsec-inside-interface sp-0/0/0.3;
      }
      then {
        remote-gateway 10.3.3.3;
        dynamic {
          ike-policy demo-ike-policy;
        }
      }
    }
  }
  match-direction input;
}
services {
  service-set demo-service-set {
    next-hop-service {
      inside-service-interface sp-0/0/0.1;
      outside-service-interface sp-0/0/0.2;
    }
    ipsec-vpn-options {
      local-gateway 10.1.1.1;
    }
    ipsec-rules demo-rule;
  }
}
interfaces sp-0/0/0 {
  unit 0 {
    family inet;
  }
  unit 1 {
    family inet;
    service-domain inside;
  }
```

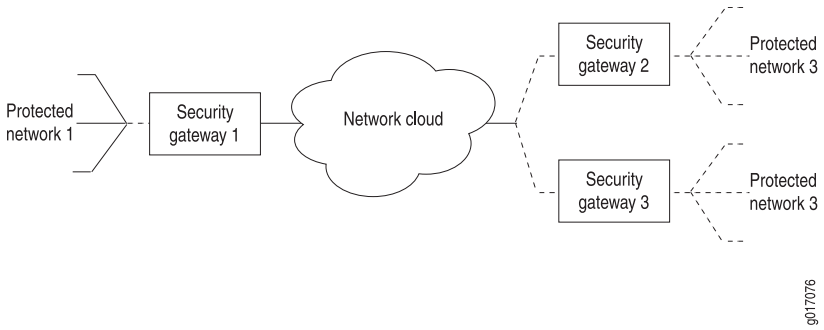
```
    unit 2 {
      family inet;
      service-domain outside;
    }
    unit 3 {
      family inet;
      service-domain inside;
    }
    unit 4 {
      family inet;
      service-domain inside;
    }
  }
```

Example: Configuring Dynamically Assigned Tunnels

The following examples are based on this network configuration (see Figure 1 on page 50):

- A local network N-1 behind security gateway SG-1, a Juniper Networks router terminating static as well as dynamic peer endpoints. The tunnel termination address on SG-1 is 10.1.1.1 and the local network address is 172.16.1.0/24.
- Two remote peer routers that obtain addresses from an ISP pool and run RFC-compliant IKE. Remote network N-2 has address 172.16.2.0/24 and resides behind security gateway SG-2 with tunnel termination address 10.2.2.2. Remote network N-3 has address 172.16.3.0/24 and resides behind security gateway SG-3 with tunnel termination address 10.3.3.3.

Figure 1: IPsec Dynamic Endpoint Tunneling Topology



The examples in this section show the following configurations:

- Configuring a Next-Hop Style Service Set with Link-Type Tunnels on page 50
- Configuring a Next-Hop Style Service-Set with Policy-Based Tunnels on page 52

NOTE: All the configurations are given for the Juniper Networks router terminating dynamic endpoint connections.

Configuring a Next-Hop Style Service

```
access {
  profile demo-access-profile client * {
```

Set with Link-Type Tunnels

```
    ike {
      allowed-proxy-pair {
        remote 0.0.0.0/0 local 0.0.0.0/0; # ANY to ANY
      }
      pre-shared-key {
        ascii-text keyfordynamicpeers;
      }
      interface-id demo-ipsec-interface-id;
    }
  }
  services {
    service-set demo-service-set {
      next-hop-service {
        inside-service-interface sp-1/0/0.1;
        outside-service-interface sp-1/0/0.2;
      }
      ipsec-vpn-options {
        local-gateway 10.1.1.1;
        ike-access-profile demo-ike-access-profile;
      }
    }
  }
}
```

NOTE: Including the **ike-access-profile** statement enables the software to incorporate implicit proposals for dynamic endpoint authentication. You do not need to configure IKE or IPsec proposals explicitly.

```
interfaces {
  sp-0/0/0 {
    unit 0 {
      family inet;
    }
    unit 1 {
      family inet;
      service-domain inside;
    }
    unit 2 {
      family inet;
      service-domain outside;
    }
    unit 3 {
      family inet;
      service-domain inside;
      dial-options {
        ipsec-interface-id demo-ipsec-interface-id;
        dedicated;
      }
    }
    unit 4 {
      family inet;
      service-domain inside;
      dial-options {
```

```
            ipsec-interface-id demo-ipsec-interface-id;
            dedicated;
          }
        }
      }
    }
```

The following results are obtained:

- Reverse routes inserted after successful negotiation:

  None

- Routes learned by routing protocol:

  172.16.2.0/24

  172.16.3.0/24

- Dynamic implicit rules created after successful negotiation:

```
rule: junos-dynamic-rule-0
term: term-0
  local-gateway-address : 10.1.1.1 #Tunnel termination address on SG-1
  remote-gateway-address: 10.2.2.2 #Tunnel termination address on SG-2
  source-address : 0.0.0.0/0
  destination-address : 0.0.0.0/0
ipsec-inside-interface: sp-0/0/0.3
term: term-1
  local-gateway-address : 10.1.1.1 #Tunnel termination address on SG-1
  remote-gateway-address: 10.3.3.3 #Tunnel termination address on SG-3
  source-address : 0.0.0.0/0
  destination-address : 0.0.0.0/0
ipsec-inside-interface: sp-0/0/0.4
match-direction: input
```

Configuring a Next-Hop Style Service-Set with Policy-Based Tunnels

```
access {
  profile demo-access-profile client * {
    ike {
      allowed-proxy-pair {
        remote 172.16.2.0/24 local 172.16.1.0/24; #N-2 <==> #N-1
        remote 172.16.3.0/24 local 172.16.1.0/24; #N-3 <==> #N-1
      }
      pre-shared-key {
        ascii-text keyfordynamicpeers;
      }
      interface-id demo-ipsec-interface-id;
    }
  }
}
services {
  service-set demo-service-set {
    next-hop-service {
      inside-service-interface sp-1/0/0.1;
      outside-service-interface sp-1/0/0.2;
    }
    ipsec-vpn-options {
      local-gateway 10.1.1.1;
```

```
    }
    ike-access-profile demo-ike-access-profile;
  }
}
```

NOTE: Including the **ike-access-profile** statement enables the software to incorporate implicit proposals for dynamic endpoint authentication. You do not need to configure IKE or IPsec proposals explicitly.

```
interfaces {
  sp-0/0/0 {
    unit 0 {
      family inet;
    }
    unit 1 {
      family inet;
      service-domain inside;
    }
    unit 2 {
      family inet;
      service-domain outside;
    }
    unit 3 {
      family inet;
      service-domain inside;
      dial-options {
        ipsec-interface-id demo-ipsec-interface-id;
        mode shared;
      }
    }
  }
}
# VRF configuration, if not inet.0
routing-instances {
  demo-vrf {
    instance-type vrf;
    interface sp-0/0/0.1;
    interface sp-0/0/0.3;
    .....
  }
}
```

The following results are obtained:

- Reverse routes injected after successful negotiation:

```
demo-vrf.inet.0: .... # Routing instance
172.11.0.0/24 *[Static/1]..
  > via sp-0/0/0.3
  172.12.0.0/24 *[Static/1]..
> via sp-0/0/0.3
```

- Dynamic implicit rules created after successful negotiation:

```
rule: junos-dynamic-rule-0
```

```
term: term-0
  local-gateway-address : 10.1.1.1 #Tunnel termination address on SG-1
    remote-gateway-address: 10.2.2.2 #Tunnel termination address on SG-2
    source-address : 172.16.1.0/24
    destination-address : 172.16.2.0/24
  ipsec-inside-interface: sp-0/0/0.3
term: term-1
  local-gateway-address : 10.1.1.1 #Tunnel termination address on SG-1
    remote-gateway-address: 10.3.3.3 #Tunnel termination address on SG-3
    source-address : 172.16.1.0/24
    destination-address : 172.16.3.0/24
  ipsec-inside-interface: sp-0/0/0.3
match-direction: input
```

CHAPTER 5

Configuration Statements

anti-replay-window-size (Services IPsec VPN)

Syntax anti-replay-window-size *bits*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced in Junos OS Release 10.0.

Description Specify the size of the IPsec antireplay window.

Options ***bits***—Size of the antireplay window, in bits.
Default: 64 bits (AS PICs), 128 bits (Multiservices PICs and DPCs)
Range: 64 through 4096 bits

Usage Guidelines See "Configuring or Disabling IPsec Anti-Replay" on page 36.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

authentication (Services IPsec VPN)

Syntax

