JUNIPER NETWORKS

Junos® OS

System Basics: Security Services Configuration Guide

Release
12.1

Published: 2012-03-13

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

Junos® OS System Basics: Security Services Configuration Guide
12.1

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
- MX Series
- T 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 *Junos OS CLI User Guide*.

Documentation Conventions

Table 1 on page xv 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 xv 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 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; interface names; 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 Overview

IPsec Overview

IP Security (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.

For a complete description of the IPsec security suite, see the *IPsec Feature Guide*.

Related Documentation

- IPsec Configuration for an ES PIC Overview on page 4
- Security Associations Overview on page 7

IPsec Requirements for Junos-FIPS

In a Junos-FIPS environment, hardware configurations with two Routing Engines must be configured to use IPsec and a private routing instance for all communications between the Routing Engines. IPsec communication between the Routing Engines and AS II FIPS PICs is also required.

Related Documentation

- Security Associations Overview on page 7
- IKE Key Management Protocol Overview on page 7
- Security Services Configuration Statements on page 73

IPsec Configuration for an ES PIC Overview

IP Security (IPsec) provides a secure way to authenticate senders and encrypt IPv4 and IPv6 traffic between network devices, such as routers and hosts. The following sections show how to configure IPsec for an ES PIC.

The key management process (**kmd**) provides IPsec authentication services for ES PICs. The key management process starts only when IPsec is configured on the router.

Related Documentation

- Configuring Minimum Manual Security Associations for IPsec on an ES PIC on page 13
- Configuring Minimum Digital Certificate Requirements for IKE on an ES PIC on page 14
- Configuring Security Associations for IPsec on an ES PIC on page 15
- Configuring an IKE Proposal for Dynamic SAs on page 26
- Example: Configuring an IKE Proposal on page 29

ES Tunnel Interface Configuration for a Layer 3 VPN

To configure an ES tunnel interface for a Layer 3 VPN, you need to configure an ES tunnel interface on the provider edge (PE) router and on the customer edge (CE) router. You also need to configure IPsec on the PE and CE routers.

Related Documentation

- IPsec Tunnel Traffic Configuration Overview on page 4

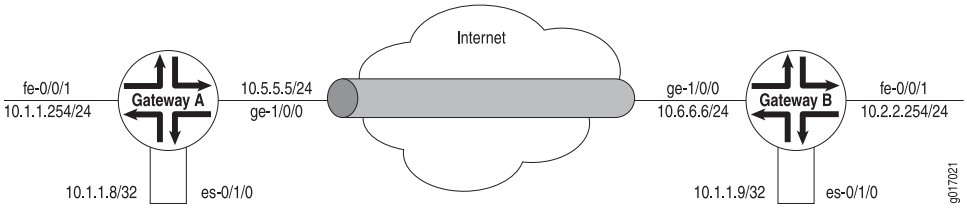
IPsec Tunnel Traffic Configuration Overview

Traffic configuration defines the traffic that must flow through the IPsec tunnel. You configure outbound and inbound firewall filters, which identify and direct traffic to be encrypted and confirm that decrypted traffic parameters match those defined for the given tunnel. The outbound filter is applied to the LAN or WAN interface for the incoming traffic you want to encrypt off of that LAN or WAN. The inbound filter is applied to the ES PIC to check the policy for traffic coming in from the remote host. Because of the complexity of configuring a router to forward packets, no automatic checking is done to ensure that the configuration is correct. Make sure that you configure the router very carefully.

NOTE: **The valid firewall filters statements for IPsec are destination-port, source-port, protocol, destination-address, and source-address.**

In Figure 1 on page 5, Gateway A protects the network **10.1.1.0/24**, and Gateway B protects the network **10.2.2.0/24**. The gateways are connected by an IPsec tunnel.