```
authentication {
  algorithm (hmac-md5-96 | hmac-sha1-96);
  key (ascii-text key | hexadecimal key);
}
```

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then manual direction *direction*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure IPsec authentication parameters for a manual security association (SA).

Options

algorithm—Hash algorithm that authenticates packet data. The algorithm can be one of the following:

- **hmac-md5-96**—Produces a 128-bit digest.
- **hmac-sha1-96**—Produces a 160-bit digest.

key—Type of authentication key. The key 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.

Usage Guidelines See "Configuring Authentication for a Manual IPsec SA" on page 14.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

authentication-algorithm (Services IKE)

Syntax authentication-algorithm (md5 | sha1 | sha-256);

Hierarchy Level [edit services ipsec-vpn ike proposal *proposal-name*]

Release Information Statement introduced before Junos OS Release 7.4.
sha-256 option added in Junos OS Release 7.6.

Description Configure the Internet Key Exchange (IKE) hash algorithm that authenticates packet data.

Options **md5**—Produces a 128-bit digest.

sha1—Produces a 160-bit digest.

sha-256—Produces a 256-bit digest.

Usage Guidelines See "Configuring the Authentication Algorithm for an IKE Proposal" on page 18.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

authentication-algorithm (Services IPsec)

Syntax authentication-algorithm (hmac-md5-96 | hmac-sha1-96);

Hierarchy Level [edit services ipsec-vpn ipsec proposal *ipsec-proposal-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the IPsec hash algorithm that authenticates packet data.

Options **hmac-md5-96**—Produces a 128-bit digest.

hmac-sha1-96—Produces a 160-bit digest.

Usage Guidelines See "Configuring the Authentication Algorithm for an IPsec Proposal" on page 26.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

authentication-method (Services IPsec VPN)

Syntax authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);

Hierarchy Level [edit services ipsec-vpn ike proposal *proposal-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure an IKE authentication method.

Options **dsa-signatures**—Digital signature algorithm (DSA).

rsa-signatures—Public key algorithm (supports encryption and digital signatures).

pre-shared-keys—A key derived from an out-of-band mechanism; the key authenticates the exchange.

Usage Guidelines See "Configuring the Authentication Method for an IKE Proposal" on page 18.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

auxiliary-spi (Services IPsec VPN)

Syntax auxiliary-spi *spi-value*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then manual direction *direction*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure an auxiliary Security Parameter Index (SPI) for a manual SA. Use the auxiliary SPI when you configure the **protocol** statement to use the **bundle** option.

Options ***spi-value***—An arbitrary value that uniquely identifies which SA to use at the receiving host (the destination address in the packet).

Range: 256 through 16,639

Usage Guidelines See "Configuring the Auxiliary Security Parameter Index" on page 14. For information about SPI, see "Configuring the Security Parameter Index" on page 14 and **spi**.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

backup-remote-gateway

Syntax backup-remote-gateway *address*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the backup remote address to which the IPsec traffic is directed when the primary remote gateway is down. Configuring this statement also enables the dead peer detection (DPD) protocol.

Options ***address***—Backup remote IPv4 or IPv6 address.

Usage Guidelines See "Configuring Destination Addresses for Dead Peer Detection" on page 35.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

clear-dont-fragment-bit (Services IPsec VPN)

Syntax clear-dont-fragment-bit;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Clear the Don't Fragment (DF) bit on all IP version 4 (IPv4) packets entering the IPsec tunnel. If the encapsulated packet size exceeds the tunnel maximum transmission unit (MTU), the packet is fragmented before encapsulation.

Usage Guidelines See "Configuring Actions in IPsec Rules" on page 34.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

clear-ike-sas-on-pic-restart

Syntax clear-ike-sas-on-pic-restart;

Hierarchy Level [edit services ipsec-vpn]

Release Information Statement introduced in Junos OS Release 8.5.

Description Clear IKE security associations (SAs) when the corresponding PIC restarts or is taken offline.

Usage Guidelines See "Clearing Security Associations" on page 16.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

clear-ipsec-sas-on-pic-restart

Syntax clear-ipsec-sas-on-pic-restart;

Hierarchy Level [edit services ipsec-vpn]

Release Information Statement introduced in Junos OS Release 9.2.

Description Clear IPsec security associations (SAs) when the corresponding PIC restarts or is taken offline.

Usage Guidelines See "Clearing Security Associations" on page 16.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

description (Services IPsec VPN)

Syntax description *description*;

Hierarchy Level [edit services ipsec-vpn ike policy *policy-name*],
[edit services ipsec-vpn ike proposal *proposal-name*],
[edit services ipsec-vpn ipsec (Services IPsec VPN) policy *policy-name*],
[edit services ipsec-vpn ipsec (Services IPsec VPN) proposal *proposal-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the text description for an IKE or IPsec policy or proposal.

Usage Guidelines See "Configuring the Description for an IKE Policy" on page 24, "Configuring the Description for an IPsec Proposal" on page 27, and "Configuring the Description for an IPsec Policy" on page 29.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

destination-address (Services IPsec VPN)

Syntax destination-address *address*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* from]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the destination address for rule matching.

Options ***address***—Destination IP address.

Usage Guidelines See "Configuring Match Conditions in IPsec Rules" on page 32.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

dh-group

Syntax dh-group (group1 | group2 | group5 |group14);

Hierarchy Level [edit services ipsec-vpn ike proposal *proposal-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the IKE Diffie-Hellman prime modulus group to use for performing the new Diffie-Hellman exchange.

Options

group1—768-bit.

group2—1024-bit.

group5—1536-bit.

group14—2048-bit.

Usage Guidelines See "Configuring the Diffie-Hellman Group for an IKE Proposal" on page 18.

Required Privilege Level
admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

direction

Syntax

```
direction (inbound | outbound | bidirectional) {
  protocol (ah | bundle | esp);
  spi spi-value;
  auxiliary-spi spi-value;
  authentication (Services IPsec VPN) {
    algorithm (hmac-md5-96 | hmac-sha1-96);
    key (ascii-text key | hexadecimal key);
  }
  encryption {
    algorithm algorithm;
    key (ascii-text key | hexadecimal key);
  }
}
```

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then manual]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the direction in which manual SAs are applied.

Options

bidirectional—Apply the SA in both directions.

inbound—Apply the SA on inbound traffic.

outbound—Apply the SA on outbound traffic.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Match Direction for IPsec Rules" on page 32.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

dynamic

Syntax

```
dynamic {
  ike-policy policy-name;
  ipsec-policy policy-name;
}
```

Hierarchy Level [edit services (IPsec VPN) ipsec-vpn rule (Services IPsec VPN) *rule-name* term (Services IPsec VPN) *term-name* then (Services IPsec VPN)]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define a dynamic IPsec SA.

Options

ike-policy *policy-name*—Name of the IKE policy. This statement is optional for the non-preshared-key authentication method. For digital signature-based authentication, this statement is optional and the default policy is used if none is supplied.

ipsec-policy *policy-name*—Name of the IPsec policy. This statement is optional and the default policy is used if none is supplied.

Usage Guidelines See "Configuring Dynamic Security Associations" on page 16.

Required Privilege Level

admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

encryption

Syntax

```
encryption {
  algorithm algorithm;
  key (ascii-text key | hexadecimal key);
}
```

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then manual direction *direction*]

Release Information Statement introduced before Junos OS Release 7.4.
aes-128-cbc, **aes-192-cbc**, and **aes-256-cbc** options added in Junos OS Release 7.6.

Description Configure an encryption algorithm and key for manual SA.

Options **algorithm**—Type of encryption algorithm. The algorithm can be one of the following:

- **des-cbc**—Has a block size of 8 bytes (64 bits); the key size is 48 bits long.
- **3des-cbc**—Has a block size of 8 bytes (64 bits); the key size is 192 bits long.
- **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.

NOTE: For 3des-cbc, the first 8 bytes should differ from the second 8 bytes, and the second 8 bytes should be the same as the third 8 bytes.

key—Type of encryption key. The key can be one of the following:

- **ascii-text**—ASCII text key. Following are the key lengths, in ASCII characters, for the different encryption options:
  - **des-cbc** option, 8 ASCII characters
  - **3des-cbc** option, 24 ASCII characters
  - **aes-128-cbc** option, 16 ASCII characters
  - **aes-192-cbc** option, 24 ASCII characters
  - **aes-256-cbc** option, 32 ASCII characters
- **hexadecimal**—Hexadecimal key. Following are the key lengths, in hexadecimal characters, for the different encryption options:
  - **des-cbc** option, 16 hexadecimal characters
  - **3des-cbc** option, 48 hexadecimal characters
  - **aes-128-cbc** option, 32 hexadecimal characters
  - **aes-192-cbc** option, 48 hexadecimal characters

- **aes-256-cbc** option, 64 hexadecimal characters

Usage Guidelines See "Configuring Encryption for a Manual IPsec SA" on page 15.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

encryption-algorithm (Services IPsec VPN)

Syntax encryption-algorithm *algorithm*;

Hierarchy Level [edit services ipsec-vpn ike proposal *proposal-name*],
[edit services ipsec-vpn ipsec proposal *proposal-name*]

Release Information Statement introduced before Junos OS Release 7.4.
aes-128-cbc, **aes-192-cbc**, and **aes-256-cbc** options added in Junos OS Release 7.6.

Description Configure an IKE or IPsec encryption algorithm.

Options **3des-cbc**—Has a block size of 24 bytes; the key size is 192 bits long.

des-cbc—Has a block size of 8 bytes; the key size is 48 bits long.

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.

Usage Guidelines See "Configuring the Encryption Algorithm for an IKE Proposal" on page 19 and "Configuring the Encryption Algorithm for an IPsec Proposal" on page 27.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

from (Services IPsec VPN)

Syntax

```
from {
  destination-address address;
  ipsec-inside-interface interface-name;
  source-address address;
}
```

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify input conditions for the IPsec term.

Options For information on match conditions, see the description of firewall filter match conditions in the Routing Policy Configuration Guide.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Match Direction for IPsec Rules" on page 32.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

ike

Syntax

```
ike {
  proposal proposal-name {
    authentication-algorithm (md5 | sha1 | sha-256);
    authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
    description description;
    dh-group (group1 | group2 | group5 |group14);
    encryption-algorithm algorithm;
    lifetime-seconds seconds;
  }
  policy policy-name {
    description description;
    local-certificate identifier;
    local-id (ipv4_addr ipv4-address | ipv6-addr ipv6-address | key-id identifier);
    version (1 | 2);
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposals [ proposal-names ];
    remote-id {
      any-remote-id;
      ipv4_addr [ values ];
      ipv6_addr [ values ];
      key_id [ values ];
    }
  }
}
```

Hierarchy Level [edit services ipsec-vpn]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure IKE.

The statements are explained separately.

Usage Guidelines See "Configuring IKE Proposals" on page 17 and "Configuring IKE Policies" on page 20.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

initiate-dead-peer-detection

Syntax initiate-dead-peer-detection;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced in Junos OS Release 9.2.

Description Enable triggering of dead peer detection (DPD) Hello messages to the remote peer for the specified tunnel.

Usage Guidelines See "Configuring Destination Addresses for Dead Peer Detection" on page 35.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- backup-remote-gateway on page 59

ipsec (Services IPsec VPN)

Syntax