Figure 1: Example: IPsec Tunnel Connecting Security Gateways



The SA and ES interfaces for Gateway A are configured as follows:

```
[edit security ipsec]
security-association manual-sa1 {
  manual {
    direction bidirectional {
      protocol esp;
      spi 2312;
      authentication {
        algorithm hmac-md5-96;
        key ascii-text 1234123412341234;
      }
      encryption {
        algorithm 3des-cbc;
        key ascii-text 12345678900987654321234;
      }
    }
  }
}
[edit interfaces es-0/1/0]
unit 0 {
  tunnel {
    source 10.5.5.5;
    destination 10.6.6.6;
  }
  family inet {
    ipsec-sa manual-sa1;
    address 10.1.1.8/32 {
      destination 10.1.1.9;
    }
  }
}
```

The SA and ES interfaces for Gateway B are configured as follows:

```
[edit security ipsec]
security-association manual-sa1 {
  manual {
    direction bidirectional {
      protocol esp;
      spi 2312;
      authentication {
        algorithm hmac-md5-96;
        key ascii-text 1234123412341234;
      }
      encryption {
        algorithm 3des-cbc;
```

```
            key ascii-text 12345678900987654321234;
          }
        }
      }
    }
[edit interfaces es-0/1/0]
unit 0 {
  tunnel {
    source 10.6.6.6;
    destination 10.5.5.5;
  }
  family inet {
    ipsec-sa manual-sa1;
    address 10.1.1.9/32; {
      destination 10.1.1.8;
    }
  }
}
```

Related Documentation

- Example: Configuring an Outbound Traffic Filter on page 61
- Example: Applying an Outbound Traffic Filter on page 62
- Example: Configuring an Inbound Traffic Filter for a Policy Check on page 62
- ES Tunnel Interface Configuration for a Layer 3 VPN on page 4

CHAPTER 2

Security Associations and IKE Overview

Security Associations Overview

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

- 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.

The Junos OS implementation of IPsec supports two modes of security (transport mode and tunnel mode).

Related Documentation

- IKE Key Management Protocol Overview on page 7
- IPsec Requirements for Junos-FIPS on page 3
- Security Services Configuration Statements on page 73

IKE Key Management Protocol Overview

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 does the following:

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

IKE occurs over two phases. In the first phase, it negotiates security attributes and establishes shared secrets to form the bidirectional IKE SA. In the second phase, inbound and outbound IPsec SAs are established. The IKE SA secures the exchanges in the second phase. IKE also generates keying material, provides Perfect Forward Secrecy, and exchanges identities.

Related Documentation

- Security Associations Overview on page 7
- IPsec Requirements for Junos-FIPS on page 3
- Security Services Configuration Statements on page 73

CHAPTER 3

Digital Certificates Overview

Digital Certificates Overview

A digital certificate provides a way of authenticating users through a trusted third-party called a certificate authority (CA). The CA validates the identity of a certificate holder and "signs" the certificate to attest that it has not been forged or altered.

A certificate includes the following information:

- The distinguished name (DN) of the owner. A DN is a unique identifier and consists of a fully qualified name including the common name (CN) of the owner, the owner's organization, and other distinguishing information.
- The public key of the owner.
- The date on which the certificate was issued.
- The date on which the certificate expires.
- The distinguished name of the issuing CA.
- The digital signature of the issuing CA.

The additional information in a certificate allows recipients to decide whether to accept the certificate. The recipient can determine if the certificate is still valid based on the expiration date. The recipient can check whether the CA is trusted by the site based on the issuing CA.

With a certificate, a CA takes the owner's public key, signs that public key with its own private key, and returns this to the owner as a certificate. The recipient can extract the certificate (containing the CA's signature) with the owner's public key. By using the CA's public key and the CA's signature on the extracted certificate, the recipient can validate the CA's signature and owner of the certificate.

When you use digital certificates, your first send in a request to obtain a certificate from your CA. You then configure digital certificates and a digital certificate IKE policy. Finally, you obtain a digitally signed certificate from a CA.

NOTE: Certificates without an alternate subject name are not appropriate for IPsec services.