```
ipsec {
  proposal proposal-name {
    authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
    description description;
    encryption-algorithm algorithm;
    lifetime-seconds seconds;
    protocol (ah | esp | bundle);
  }
  policy policy-name {
    description description;
    perfect-forward-secrecy {
      keys (group1 | group2);
    }
    proposals [ proposal-names ];
  }
}
```

Hierarchy Level [edit services ipsec-vpn]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure IPsec.

The statements are explained separately.

Usage Guidelines See "Configuring Security Associations" on page 11.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

ipsec-inside-interface

Syntax ipsec-inside-interface *interface-name*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 7.4.

Description Specify the interface name for next-hop-style service sets. This value is also implicitly generated in dynamic endpoint tunneling.

Options ***interface-name***—Service interface for internal network.

Usage Guidelines See "Configuring Match Conditions in IPsec Rules" on page 32 or "Configuring Dynamic Endpoints for IPsec Tunnels" on page 38.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

lifetime-seconds (Services IPsec VPN)

Syntax lifetime-seconds *seconds*;

Hierarchy Level [edit services ipsec-vpn ike proposal *proposal-name*],
[edit services ipsec-vpn ipsec proposal *proposal-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the lifetime of an IKE or IPsec SA. This statement is optional.

Options ***seconds***—Lifetime

Default: 3600 seconds (IKE); 28,800 seconds (IPsec)

Range: 180 through 86,400

Usage Guidelines See "Configuring the Lifetime for an IKE SA" on page 20 and "Configuring the Lifetime for an IPsec SA" on page 27.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

local-certificate (Services IPsec VPN)

Syntax local-certificate *identifier*;

Hierarchy Level [edit services ipsec-vpn ike policy *policy-name*]

Release Information Statement introduced in Junos OS Release 7.5.

Description Name of the certificate that needs to be sent to the peer during the IKE authentication phase.

Options ***identifier***—Name of certificate.

Usage Guidelines See "Configuring the Local Certificate for an IKE Policy" on page 23.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

local-id

Syntax local-id (ipv4_addr *ipv4-address* | ipv6-addr *ipv6-address* | key-id *identifier*);

Hierarchy Level [edit services ipsec-vpn ike policy *policy-name*]

Release Information Statement introduced before Junos OS Release 7.4.
ipv6_addr option added in Junos OS Release 7.6.

Description Specify local identifiers for IKE Phase 1 negotiation. This statement is optional.

Options **ipv4_addr** ***ipv4-address***—IPv4 address identification value.

ipv6_addr ***ipv6-address***—IPv6 address identification value.

key_id ***identifier***—Key identification value.

Usage Guidelines See "Configuring Local and Remote IDs for IKE Phase 1 Negotiation" on page 24.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

manual

Syntax

```
manual {
  direction (inbound | outbound | bidirectional) {
    authentication {
      algorithm (hmac-md5-96 | hmac-sha1-96);
      key (ascii-text key | hexadecimal key);
    }
    auxiliary-spi spi-value;
    encryption {
      algorithm algorithm;
      key (ascii-text key | hexadecimal key);
    }
    spi spi-value;
    protocol (ah | esp | bundle);
  }
}
```

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define a manual IPsec SA.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Manual Security Associations" on page 12.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

match-direction (Services IPsec VPN)

Syntax match-direction (input | output);

Hierarchy Level [edit services ipsec-vpn rule *rule-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the direction in which the rule match is applied.

Options **input**—Apply the rule match on input.

output—Apply the rule match on output.

Usage Guidelines See "Configuring Match Direction for IPsec Rules" on page 32.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

mode (Services IPsec VPN)

Syntax mode (aggressive | main);

Hierarchy Level [edit services ipsec-vpn ike policy *policy-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an IKE policy mode.

Default main

Options
- **aggressive**—Takes half the number of messages of main mode, has less negotiation power, and does not provide identity protection.
- **main**—Uses six messages, in three peer-to-peer exchanges, to establish the IKE SA. These three steps include the IKE SA negotiation, a Diffie-Hellman exchange, and authentication of the peer. Also provides identity protection.

Usage Guidelines See "Configuring the Mode for an IKE Policy" on page 22.

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

no-anti-replay (Services IPsec VPN)

Syntax no-anti-replay;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Disable IPsec antireplay service, which occasionally causes interoperability issues for security associations.

Usage Guidelines See "Configuring or Disabling IPsec Anti-Replay" on page 36.

Required Privilege Level
admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

no-ipsec-tunnel-in-traceroute

Syntax no-ipsec-tunnel-in-traceroute;

Hierarchy Level [edit services ipsec-vpn]

Release Information Statement introduced in Junos OS Release 10.0.

Description Disables displaying the IPsec tunnel endpoint in the trace route output. The IPsec tunnel is not treated as a next hop and TTL is not decremented. If the TTL becomes zero, the ICMP time exceeded message will not be generated.

Usage Guidelines See "Configuring or Disabling IPsec Anti-Replay" on page 36.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

perfect-forward-secrecy (Services IPsec VPN)

Syntax

```
perfect-forward-secrecy {
  keys (group1 | group2 |group5 |group14);
}
```

Hierarchy Level [edit services ipsec-vpn ipsec policy *policy-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define Perfect Forward Secrecy (PFS). Creates single-use keys. This statement is optional.

Options **keys**—Type of Diffie-Hellman prime modulus group that IKE uses when performing the new Diffie-Hellman exchange. The key can be one of the following:

- **group1**—768-bit.
- **group2**—1024-bit.
- **group5**—1536-bit.
- **group14**—2048-bit.

Usage Guidelines See "Configuring Perfect Forward Secrecy" on page 29.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

policy (Services IKE)

Syntax

```
policy policy-name {
    description description;
    local-certificate identifier;
    local-id (ipv4_addr ipv4-address | ipv6-addr ipv6-address | key-id identifier);
    version (1 | 2);
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposals [ proposal-names ];
    remote-id {
        any-remote-id;
        ipv4_addr [ values ];
        ipv6_addr [ values ];
        key_id [ values ];
    }
}
```

Hierarchy Level [edit services ipsec-vpn ike]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an IKE policy.

Options ***policy-name***—IKE policy name.

The remaining statements are explained separately.

Usage Guidelines See "Configuring IKE Policies" on page 20.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

policy (Services IPsec VPN)

Syntax

```
policy policy-name {
  description description;
  perfect-forward-secrecy {
    keys (group1 | group 14 | group2 | group 5);
  }
  proposals [ proposal-names ];
}
```

Hierarchy Level [edit services ipsec-vpn ipsec]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an IPsec policy.

Options *policy-name*—IPsec policy name.

The remaining statements are explained separately.

Usage Guidelines See "Configuring IPsec Policies" on page 28.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

pre-shared-key (Services IKE)

Syntax pre-shared-key (ascii-text *key* | hexadecimal *key*);

Hierarchy Level [edit services ike policy *policy-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define a preshared key for an IKE policy.

Options *key*—Value of preshared key. The key can be one of the following:

- **ascii-text**—ASCII text key.
- **hexadecimal**—Hexadecimal key.

Usage Guidelines See "Configuring IKE Policies" on page 20.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

proposal (Services IKE)

Syntax

```
proposal proposal-name {
  authentication-algorithm (md5 | sha1 | sha-256);
  authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
  description description;
  dh-group (group1 | group2 | group5 |group14);
  encryption-algorithm algorithm;
  lifetime-seconds seconds;
}
```

Hierarchy Level [edit services ipsec-vpn ike]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an IKE proposal for a dynamic SA.

Options *proposal-name*—IKE proposal name.

The remaining statements are explained separately.

Usage Guidelines See "Configuring IKE Proposals" on page 17.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

proposal (Services IPsec VPN)

Syntax

```
proposal proposal-name {
  authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
  description description;
  encryption-algorithm algorithm;
  lifetime-seconds seconds;
  protocol (ah | esp | bundle);
}
```

Hierarchy Level [edit services ipsec-vpn ipsec]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an IPsec proposal for a dynamic SA.

Options *proposal-name*—IPsec proposal name.

The remaining statements are explained separately.

Usage Guidelines See "Configuring IPsec Proposals" on page 26.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

proposals

Syntax proposals [*proposal-names*];

Hierarchy Level [edit services ipsec-vpn ike policy *policy-name*],
[edit services ipsec-vpn ipsec policy *policy-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define a list of proposals to include in the IKE or IPsec policy.

Options ***proposal-names***—List of IKE or IPsec proposal names.

Usage Guidelines See "Configuring the Proposals in an IKE Policy" on page 22 and "Configuring the Proposals in an IPsec Policy" on page 30.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

protocol

Syntax protocol (ah | esp | bundle);

Hierarchy Level [edit services ipsec-vpn ipsec proposal *proposal-name*],
[edit services ipsec-vpn rule *rule-name* term *term-name* then manual direction *direction*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an IPsec protocol for a dynamic or manual SA.

Options **ah**—Authentication Header protocol.

esp—Encapsulating Security Payload protocol.

bundle—AH and ESP protocol.

Usage Guidelines See "Configuring the Protocol for a Manual IPsec SA" on page 13.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

remote-gateway

Syntax remote-gateway *address*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the remote address to which the IPsec traffic is directed.

Options ***address***—Remote IPv4 or IPv6 address.

Usage Guidelines See "Configuring Actions in IPsec Rules" on page 34.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

remote-id

Syntax

```
remote-id {
  any-remote-id;
  ipv4_addr [ values ];
  ipv6_addr [ values ];
  key_id [ values ];
}
```

Hierarchy Level [edit services ipsec-vpn ikepolicy *policy-name*]

Release Information Statement introduced before Junos OS Release 7.4.
ipv6_addr option added in Junos OS Release 7.6.
any-remote-id option added in Junos OS Release 8.2.

Description Define the remote identification values to which the IKE policy applies.

Options **any-remote-id**—Allow any remote address to connect. This option is supported only in dynamic endpoints configurations and cannot be configured along with specific values.

ipv4_addr [*values*]—Define one or more IPv4 address identification values.

ipv6_addr [*values*]—Define one or more IPv6 address identification values.

key_id [*values*]—Define one or more key identification values.

Usage Guidelines See "Configuring Local and Remote IDs for IKE Phase 1 Negotiation" on page 24.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

rule (Services IPsec VPN)

Syntax

```
rule rule-name {
  match-direction (input | output);
  term term-name {
    from {
      destination-address address;
      ipsec-inside-interface interface-name;
      source-address address;
    }
    then {
      anti-replay-window-size bits;
      backup-remote-gateway address;
      clear-dont-fragment-bit;
      dynamic {
        ike-policy policy-name;
        ipsec-policy policy-name;
      }
      initiate-dead-peer-detection;
      manual {
        direction (inbound | outbound | bidirectional) {
          authentication {
            algorithm (hmac-md5-96 | hmac-sha1-96);
            key (ascii-text key | hexadecimal key);
          }
          auxiliary-spi spi-value;
          encryption {
            algorithm algorithm;
            key (ascii-text key | hexadecimal key);
          }
          protocol (ah | bundle | esp);
          spi spi-value;
        }
      }
      no-anti-replay;
      remote-gateway address;
      syslog;
      tunnel-mtu bytes;
    }
  }
}
```

Hierarchy Level

[edit services ipsec-vpn],
[edit services ipsec-vpn rule-set *rule-set-name*]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Specify the rule the router uses when applying this service.

Options

rule-name—Identifier for the collection of terms that comprise this rule.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Match Direction for IPsec Rules" on page 32.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

rule-set (Services IPsec VPN)

Syntax