Related Documentation

- Configuration Statements for Configuring Digital Certificates for an ES PIC on page 39
- Obtaining a Certificate from a Certificate Authority for an ES PIC on page 40
- Requesting a CA Digital Certificate for an ES PIC on an M Series or T Series Router on page 41
- Generating a Private and Public Key Pair for Digital Certificates for an ES PIC on page 41
- Configuring Digital Certificates for an ES PIC on page 43
- Configuring an IKE Policy for Digital Certificates for an ES PIC on page 47
- Associating the Configured Security Association with a Logical Interface on page 49

PART 2

Configuration

CHAPTER 4

Configuring IPsec for an ES PIC

Configuring Minimum Manual Security Associations for IPsec on an ES PIC

To define a manual security association (SA) configuration for an ES PIC, include at least the following statements at the **[edit security ipsec]** hierarchy level:

```
[edit security ipsec]
security-association sa-name {
  manual {
    direction (inbound | outbound | bidirectional) {
      authentication {
        algorithm (hmac-md5-96 | hmac-sha1-96);
        key (ascii-text key | hexadecimal key);
      }
      encryption {
        algorithm (des-cbc | 3des-cbc);
        key (ascii-text key | hexadecimal key);
      }
      protocol (ah | esp | bundle);
```

```
        spi spi-value;
      }
    }
  }
```

Related Documentation

- IPsec Configuration for an ES PIC Overview on page 4

Configuring Minimum IKE Requirements for IPsec on an ES PIC

To define an IKE configuration for an ES PIC, include at least the following statements at the **[edit security]** hierarchy level:

```
[edit security ike]
proposal ike-proposal-name {
  authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
  dh-group (group1 | group2);
  encryption-algorithm (3des-cbd | des-cbc | ase-128-cbc | ase-192-cbc | ase-256-cbc);
}
policy ike-peer-address {
  proposals [ ike-proposal-names ];
  pre-shared-key (ascii-text key | hexadecimal key);
}
```

Related Documentation

- IPsec Configuration for an ES PIC Overview on page 4

Configuring Minimum Digital Certificate Requirements for IKE on an ES PIC

To define a digital certificate configuration for IKE for an encryption interface on M Series and T Series routers, include at least the following statements at the **[edit security certificates]** and **[edit security ike]** hierarchy levels:

```
[edit security]
certificates {
  certification-authority ca-profile-name {
    ca-name ca-identity;
    crl filename;
    enrollment-url url-name;
    file certificate-filename;
    ldap-url url-name;
  }
}
ike {
  policy ike-peer-address {
    local-certificate certificate-filename;
    local-key-pair private-public-key-file;
    proposal [ ike-proposal-names ];
  }
  proposal ike-proposal-name {
    authentication-method rsa-signatures;
  }
}
```

Related Documentation

- IPsec Configuration for an ES PIC Overview on page 4

Configuring Security Associations for IPsec on an ES PIC

To use IPsec security 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 IPsec Security Associations for an ES PIC" on page 17.
- Dynamic—Specify 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 "Associating the Configured Security Association with a Logical Interface" on page 49.

NOTE: The Junos OS does not perform a commit check when an SA name referenced in the Border Gateway Protocol (BGP) protocol section is not configured at the **[edit security ipsec]** hierarchy level.

We recommend that you configure no more than 512 dynamic security associations per ES Physical Interface Card (PIC).

To configure an SA for IPsec for an ES PIC, include the **security-association** statement at the **[edit security ipsec]** hierarchy level:

```
[edit security ipsec]
security-association sa-name;
```

NOTE: You configure a dynamic SA for the AS and MultiServices PICs at the **[edit services ipsec-vpn rule** *rule-name* **term** *term-name* **then dynamic]**, **[edit services ipsec-vpn ike]**, and **[edit services ipsec-vpn ipsec]** hierarchy levels.

For more information, see the "IPsec" chapter of the *Junos OS Feature Guides* and the "IPsec Services Configuration Guidelines" chapter of the *Junos OS Services Interfaces Configuration Guide*.

Tasks to configure SAs for IPsec for an ES PIC are:

1. Configuring the Description for an SA on page 16
2. Configuring IPsec Transport Mode on page 16
3. Configuring IPsec Tunnel Mode on page 16

4. Configuring Manual IPsec Security Associations for an ES PIC on page 17
5. Configuring Dynamic IPsec Security Associations on page 21
6. Enabling Dynamic IPsec Security Associations on page 21

Configuring the Description for an SA

To specify a description for an IPsec SA, include the **description** statement at the **edit security ipsec security-association *sa-name*]** hierarchy level:

```
[edit security ipsec security-association sa-name]
description description;
```

Configuring IPsec Transport Mode

In transport mode, the data portion of the IP packet is encrypted, but the IP header is not. Transport mode can be used only when the communication endpoint and cryptographic endpoint are the same. Virtual private network (VPN) gateways that provide encryption and decryption services for protected hosts cannot use transport mode for protected VPN communications. You configure manual SAs, and you must configure static values on both ends of the SA.