```
rule-set rule-set-name {
    [ rule rule-names ];
}
```

Hierarchy Level [edit services ipsec-vpn]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the rule set the router uses when applying this service.

Options ***rule-set-name***—Identifier for the collection of rules that constitute this rule set.

Usage Guidelines See "Configuring IPsec Rule Sets" on page 37.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

services (IPsec VPN)

Syntax services ipsec-vpn { ... }

Hierarchy Level [edit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the service rules to be applied to traffic.

Options **ipsec-vpn**—IPsec set of rules statements.

Usage Guidelines See IPsec Properties.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

source-address (Services IPsec VPN)

Syntax source-address *address*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* from]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the source address for rule matching.

Options ***address***—Source IP address.

Usage Guidelines See "Configuring Match Conditions in IPsec Rules" on page 32.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

spi

Syntax spi *spi-value*;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then manual direction *direction*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the SPI for an SA.

Options ***spi-value***—An arbitrary value that uniquely identifies which SA to use at the receiving host (the destination address in the packet).

Range: 256 through 16,639

NOTE: Use the auxiliary SPI when you configure the protocol statement to use the bundle option.

Usage Guidelines See "Configuring the Security Parameter Index" on page 14.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

syslog (Services IPsec VPN)

Syntax syslog;

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Enable system logging. The system log information for the Adaptive Services or Multiservices Physical Interface Card (PIC) is passed to the kernel for logging in the **/var/log** directory.

Usage Guidelines See "Configuring Actions in IPsec Rules" on page 34.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

term (Services IPsec VPN)

Syntax

```
term term-name {
  from {
    destination-address address;
    ipsec-inside-interface interface-name;
    source-address address;
  }
  then {
    anti-replay-window-size bits;
    backup-remote-gateway address;
    clear-dont-fragment-bit;
    dynamic {
      ike-policy policy-name;
      ipsec-policy policy-name;
    }
    initiate-dead-peer-detection;
    manual {
      direction (inbound | outbound | bidirectional) {
        authentication {
          algorithm (hmac-md5-96 | hmac-sha1-96);
          key (ascii-text key | hexadecimal key);
        }
        auxiliary-spi spi-value;
        encryption {
          algorithm algorithm;
          key (ascii-text key | hexadecimal key);
        }
        protocol (ah | bundle | esp);
        spi spi-value;
      }
    }
    no-anti-replay;
    remote-gateway address;
    syslog;
    tunnel-mtu bytes;
  }
}
```

Hierarchy Level [edit services ipsec-vpn rule *rule-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the IPsec term properties.

Options *term-name*—Identifier for the term.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Match Direction for IPsec Rules" on page 32.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

then (Services IPsec VPN)

Syntax

```
then {
  anti-replay-window-size bits;
  backup-remote-gateway address;
  clear-dont-fragment-bit;
  dynamic {
    ike-policy policy-name;
    ipsec-policy policy-name;
  }
  initiate-dead-peer-detection;
  manual {
    direction (inbound | outbound | bidirectional) {
      authentication {
        algorithm (hmac-md5-96 | hmac-sha1-96);
        key (ascii-text key | hexadecimal key);
      }
      auxiliary-spi spi-value;
      encryption {
        algorithm algorithm;
        key (ascii-text key | hexadecimal key);
      }
      protocol (ah | bundle | esp);
      spi spi-value;
    }
  }
  no-anti-replay;
  remote-gateway address;
  syslog;
  tunnel-mtu bytes;
}
```

Hierarchy Level [edit services ipsec-vpn rule *rule-name* term *term-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the IPsec term actions.

Options The remaining statements are explained separately.

Usage Guidelines See "Configuring Match Direction for IPsec Rules" on page 32.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

traceoptions (Services IPsec VPN)

Syntax

```
traceoptions {
  file <filename> <files number> <match regular-expression> <size bytes> <world-readable |
    no-world-readable>;
  flag flag;
  level level;
  no-remote-trace;
}
```

Hierarchy Level [edit services ipsec-vpn]

Release Information Statement introduced in Junos OS Release 7.5.
level option added in Junos OS Release 10.0.

Description Configure IPsec tracing operations. By default, messages are written to **/var/log/kmd**.

Options **files *number***—Maximum number of trace data files.

Range: 2 through 1000

flag *flag*—Tracing operation to perform:

- **all**—Trace everything.
- **certificates**—Trace certificates that apply to the IPsec service set.
- **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.
- **snmp**—Trace SNMP operations.
- **timer**—Trace internal timer events.

level *level*—Key management process (kmd) tracing level. The following values are supported:

- **all**—Match all levels.
- **error**—Match error conditions.
- **info**–Match informational messages.
- **notice**—Match conditions that should be handled specially.
- **verbose**—Match verbose messages.
- **warning**—Match warning messages.

size *bytes*—Maximum trace file size.

Usage Guidelines See "Tracing IPsec Operations" on page 43.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

traceoptions (PKI)

Syntax

```
traceoptions {
  file filename <files number> <match regular-expression> <size maximum-file-size>
    <world-readable | no-world-readable>;
  flag flag;
}
```

Hierarchy Level [edit security pki]

Description Configure security public key infrastructure (PKI) trace options. To specify more than one trace option, include multiple **flag** statements. Trace option output is recorded in the **/var/log/pkid** file.

Options

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. To include the **file** statement, you must specify a filename.

files *number*—(Optional) Maximum number of trace files. When a trace file (for example, **pkid**) reaches its maximum size, it is renamed **pkid.0**, then **pkid.1**, and so on, until the maximum number of trace files is reached. When the maximum number is reached, the oldest trace file is overwritten. If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 2 files

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.

enrollment—PKI certificate enrollment tracing.

match *regular-expression*—(Optional) Refine the output to include lines that contain the regular expression.

size *maximum-file-size*—(Optional) Maximum size of each trace file, in kilobytes (KB). If you specify a maximum file size, you also must specify a maximum number of trace files with the **files *number*** option.

Default: 1024 KB

world-readable | no-world-readable—(Optional) 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.

Required Privilege Level
trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

tunnel-mtu (Services IPsec VPN)

Syntax
tunnel-mtu *bytes*;

Hierarchy Level
[edit services ipsec-vpn rule *rule-name* term *term-name* then]

Release Information
Statement introduced in Junos OS Release 7.5.

Description
Maximum transmission unit (MTU) size for IPsec tunnels.

Options
bytes—MTU size.
Default: 1500 bytes
Range: 256 through 9192 bytes

Usage Guidelines
See "Specifying the MTU for IPsec Tunnels" on page 37.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- mtu

version (IKE)

Syntax
version (1 | 2);

Hierarchy Level
[edit services ipsec-vpn ike policy *policy-name*],

Release Information
Statement introduced in Junos OS Release 11.4.

Description
Configure the Internet Key Exchange (IKE) version that is used to negotiate dynamic SAs for IPSec.

Options
1—Uses IKEv1.

2—Uses IKEv2.

Usage Guidelines
See "Configuring IKE Policies" on page 20.

Required Privilege Level
admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

PART 3

Administration

CHAPTER 6

IP Security Operational Mode Commands

clear security pki ca-certificate

Syntax clear security pki ca-certificate (all | ca-profile *ca-profile-name*)

Release Information Command introduced in Junos OS Release 7.5.

Description Delete certificate authority (CA) digital certificates from the router.

Options **all**—Delete all CA digital certificates from the router.

ca-profile *ca-profile-name*—Delete the specified CA profile.

Required Privilege Level clear

Related Documentation

- request security pki ca-certificate enroll on page 103
- request security pki ca-certificate load on page 104
- show security pki ca-certificate on page 116

List of Sample Output clear security pki ca-certificate all on page 94

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security pki ca-certificate all

```
user@host> clear security pki ca-certificate all
```

clear security pki certificate-request

Syntax clear security pki certificate-request (all | certificate-id *certificate-id-name*)

Release Information Command introduced in Junos OS Release 7.5.

Description Delete manually generated local digital certificate requests from the router.

Options **all**—Delete all local digital certificate requests from the router.

certificate-id *certificate-id-name*—Delete the specified local digital certificate and corresponding public/private key pair.

Required Privilege Level clear

Related Documentation
- show security pki certificate-request on page 120

List of Sample Output clear security pki certificate-request all on page 95

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security pki certificate-request all

```
user@host> clear security pki certificate-request all
```

clear security pki crl

Syntax clear security pki crl (all | ca-profile *ca-profile-name*)

Release Information Command introduced in Junos 8.1

Description Delete certificate revocation lists (CRLs) from the router.

Options **all**—Delete all CRLs from the router.

ca-profile *ca-profile-name*—Delete CRLs associated with the specified CA profile.

Required Privilege Level clear

List of Sample Output clear security pki crl ca-profile all on page 96

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security pki crl ca-profile all

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

clear security pki key-pair

Syntax clear security pki key-pair (all | certificate-id *certificate-id-name*)

Release Information Command introduced in Junos OS Release 8.5.

Description Clear public key infrastructure (PKI) key pair information for local digital certificates from the router.

Options

all—Delete all local digital certificates, certificate requests, and the corresponding public and private key pairs from the router.

certificate-id *certificate-id-name*—Delete the specified local digital certificate and corresponding public/private key pair.

Required Privilege Level clear

Related Documentation

- request security pki local-certificate enroll on page 110
- show security pki local-certificate on page 124

Output Fields This command produces no output.

Sample Output

clear security pki key pair

```
user@host> clear security pki key pair
```

clear security pki local-certificate

Syntax clear security pki local-certificate
<all | certificate-id *certificate-id-name* | system-generated>

Release Information Command introduced in Junos OS Release 7.5.

Description Delete local digital certificates, certificate requests, and the corresponding public/private key pairs from the router.

Options

all—(Optional) Delete all local digital certificates, certificate requests, and the corresponding public and private key pairs from the router.

certificate-id *certificate-id-name*—(Optional) Delete the specified local digital certificate and corresponding public and private key pair.

system-generated—(Optional) Auto-generated self-signed certificate.

Required Privilege Level clear

Related Documentation

- request security pki local-certificate enroll on page 110
- show security pki local-certificate on page 124

List of Sample Output clear security pki local-certificate all on page 98

Output Fields This command produces no output.

Sample Output

clear security pki local-certificate all

```
user@host> clear security pki local-certificate all
```

clear services ipsec-vpn certificates

Syntax clear services ipsec-vpn certificates (all | service-set *service-set*) <certificate-cache-entry *number*>

Release Information Command introduced in Junos OS Release 7.5.

Description (Adaptive services interfaces only) Delete digital certificates from the IPsec configuration memory cache. Issuing this command also clears the certificate revocation list (CRL) from the cache along with the certificates.

Options **all**—Delete digital certificates for all service sets.

service-set *service-set*—Delete digital certificates for the specified service set.

Required Privilege Level clear

List of Sample Output clear services ipsec-vpn certificates all on page 99

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear services ipsec-vpn certificates all