NOTE: When you use transport mode, the Junos OS supports both BGP and OSPFv3 for manual SAs.

To configure IPsec security for transport mode, include the **mode** statement with the **transport** option at the **edit security ipsec security-association *sa-name*]** hierarchy level:

```
[edit security ipsec security-association sa-name]
mode transport;
```

To apply tunnel mode, you configure manual SAs in transport mode and then reference the SA by name at the **[edit protocols bgp]** hierarchy level to protect a session with a given peer.

NOTE: You can configure BGP to establish a peer relationship over encrypted tunnels.

Configuring IPsec Tunnel Mode

You use tunnel mode when you use preshared keys with IKE to authenticate peers, or digital certificates with IKE to authenticate peers.

When you use preshared keys, you manually configure a preshared key, which must match that of its peer. With digital certificates, each router is dynamically or manually enrolled with a certificate authority (CA). When a tunnel is established, the public keys used for IPsec are dynamically obtained through IKE and validated against the CA certificate. This avoids the manual configuration of keys on routers within the topology. Adding a new router to the topology does not require any security configuration changes to existing routers.

To configure the IPsec in tunnel mode, include the **mode** statement with the **tunnel** option at the **edit security ipsec security-association *sa-name*]** hierarchy level:

```
[edit security ipsec security-association sa-name]
mode tunnel;
```

NOTE: The Junos OS supports both both BGP and OSPFv3 in transport mode.

To enable tunnel mode, follow the steps in these sections:

- Configuring Security Associations for IPsec on an ES PIC on page 15
- Configuring an IKE Proposal for Dynamic SAs on page 26
- Associating the Configured Security Association with a Logical Interface on page 49
- IPsec Tunnel Traffic Configuration Overview on page 4

Configuring Manual IPsec Security Associations for an ES PIC

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

Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. As a result, peers can communicate only when they all share the same configured options.

To configure the manual IPsec SA for an ES PIC, include the **manual** statement at the **edit security ipsec security-association *sa-name*]** hierarchy level:

```
[edit security ipsec security-association sa-name]
manual {
  direction (inbound | outbound | bi-directional) {
    authentication {
      algorithm (hmac-md5-96 | hmac-sha1-96);
      key (ascii-text key | hexadecimal key);
    }
    auxiliary-spi auxiliary-spi-value;
    encryption {
      algorithm (des-cbc | 3des-cbc);
      key (ascii-text key | hexadecimal key);
    }
    protocol (ah | esp | bundle);
    spi spi-value;
  }
}
```

Tasks to configure a manual SA are:

1. Configuring the Processing Direction on page 18
2. Configuring the Protocol for a Manual SA on page 19
3. Configuring the Security Parameter Index on page 19

4. Configuring the Auxiliary Security Parameter Index on page 19
5. Configuring the Authentication Algorithm and Key on page 20
6. Configuring the Encryption Algorithm and Key on page 20

Configuring the Processing Direction

The **direction** statement sets inbound and 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 and specify the direction at the **[edit security ipsec security-association *sa-name* manual]** hierarchy level:

```
[edit security ipsec security-association sa-name manual]
direction (inbound | outbound | bidirectional);
```

The following example shows how to define different algorithms, keys, and security parameter index values for inbound and outbound processing directions:

```
[edit security ipsec security-association sa-name]
manual {
    direction inbound {
        encryption {
            algorithm 3des-cbc;
            key ascii-text 2345678901234567890234;
        }
        protocol esp;
        spi 16384;
    }
    direction outbound {
        encryption {
            algorithm 3des-cbc;
            key ascii-text 12345678901234567890abcd;
        }
        protocol esp;
        spi 24576;
    }
}
```

The following example shows how to define the same algorithms, keys, and security parameter index values for bidirectional processing:

```
[edit security ipsec security-association sa-name manual]
direction bidirectional {
    authentication {
        algorithm hmac-md5-96;
        key ascii-text 123456789012abcd;
    }
    protocol ah;
    spi 20001;
}
```

Configuring the Protocol for a Manual SA

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

NOTE: The AH protocol is supported only on M Series routers.