```
user@host> clear services ipsec-vpn certificates all
```

clear services ipsec-vpn ike security-associations

Syntax clear services ipsec-vpn ike security-associations
<*peer-address-name*>
<service-set *service-set-name*>

Release Information Command introduced before Junos OS Release 7.4.
service-set option added in Junos OS Release 8.5.

Description (Adaptive services interfaces only) Clear Internet Key Exchange (IKE) security associations.

Options ***peer-address-name***—(Optional) Clear only the security association specified by the peer address.

service-set *service-set-name*—(Optional) Clear only the security association specified by the service-set name.

Required Privilege Level view

Related Documentation
- show services ipsec-vpn ike security-associations on page 131

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear services ipsec-vpn ike security-associations

```
user@host> clear services ipsec-vpn ike security-associations
```

clear services ipsec-vpn ipsec statistics

Syntax
clear services ipsec-vpn ipsec statistics
<remote-gateway *address*>
<service-set *service-set-name*>

Release Information
Command introduced in Junos OS Release 8.1.

Description
(Adaptive services interface only) Clear IP Security (IPsec) statistics.

Options
remote-gateway *address*—(Optional) Clear statistics for the specified remote system.

service-set *service-set-name*—(Optional) Clear statistics for the specified service set.

Required Privilege Level
view

Related Documentation
- show services ipsec-vpn ipsec statistics on page 139

List of Sample Output
clear services ipsec-vpn ipsec statistics on page 101

Output Fields
When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear services ipsec-vpn ipsec statistics

```
user@host> clear services ipsec-vpn ipsec statistics
```

clear services ipsec-vpn ipsec security-associations

Syntax

clear services ipsec-vpn security-associations
<*peer-address-name*>
<remote-gateway *remote-gateway-address*>
<*service-set-name*>
<tunnel-index *tunnel-index-number*>

Release Information

Command introduced before Junos OS Release 7.4.
remote-gateway, ***service-set-name***, and **tunnel-index** options added in Junos OS Release 8.4.

Description

(Adaptive services interfaces only) Clear IP Security (IPsec) security associations. You can combine the options for greater specificity.

Options

peer-address-name—(Optional) Clear only the security association specified by the peer address.

remote-gateway ***remote-gateway-address***—(Optional) Clear only the security association specified by the remote gateway address.

service-set-name—(Optional) Clear only the security association specified by the service-set name.

tunnel-index ***tunnel-index-number***—(Optional) Clear only the security association specified by the tunnel index number.

Required Privilege Level

view

Related Documentation

- show services ipsec-vpn ipsec security-associations on page 136

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear services ipsec-vpn ipsec security-associations

```
user@host> clear services ipsec-vpn ipsec security-associations
```

request security pki ca-certificate enroll

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
- clear security pki ca-certificate on page 94
- show security pki ca-certificate on page 116

List of Sample Output request security pki ca-certificate enroll on page 103

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=juniper, 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=juniper, 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=juniper
    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

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
- clear security pki ca-certificate on page 94
- show security pki ca-certificate on page 116

List of Sample Output request security pki ca-certificate load on page 104

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 ca-private filename pki-file
```

request security pki ca-certificate verify

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***—Name of the local digital certificate identifier.

Required Privilege Level maintenance

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 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

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 router 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

List of Sample Output request security pki crl load on page 106

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-private filename pki-file
```

request security pki generate-certificate-request

Syntax

request security pki generate-certificate-request certificate-id *certificate-id-name*
domain-name *domain-name* subject *subject-distinguished-name*
<email *email-address*>
<filename (*path* | terminal)>
<ip-address *ip-address*>

Release Information

Command introduced in Junos OS Release 7.5.

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). The 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 that contains the common name, department, company name, state, and country:

- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **ST**—State
- **C**—Country

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

- clear security pki certificate-request on page 95
- show security pki certificate-request on page 120

List of Sample Output

request security pki generate-certificate-request on page 108

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.juniper.net filename entrust-req2 subject cn=router2.juniper.net

Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHAxLmp1bmlwZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFklQws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8o0Xyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH4O6gQ3G
3iHOZfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ7OfO9Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----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

Syntax request security pki generate-key-pair certificate-id *certificate-id-name* <size (512 | 1024 | 2048)>

Release Information Command introduced in Junos OS Release 7.5.

Description Generate a Public Key Infrastructure (PKI) public and 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—(Optional) Key pair size. The key pair size can be **512**, **1024**, or **2048** bits.

Required Privilege Level maintenance

List of Sample Output request security pki generate-key-pair on page 109

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@host> request security pki generate-key-pair certificate-id billy size 2048
Generated key pair billy, key size 2048 bits
```

request security pki local-certificate enroll

Syntax

request security pki local-certificate enroll ca-profile *ca-profile-name*
certificate-id *certificate-id-name* challenge-password *password* domain-name *domain-name* subject *subject-distinguished-name*
<email *email-address*>
<ip-address *ip-address*>

Release Information

Command introduced in Junos OS Release 7.5.

Description

Request that a certificate authority (CA) enroll and install a local digital certificate online by using the 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.

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.

subject *subject-distinguished-name*—Distinguished name format that contains the common name, department, company name, state, and country:

- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **ST**—State
- **C**—Country

email *email-address*—(Optional) E-mail address of the certificate holder.

ip-address *ip-address*—(Optional) IP address of the router.

Required Privilege Level

maintenance

Related Documentation

- show security pki local-certificate on page 124

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 certificate-id r3-entrust-scep ca-profile
entrust domain-name router3.juniper.net subject "CN=router3,OU=Engineering,O=juniper,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 generate-self-signed

Syntax request security pki local-certificate generate-self-signed certificate-id *certificate-id-name* domain-name *domain-name* ip-address *ip-address* email *email-address* subject *subject-distinguished-name*

Release Information Command introduced in Junos OS Release 9.1.

Description Manually generate a self-signed certificate for the given distinguished name.

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). 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.

subject ***subject-distinguished-name***—Distinguished name format that contains the common name, department, company name, state, and country:

- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **ST**—State
- **C**—Country

Required Privilege Level maintenance
security

Related Documentation

- show security pki local-certificate on page 124

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 generate-self-signed certificate-id self-cert
subject cn=abc domain-name juniper.net email mholmes@juniper.net
Self-signed certificate generated and loaded successfully
```

request security pki local-certificate load

Syntax request security pki local-certificate load certificate-id *certificate-id-name* filename *path*

Release Information Command introduced in Junos OS Release 7.5.

Description Manually load a local digital certificate from a specified location.

Options

certificate-id *certificate-id-name*—Name of the public/private key pair mapped to the local digital certificate.

filename *path/filename*—Directory location and filename of the local digital certificate provided by the CA.

Required Privilege Level maintenance

List of Sample Output request security pki local-certificate load on page 113

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 /tmp/router2-cert certificate-id
local-entrust2
Local certificate local-entrust2 loaded successfully
```

request security pki local-certificate verify

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*** —Display the specified certificate identifier name.

Required Privilege Level maintenance

Related Documentation
- show security pki local-certificate on page 124

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
```

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 services ipsec-vpn ipsec switch tunnel

Syntax request services ipsec-vpn ipsec switch tunnel local-gateway *address* remote-gateway *address*
<routing-instance *instance-name*>

Release Information Command introduced before Junos OS Release 7.4.
routing-instance option added in Release 8.1.

Description (Adaptive services interface only) Manually switch between primary and backup IP Security (IPsec) tunnels.

Options
local-gateway *address*—Gateway address of the local system.

remote-gateway *address*—Gateway address of the remote system.

routing-instance *instance-name*—(Optional) VRF instance associated with local gateway address.

Required Privilege Level view

Related Documentation
- show services ipsec-vpn ipsec security-associations on page 136

List of Sample Output request services ipsec-vpn ipsec switch tunnel on page 115

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request services ipsec-vpn ipsec switch tunnel

```
user@host> request services ipsec-vpn ipsec switch tunnel local-gateway 10.1.1.1 remote gateway
10.100.10.1
```

show security pki ca-certificate

Syntax show security pki ca-certificate
<brief | detail>
<ca-profile *ca-profile-name*>

Release Information Command introduced in Junos OS Release 7.5.

Description Display information about certificate authority (CA) digital certificates installed in the router.

Options **none**—(Same as brief) Display information about all CA digital certificates.

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

List of Sample Output show security pki ca-certificate on page 118
show security pki ca-certificate detail on page 118

Output Fields Table 5 on page 116 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 5: show security pki ca-certificate Output Fields

Field Name	Field Description	Level of Output
Certificate identifier	Name of the digital certificate.	All levels
Certificate version	Revision number of the digital certificate.	**detail**
Serial number	Unique serial number of the digital certificate.	**detail**
Issued by	Authority that issued the digital certificate.	**none brief**
Issued to	Device that was issued the digital certificate.	**none brief**
Issuer	Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are: • **Common name**—Name of the authority. • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **State**—State of origin. • **Country**—Country of origin.	**detail**

Table 5: show security pki ca-certificate Output Fields *(continued)*

Field Name	Field Description	Level of Output
Subject	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • **Common name**—Name of the requestor. • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **State**—State of origin. • **Country**—Country of origin.	**detail**
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.	All levels
Public key algorithm	Encryption algorithm used with the private key, such as **rsaEncryption(1024 bits)**.	All levels
Signature algorithm	Encryption algorithm that the CA used to sign the digital certificate, such as **sha1WithRSAEncryption**.	**detail**
Fingerprint	Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate.	**detail**
Distribution CRL	Distinguished name information and the URL for the certificate revocation list (CRL) server.	**detail**
Use for key	Use of the public key, such as **Certificate signing**, **CRL signing**, **Digital signature**, or **Key encipherment**.	**detail**

Sample Output

show security pki ca-certificate

```
user@host> show security pki ca-certificate
Certificate identifier: entrust
  Issued to: juniper, Issued by: juniper
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)

Certificate identifier: entrust
  Issued to: First Officer, Issued by: juniper
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)

Certificate identifier: entrust
  Issued to: First Officer, Issued by: juniper
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
```

show security pki ca-certificate detail

```
user@host> show security pki ca-certificate detail
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 9235
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
    0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
    78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
    19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
    bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
    c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
    04:47:08:07:de:17:23:13
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
    71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: CRL signing, Certificate signing
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925c
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
```

```
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
    1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
    34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
    19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
    ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
    42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
    da:eb:10:27:bd:46:34:33
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
    23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Key encipherment
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925b
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
    d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
    00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
    e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
    90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
    b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
    af:44:bf:53:aa:d4:5f:67
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
    ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

show security pki certificate-request

Syntax
show security pki certificate-request
<brief | detail>
<certificate-id *certificate-id-name*>

Release Information
Command introduced in Junos OS Release 7.5.

Description
Display information about manually generated local digital certificate requests that are stored in the router.

Options
none—(same as brief) Display 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 request

Required Privilege Level
view

Related Documentation
- clear security pki certificate-request on page 95

List of Sample Output
show security pki certificate-request on page 121
show security pki certificate-request detail on page 121

Output Fields
Table 6 on page 120 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 6: show security pki certificate-request Output Fields

Field Name	Field Description	Level of Output
Certificate identifier	Name of the digital certificate.	All levels
Certificate version	Revision number of the digital certificate.	**detail**
Issued to	Device that was issued the digital certificate.	**none brief**
Subject	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • **Common name**—Name of the authority. • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **State**—State of origin. • **Country**—Country of origin.	**detail**
Alternate subject	Domain name or IP address of the device related to the digital certificate.	**detail**

Table 6: show security pki certificate-request Output Fields *(continued)*

Field Name	Field Description	Level of Output
Validity	Time period when the digital certificate is valid. Values are: • **Not before**—Time when the digital certificate becomes valid. • **Not after**—End time when the digital certificate becomes invalid.	All levels
Public key algorithm	Encryption algorithm used with the private key, such as **rsaEncryption(1024 bits)**.	All levels
Public key verification status	Public key verification status: **Failed** or **Passed**. The **detail** output also provides the verification hash.	All levels
Fingerprint	Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate.	**detail**
Use for key	Use of the public key, such as **Certificate signing**, **CRL signing**, **Digital signature**, or **Key encipherment**.	**detail**

Sample Output

show security pki certificate-request

```
user@host> show security pki certificate-request
Certificate identifier: local-microsoft-2
  Issued to: router2.juniper.net
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

show security pki certificate-request detail

```
user@host> show security pki certificate-request detail
Certificate identifier: local-entrust3
  Certificate version: 3
  Subject:
    Common name: router3.juniper.net
  Alternate subject: router3.juniper.net
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
    fb:79:df:d4:a9:03:0f:d3:69:7e:c1:e4:27:35:9c:d9:b1:a2:47:78
    d2:6d:f3:e5:f4:68:4f:b3:04:45:88:57:99:82:39:a6:51:9e:5f:42
    23:3f:d7:6e:3d:a5:54:a9:b1:2d:6e:90:dd:12:8a:bf:ef:2b:20:50
    ba:f0:da:d9:0c:ad:5e:d6:c6:98:3a:ae:3f:90:dd:94:78:c1:ea:2e
    7c:f0:2d:d4:79:d4:cd:f0:52:df:5e:72:f2:e7:ae:66:f7:61:f4:bc
    72:57:3e:6c:6d:d3:24:58:8b:f4:ef:da:2a:6a:fa:eb:98:f8:34:84
    79:54:da:4f:d3:6f:52:1f
  Fingerprint:
    7c:e8:f9:45:93:8d:a3:92:7f:18:29:02:f1:c8:e2:85:3d:ad:df:1f (sha1)
    00:4e:df:a0:6b:ad:8c:50:da:7c:a1:cf:5d:37:b0:ea (md5)
  Use for key: Digital signature
```

show security pki crl

Syntax
show security pki crl
<brief | detail>
<ca-profile *ca-profile-name*>

Release Information
Command introduced in Junos OS Release 8.1.

Description
Display information about the certificate revocation lists (CRLs) that are stored in the router.

Options
none—(same as brief) Display information about all CRLs.

brief | detail—(Optional) Display the specified level of output.

ca-profile *ca-profile-name*—(Optional) Display CRL information about only the specified CA profile.

Required Privilege Level
view

Related Documentation
- clear security pki crl on page 96

List of Sample Output
show security pki crl on page 123
show security pki crl detail on page 123

Output Fields
Table 7 on page 122 shows the output fields for the **show security pki crl** command. Output fields are listed in the approximate order in which they appear.

Table 7: show security pki crl Output Fields

Field Name	Field Description	Level of Output
CA profile	Name of the configured CA profile.	All levels
CRL version	Revision number of the certificate revocation list.	All levels
CRL number	Number of the certificate revocation list	All levels
CRL Issuer	Device that was issued the certificate revocation list.	All levels
Issuer	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • **Common name**—Name of the authority. • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **State**—State of origin. • **Country**—Country of origin.	**detail**
Effective date	Date and time the certificate revocation list becomes valid.	All levels

Table 7: show security pki crl Output Fields *(continued)*

Field Name	Field Description	Level of Output
Next update	Date and time the router will download the latest version of the certificate revocation list.	All levels
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.	**detail**

Sample Output

show security pki crl

```
CA profile entrust
 CRL version: V2
 CRL number: 24
 CRL issuer: C=CA, O=juniper
 Effective date: 2006 May 31st, 05:35:25 GMT
 Next update: 2006 Jun 1st, 06:35:25 GMT
```

show security pki crl detail

```
CA profile: entrust
 CRL version: V2
 CRL number: 24
 Issuer:
  Organization: juniper, Country: ca
 Validity:
  Effective date: 2006 May 31st, 05:35:25 GMT
  Next update: 2006 Jun 1st, 06:35:25 GMT
 Revocation List:
  Serial number      Revocation date
  4451aca3 2006      May 25th, 09:13:38 GMT
  4451aca4 2006      May 25th, 10:11:33 GMT
  4451acb4 2006      May 29th, 11:28:54 GMT
  4451aceb 2006      May 29th, 11:29:01 GMT
  4451acfe 2006      May 29th, 11:29:17 GMT
  4451acff 2006      May 31st, 05:29:55 GMT
```

show security pki local-certificate

Syntax
show security pki local-certificate
<brief | detail>
<certificate-id *certificate-id-name*>
<system-generated>

Release Information
Command introduced in Junos OS Release 7.5.

Description
Display information about the local digital certificates and the corresponding public keys installed in the router.

Options
none—(same as brief) Display information about all local digital certificates and corresponding public keys.

brief | detail—(Optional) Display the specified level of output.

certificate-id *certificate-id-name*—(Optional) Display information about only the specified the local digital certificate and corresponding public keys.

system-generated—(Optional) Auto-generated self-signed certificate.

Required Privilege Level
view

Related Documentation
- clear security pki local-certificate on page 98

List of Sample Output
show security pki local-certificate on page 126
show security pki local-certificate detail on page 126

Output Fields
Table 8 on page 124 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 8: show security pki local-certificate Output Fields

Field Name	Field Description	Level of Output
Certificate identifier	Name of the digital certificate.	All levels
Certificate version	Revision number of the digital certificate.	**detail**
Serial number	Unique serial number of the digital certificate.	**detail**
Issued by	Authority that issued the digital certificate.	none **brief**
Issued to	Device that was issued the digital certificate.	none **brief**

Table 8: show security pki local-certificate Output Fields *(continued)*

Field Name	Field Description	Level of Output
Issuer	Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are: • **Common name**—Name of the authority. • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **State**—State of origin. • **Country**—Country of origin.	**detail**
Subject	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • **Common name**—Name of the authority. • **Organization**—Organization of origin. • **Organizational unit**—Department within an organization. • **State**—State of origin. • **Country**—Country of origin.	**detail**
Alternate subject	Domain name or IP address of the device related to the digital certificate.	**detail**
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.	All levels
Public key algorithm	Encryption algorithm used with the private key, such as **rsaEncryption (1024 bits)**.	All levels
Public key verification status	Public key verification status: **Failed** or **Passed**. The **detail** output also provides the verification hash.	All levels
Signature algorithm	Encryption algorithm that the CA used to sign the digital certificate, such as **sha1WithRSAEncryption**.	**detail**
Fingerprint	Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate.	**detail**
Distribution CRL	Distinguished name information and URL for the certificate revocation list (CRL) server.	**detail**
Use for key	Use of the public key, such as **Certificate signing**, **CRL signing**, **Digital signature**, or **Key encipherment**.	**detail**

Sample Output

show security pki local-certificate

```
user@host> show security pki local-certificate
Certificate identifier: local-entrust2
  Issued to: router2.juniper.net, Issued by: juniper
  Validity:
    Not before: 2005 Nov 21st, 23:28:22 GMT
    Not after: 2008 Nov 21st, 23:58:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

show security pki local-certificate detail

```
user@host> show security pki local-certificate detail
Certificate identifier: local-entrust3
  Certificate version: 3
  Serial number: 4355 94f9
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: router3.juniper.net
  Alternate subject: router3.juniper.net
  Validity:
    Not before: 2005 Nov 21st, 23:33:58 GMT
    Not after: 2008 Nov 22nd, 00:03:58 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
    fb:79:df:d4:a9:03:0f:d3:69:7e:c1:e4:27:35:9c:d9:b1:a2:47:78
    d2:6d:f3:e5:f4:68:4f:b3:04:45:88:57:99:82:39:a6:51:9e:5f:42
    23:3f:d7:6e:3d:a5:54:a9:b1:2d:6e:90:dd:12:8a:bf:ef:2b:20:50
    ba:f0:da:d9:0c:ad:5e:d6:c6:98:3a:ae:3f:90:dd:94:78:c1:ea:2e
    7c:f0:2d:d4:79:d4:cd:f0:52:df:5e:72:f2:e7:ae:66:f7:61:f4:bc
    72:57:3e:6c:6d:d3:24:58:8b:f4:ef:da:2a:6a:fa:eb:98:f8:34:84
    79:54:da:4f:d3:6f:52:1f
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    61:3a:d0:b4:7a:16:9b:39:ba:81:3f:9d:ab:34:e5:c8:be:3b:a1:6d (sha1)
    60:a0:ff:58:05:4a:65:73:9d:74:3a:e1:83:6f:1b:c8 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

show services ipsec-vpn certificates

Syntax show services ipsec-vpn certificates
<brief | detail>
<service-set *service-set*>

Release Information Command introduced in Junos OS Release 7.5.

Description (Adaptive services interfaces only) Display local and remote certificates installed in the IPsec configuration memory cache that are used for the IKE negotiation.

Options

none—(same as brief) Display information about local and remote certificates associated with all service sets.

brief | detail—(Optional) Display the specified level of output.

service-set *service-set*—(Optional) Display information about local and remote certificates associated with only the specified service set.

Required Privilege Level view

List of Sample Output
show security ipsec-vpn certificates on page 129
show security ipsec-vpn certificates detail on page 129

Output Fields Table 9 on page 127 lists the output fields for the **show services ipsec-vpn certificates** command. Output fields are listed in the approximate order in which they appear.

Table 9: show services ipsec-vpn certificates Output Fields

Field Name	Field Description	Level of Output
Service set	Name of the IPsec service set.	All levels
Total entries	Number of certificate cache entries.	All levels
Certificate cache entry	Identification number of the certificate cache entry.	All levels
Flags	Information about the digital certificate, including whether the certificate is a root certificate and trusted.	none **brief**
Issued to	Device that was issued the digital certificate.	none **brief**
Issued by	Authority that issued the digital certificate.	none **brief**
Certificate version	Revision number of the digital certificate.	**detail**
Serial number	Unique serial number of the digital certificate.	**detail**
Alternate subject	Domain name or IP address of the device related to the digital certificate.	All levels

Table 9: show services ipsec-vpn certificates Output Fields *(continued)*

Field Name	Field Description	Level of Output
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.	none **brief**
Public key algorithm	Specifies the encryption algorithm used with the private key, such as **rsaEncryption (1024 bits)**.	**detail**
Signature algorithm	Encryption algorithm that the CA used to sign the digital certificate, such as **sha1WithRSAEncryption**.	**detail**
Fingerprint	Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate.	**detail**
Distribution CRL	Distinguished name information and the URL for the certificate revocation list (CRL) server.	**detail**
Use for key	Use of the public key, such as **Certificate signing**, **CRL signing**, **Digital signature**, or **Key encipherment**.	**detail**

Sample Output

show security ipsec-vpn certificates

```
user@host> show services ipsec-vpn certificates
Service set: serviceset-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
    Flags: Non-root Trusted
    Issued to: router3.juniper.net, Issued by: juniper
    Alternate subject: router3.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:33:58 GMT
      Not after: 2008 Nov 22nd, 00:03:58 GMT

 Certificate cache entry: 2
    Flags: Non-root Trusted
    Issued to: router2.juniper.net, Issued by: juniper
    Alternate subject: router2.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:28:22 GMT
      Not after: 2008 Nov 21st, 23:58:22 GMT

  Certificate cache entry: 1
    Flags: Root Trusted
    Issued to: juniper, Issued by: juniper
    Validity:
      Not before: 2005 Oct 18th, 23:54:22 GMT
      Not after: 2025 Oct 19th, 00:24:22 GMT
```

show security ipsec-vpn certificates detail

```
user@host> show services ipsec-vpn certificates detail
Service set: serviceset-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
    Certificate version: 3
    Serial number: 4355 94f9
    Alternate subject: router3.juniper.net
    Public key algorithm: rsaEncryption
    Signature algorithm: sha1WithRSAEncryption
    Fingerprint:
      61:3a:d0:b4:7a:16:9b:39:ba:81:3f:9d:ab:34:e5:c8:be:3b:a1:6d (sha1)
      60:a0:ff:58:05:4a:65:73:9d:74:3a:e1:83:6f:1b:c8 (md5)
    Distribution CRL:
      C=us, O=juniper, CN=CRL1
      http://CA-1/CRL/juniper_us_crlfile.crl
    Use for key: Digital signature

  Certificate cache entry: 2
    Certificate version: 3
    Serial number: 4355 94f8
    Alternate subject: router2.juniper.net
    Public key algorithm: rsaEncryption
    Signature algorithm: sha1WithRSAEncryption
    Fingerprint:
      30:c3:a4:04:da:33:9d:60:23:5a:48:75:48:2c:f0:c6:96:6c:31:fa (sha1)
      9a:a2:ce:ef:7e:10:80:a0:c8:4d:2f:e7:e1:d3:69:9d (md5)
    Distribution CRL:
      C=us, O=juniper, CN=CRL1
      http://CA-1/CRL/juniper_us_crlfile.crl
    Use for key: Digital signature

  Certificate cache entry: 1
    Certificate version: 3
```

```
Flags: Root
Serial number: 4355 9235
Public key algorithm: rsaEncryption
Signature algorithm: sha1WithRSAEncryption
Fingerprint:
  00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
  71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
Distribution CRL:
  C=us, O=juniper, CN=CRL1
  http://CA-1/CRL/juniper_us_crlfile.crl
Use for key: CRL signing, Certificate signing
```

show services ipsec-vpn ike security-associations

Syntax
show services ipsec-vpn ike security-associations
<brief | detail>
<*peer-address*>

Release Information
Command introduced before Junos OS Release 7.4.
Statistics for Internet Key Exchange (IKE) security associations for each services PIC introduced in Junos OS Release 12.1.

Description
(Adaptive services interface only) Display information for Internet Key Exchange (IKE) security associations. If no security association is specified, the information for all security associations is displayed.

Options
none—(same as brief) Display standard information for all IPsec security associations.

brief | detail—(Optional) Display the specified level of output.

peer-address—(Optional) Display information about a particular security association address.

Required Privilege Level
view

List of Sample Output
show services ipsec-vpn ike security-associations on page 134
show services ipsec-vpn ike security-associations detail on page 134

Output Fields
Table 10 on page 131 lists the output fields for the **show services ipsec-vpn ike security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 10: show services ipsec-vpn ike security-associations Output Fields

Field Name	Field Description	Level of Output
IKE peer	Remote end of the IKE negotiation.	**detail**
Role	Part played in the IKE session. The router triggering the IKE negotiation is the initiator, and the router accepting the first IKE exchange packets is the responder.	**detail**
Remote Address	Responder's address.	none specified
State	State of the IKE security association: • **Matured**—IKE security association is established. • **Not matured**—The IKE security association is in the process of negotiation.	none specified
Initiator cookie	When the IKE negotiation is triggered, a random number is sent to the remote node.	All levels

Table 10: show services ipsec-vpn ike security-associations Output Fields *(continued)*

Field Name	Field Description	Level of Output
Responder cookie	The remote node generates its own random number and sends it back to the initiator as a verification that the packets were received. Of the numerous security services available, protection against denial of service (DoS) is one of the most difficult to address. A “cookie” or anticlogging token (ACT) is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine the cookie's authenticity. An exchange prior to CPU-intensive public key operations can thwart some DoS attempts (such as simple flooding with invalid IP source addresses).	All levels
Exchange type	Specifies the number of messages in an IKE exchange, and the payload types that are contained in each message. Each exchange type provides a particular set of security services, such as anonymity of the participants, perfect forward secrecy of the keying material, and authentication of the participants. Junos OS supports two types of exchanges: • **Main**—The exchange is done with six messages. **Main** encrypts the payload, protecting the identity of the neighbor. • **Aggressive**—The exchange is done with three messages. **Aggressive** does not encrypt the payload, leaving the identity of the neighbor unprotected. • **IKEv2**—The exchange is negotiated using IKE version 2.	All levels
PIC	The services PIC for which the IKE security associations are displayed.	All levels
Authentication method	Type of authentication determines which payloads are exchanged and when they are exchanged. The Junos OS supports only **pre-shared keys**.	**detail**
Local	Prefix and port number of the local end.	**detail**
Remote	Prefix and port number of the remote end.	**detail**
Lifetime	Number of seconds remaining until the IKE security association expires.	**detail**
Algorithms	Header for the IKE algorithms output. • **Authentication**—(**detail** output only) Type of authentication algorithm used: **md5** or **sha1** • **Encryption**—(**detail** output only) Type of encryption algorithm used: **des-cbc**, **3des-cbc**, or **None**. • **Pseudo random function**—Function that generates highly unpredictable random numbers: **hmac-md5** or **hmac-sha1**.	**detail**
Traffic statistics	Number of bytes and packets received and transmitted on the IKE security association. • **Input bytes, Output bytes**—Number of bytes received and transmitted on the IKE security association. • **Input packets, Output packets**—Number of packets received and transmitted on the IKE security association.	**detail**

Table 10: show services ipsec-vpn ike security-associations Output Fields *(continued)*

Field Name	Field Description	Level of Output
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.	**detail**
IPsec security associates	Number of IPsec security associations created and deleted with this IKE security association.	**detail**

Phase 2 negotiations in progress	Number of phase 2 negotiations in progress and status information:
	• Negotiation type—Type of phase 2 negotiation. The 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.
		detail

Sample Output

show services ipsec-vpn ike security-associations

```
user@host> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type

6.6.6.1         Matured       062d291d21275fc7  82ef00e3d1f1c981  Main

6.6.6.2         Matured       cd6d581d7bb1664d  88a707779f3ad8d1  Main

6.6.6.3         Matured       86621051e3e78360  6bc5cc83fd67baa4  IKEv2

PIC: sp-0/3/0

6.6.6.7         Matured       565e2813075e6fdb  67886757a74edcd6  IKEv2
```

show services ipsec-vpn ike

```
user@host> show services ipsec-vpn ike security-associations detail
IKE peer 3.1.0.2
      Role: Responder, State: Matured
```

security-associations detail

```
        Initiator cookie: d91c9f20f78e1d4e, Responder cookie: 727a04ed8d5021a1
        Exchange type: IKEv2, Authentication method: Pre-shared-keys
        Local: 4.1.0.2:500, Remote: 3.1.0.2:500
        Lifetime: Expires in 1357 seconds
        Algorithms:
         Authentication        : sha1
         Encryption            : 3des-cbc
         Pseudo random function: hmac-sha1
        Traffic statistics:
         Input  bytes  :                22244
         Output bytes  :                22236
         Input  packets:                  263
         Output packets:                  263
        Flags: Caller notification sent
        IPSec security associations: 0 created, 0 deleted
        Phase 2 negotiations in progress: 0

IKE peer 4.4.4.4
  Role: Initiator, State: Matured
  Initiator cookie: cf22bd81a7000001, Responder cookie: fe83795c2800002e
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 4.4.4.5:500, Remote: 4.4.4.4:500
  Lifetime: Expires in 187 seconds
  Algorithms:
   Authentication        : md5
   Encryption            : 3des-cbc
   Pseudo random function: hmac-md5
  Traffic statistics:
   Input  bytes  :                1000
   Output bytes  :                1280
   Input  packets:                   5
   Output packets:                   9
  Flags: Caller notification sent
  IPsec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

Negotiation type: Quick mode, Role: Initiator, Message ID: 3582889153
    Local: 4.4.4.5:500, Remote: 4.4.4.4:500
    Local identity: ipv4_subnet(tcp:80,[0..7]=10.1.1.0/24)
    Remote identity: ipv4_subnet(tcp:100,[0..7]=10.1.2.0/24)
    Flags: Caller notification sent, Waiting for done
```

show services ipsec-vpn ipsec security-associations

Syntax show services ipsec-vpn ipsec security-associations
<brief | detail | extensive>
<service-set *service-set-name*>

Release Information Command introduced before Junos OS Release 7.4.

Description (Adaptive services interface only) Display IPsec security associations for the specified service set. If no service set is specified, the security associations for all service sets are displayed.

Options **none**—Display standard information about IPsec security associations for all service sets.

brief | detail | extensive—(Optional) Display the specified level of output.

service-set *service-set-name*—(Optional) Display information about a particular service set.

Required Privilege Level view

List of Sample Output show services ipsec-vpn ipsec security associations extensive on page 138

Output Fields Table 11 on page 136 lists the output fields for the **show services ipsec-vpn ipsec security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 11: show services ipsec-vpn ipsec security-associations Output Fields

Field Name	Field Description	Level of Output
Service set	Name of the service set for which the IPsec security associations are defined. If appropriate, includes the outside service interface VRF name.	All levels
Rule	Name of the rule set applied to the security association.	**detail extensive**
Term	Name of the IPsec term applied to the security association.	**detail extensive**
Tunnel index	Numeric identifier of the specific IPsec tunnel for the security association.	**detail extensive**
Local gateway	Gateway address of the local system.	All levels
Remote gateway	Gateway address of the remote system.	All levels
IPsec inside interface	Name of the logical interface hosting the IPsec tunnels.	All levels
Tunnel MTU	MTU of the IPsec tunnel.	All levels
Local identity	Prefix and port number of the local end	All levels

Table 11: show services ipsec-vpn ipsec security-associations Output Fields *(continued)*

Field Name	Field Description	Level of Output
Remote identity	Prefix and port number of the remote end.	All levels
Primary remote gateway	IP address of the configured primary remote peer.	All levels
Backup remote gateway	IP address of the configured backup remote peer.	All levels
State	State of the primary or backup interface: **Active**, **Offline**, or **Standby**. Both ES PICs are initialized to **Offline**. For primary and backup peers, **State** can be **Active** or **Standby**. If both peers are in a state of **Standby**, no connection exists yet between the two peers.	All levels
Failover counter	Number of times a PIC switched between primary and backup interfaces, or the number of times the tunnel switched between the primary and remote peers since the software has been activated.	All levels
Direction	Direction of the security association: **inbound** or **outbound**.	All levels
SPI	Value of the security parameter index.	All levels
AUX-SPI	Value of the auxiliary security parameter index. • When the value of **Protocol** is **AH** or **ESP**, **AUX-SPI** is always **0**. • When the value of **Protocol** is **AH+ESP**, **AUX-SPI** is always a positive integer.	All levels
Mode	Mode of the security association: • **transport**—Protects single host-to-host protections. • **tunnel**—Protects connections between security gateways.	**detail extensive**
Type	Type of security association: • **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 security associations are not supported in transport mode.	**detail extensive**
State	Status of the security association: • **Installed**—The security association is installed in the security association database. (For transport mode security associations, the value of State must always be Installed) • **Not installed**—The security association is not installed in the security association database.	**detail extensive**
Protocol	Protocol supported: • **transport** mode supports Encapsulation Security Protocol (**ESP**) or Authentication Header (**AH**). • **tunnel** mode supports **ESP** or **AH+ESP**.	All levels

Table 11: show services ipsec-vpn ipsec security-associations Output Fields *(continued)*

Field Name	Field Description	Level of Output
Authentication	Type of authentication used: **hmac-md5-96**, **hmac-sha1-96**, or **none**.	**detail extensive**
Encryption	Type of encryption algorithm used: can be **aes-cbc (128 bits)**, **aes-cbc (192 bits)**, **aes-cbc (256 bits)**, **des-cbc**, **3des-cbc**, or **None**.	**detail**
Soft lifetime **Hard lifetime**	Each lifetime of a security association has two display options, hard and soft, one of which must be present for a dynamic security association. The hard lifetime specifies the lifetime of the SA. The soft lifetime, which is derived from the hard lifetime, informs the IPsec key management system that the SA is about to expire. This information allows the key management system to negotiate a new SA before the hard lifetime expires. • **Expires in** *seconds* **seconds**—Number of seconds left until the security association expires. • **Expires in** *kilobytes* **kilobytes**—Number of kilobytes left until the security association expires.	**detail extensive**
Anti-replay service	State of the service that prevents packets from being replayed: **Enabled** or **Disabled**.	**detail extensive**
Replay window size	Configured size, in packets, of the antireplay service window: **32** or **64**. The antireplay window size protects the receiver against replay attacks by rejecting old or duplicate packets. If the replay window size is **0**, antireplay service is disabled.	**detail**

Sample Output

show services ipsec-vpn ipsec security associations extensive

```
user@host> show services ipsec-vpn ipsec security-associations extensive
Service set: service-set-1
  Rule: _junos_, Term: term-1, Tunnel index: 1
  Local gateway: 101.101.101.2, Remote gateway: 14.14.14.4
  IPSec inside interface: sp-2/0/0.1  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)
  Primary remote gateway: 101.101.101.1, State: Standby
  Backup remote gateway: 14.14.14.4, State: Active
  Failover counter: 1

   Direction: inbound, SPI: 3743521590, AUX-SPI: 0
   Mode: tunnel, Type: dynamic, State: Installed
   Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
   Soft lifetime: Expires in 23043 seconds
   Hard lifetime: Expires in 23178 seconds
   Anti-replay service: Enabled, Replay window size: 64

   Direction: outbound, SPI: 2551045240, AUX-SPI: 0
   Mode: tunnel, Type: dynamic, State: Installed
   Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
   Soft lifetime: Expires in 23043 seconds
   Hard lifetime: Expires in 23178 seconds
   Anti-replay service: Enabled, Replay window size: 64
```

show services ipsec-vpn ipsec statistics

Syntax
show services ipsec-vpn ipsec statistics
<brief | detail>
<remote-gw *remote-peer-address*>
<service-set *service-set-name*>

Release Information
Command introduced before Junos OS Release 7.4.
New fields added in Junos OS Release 10.0.

Description
(Adaptive services interface only) Display IPsec statistics for the specified service set. If no service set is specified, the statistics for all service sets are displayed.

Options
none—Display standard IPsec statistics for all service sets.

brief | detail—(Optional) Display the specified level of output.

remote-gw *remote-peer-address*—(Optional) Display IPsec statistics for an individual IPsec tunnel and an individual remote host.

service-set *service-set-name*—(Optional) Display information about a particular service set.

Required Privilege Level
view

List of Sample Output
show services ipsec-vpn ipsec statistics detail on page 141
show services ipsec-vpn ipsec statistics remote-gw on page 141

Output Fields
Table 12 on page 139 lists the output fields for the **show services ipsec-vpn ipsec statistics** command. Output fields are listed in the approximate order in which they appear.

Table 12: show services ipsec-vpn ipsec statistics Output Fields

Field Name	Field Description	Level of Output
PIC	The physical interface on which the IPsec tunnel is configured.	All levels
Service set	Name of the service set for which the IPsec tunnel is defined.	All levels
Local gateway	Gateway address of the local system.	All levels
Remote gateway	Gateway address of the remote system.	All levels
Tunnel index	Numeric identifier of the specific IPsec tunnel for the security association.	All levels

Table 12: show services ipsec-vpn ipsec statistics Output Fields *(continued)*

Field Name	Field Description	Level of Output
ESP statistics	Encapsulation Security Payload (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	All levels
AH Statistics	Authentication Header 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.	All levels
Errors	• **AH authentication failures**—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. • **ESP authentication failures**—Number of Encapsulation Security Payload (ESP) failures. An ESP failure occurs when there is an authentication mismatch in ESP packets. • **ESP Decryption failures**—Number of ESP decryption failures. • **Bad headers**—Number of invalid headers detected. • **Bad trailers**—Number of invalid trailers detected. • **Replay before window drops**—Number of replay errors. A replay error is generated when a duplicate packet is received within the replay window. • **Replayed pkts**—Number of packets replayed. • **IP integrity errors**—Number of IP integrity errors. • **Exceeds tunnel MTU**—Number of times the tunnel maximum transmission unit (MTU) value was exceeded. • **Rule lookup failures**—Number of rule lookup failures. • **No SA errors**—Number of errors resulting from a missing security association (SA). • **Flow errors**—Number of flow errors. • **Misc errors**—Number of miscellaneous errors.	All levels

Sample Output

show services ipsec-vpn ipsec statistics detail

```
user@host> show services ipsec-vpn ipsec statistics
PIC: sp-0/2/0, Service set: ss0

ESP Statistics:
  Encrypted bytes:                0
  Decrypted bytes:                0
  Encrypted packets:              0
  Decrypted packets:              0
AH Statistics:
  Input bytes:                  168
  Output bytes:                 168
  Input packets:                  2
  Output packets:                 2
Errors:
  AH authentication failures:     0
  ESP authentication failures:    0
  ESP decryption failures:        0
  Bad headers: 0, Bad trailers: 0
  Replay before window drops: 0, Replayed pkts: 0
  IP integrity errors: 0, Exceeds tunnel MTU: 0
  Rule lookup failures: 0, No SA errors: 0
  Flow errors: 0, Misc errors: 0
```

show services ipsec-vpn ipsec statistics remote-gw

```
user@host> show services ipsec-vpn ipsec statistics remote-gw 22.22.2.1
PIC: sp-3/1/0, Service set: service-set-2
Local gateway: 22.22.1.1, Remote gateway: 22.22.2.1, Tunnel index: 2
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
  ESP authentication failures:    0
  ESP decryption failures:        0
  Bad headers: 0, Bad trailers: 0
  Replay before window drops: 0, Replayed pkts: 0
  IP integrity errors: 0, Exceeds tunnel MTU: 0
  Rule lookup failures: 0, No SA errors: 0
  Flow errors: 0, Misc errors: 0
```

PART 4

Index

Index

Symbols

A

B

C

D

E

F

I

R

S

T

V