To configure the IPsec protocol on an ES PIC, include the **protocol** statement at the **edit security ipsec security-association** *sa-name* **manual direction (inbound | outbound | bidirectional)]** hierarchy level and specify the **ah**, **bundle**, or **esp** option:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bi-directional)]
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 on an ES PIC, include the **spi** statement and specify a value (256 through 16,639) at the **[edit security ipsec security-association** *sa-name* **manual direction (inbound | outbound | bi-directional]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bidirectional)]
spi spi-value;
```

Configuring the Auxiliary Security Parameter Index

When you configure the **protocol statement to use the bundle** option, the Junos OS uses the auxiliary SPI for the ESP and the SPI for the AH.

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

To configure the auxiliary SPI, include the **auxiliary-spi** statement at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level and set the value to an integer between 256 and 16,639:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bidirectional)]
auxiliary-spi auxiliary-spi-value;
```

Configuring the Authentication Algorithm and Key

To configure an authentication algorithm and key, include the **authentication** statement at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound | outbound |
  bidirectional)]
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 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 *key***—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 *key***—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 the Encryption Algorithm and Key

To configure IPsec encryption, include the **encryption** statement and specify an algorithm and key at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound | outbound |
  bi-directional)]
encryption {
  algorithm (des-cbc | 3des-cbc);
  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.

NOTE: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409. For **3des-cbc**, we recommend that the first 8 bytes not be the same as the second 8 bytes, and that the second 8 bytes be the same as the third 8 bytes.

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 IPsec 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 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 configure a dynamic SA, include the **dynamic** statement at the **[edit security ipsec security-association** *sa-name***]** hierarchy level. Specify an IPsec policy name, and optionally, a 32-packet or 64-packet replay window size.

```
[edit security ipsec security-association sa-name]
dynamic {
   ipsec-policy policy-name;
   replay-window-size (32 | 64);
}
```

Enabling Dynamic IPsec Security Associations

To enable a dynamic SA, follow these steps:

1. Configure 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.

NOTE: Dynamic tunnel SAs require an ES PIC. 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.

The replay window is not used with manual SAs.

Configuring Manual IPsec Security Associations for an ES PIC

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

Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. As a result, peers can communicate only when they all share the same configured options.

To configure the manual IPsec SA for an ES PIC, include the **manual** statement at the **edit security ipsec security-association *sa-name*]** hierarchy level:

```
[edit security ipsec security-association sa-name]
manual {
   direction (inbound | outbound | bi-directional) {
      authentication {
         algorithm (hmac-md5-96 | hmac-sha1-96);
         key (ascii-text key | hexadecimal key);
      }
      auxiliary-spi auxiliary-spi-value;
      encryption {
         algorithm (des-cbc | 3des-cbc);
         key (ascii-text key | hexadecimal key);
      }
      protocol (ah | esp | bundle);
      spi spi-value;
   }
}
```

Tasks to configure a manual SA are:

1. Configuring the Processing Direction on page 22
2. Configuring the Protocol for a Manual SA on page 23
3. Configuring the Security Parameter Index on page 24
4. Configuring the Auxiliary Security Parameter Index on page 24
5. Configuring the Authentication Algorithm and Key on page 25
6. Configuring the Encryption Algorithm and Key on page 25

Configuring the Processing Direction

The **direction** statement sets inbound and 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 and specify the direction at the **[edit security ipsec security-association *sa-name* manual]** hierarchy level:

```
[edit security ipsec security-association sa-name manual]
direction (inbound | outbound | bidirectional);
```

The following example shows how to define different algorithms, keys, and security parameter index values for inbound and outbound processing directions:

```
[edit security ipsec security-association sa-name]
manual {
    direction inbound {
        encryption {
            algorithm 3des-cbc;
            key ascii-text 2345678901234567890123456;
        }
        protocol esp;
        spi 16384;
    }
    direction outbound {
        encryption {
            algorithm 3des-cbc;
            key ascii-text 1234567890123456789abcdef;
        }
        protocol esp;
        spi 24576;
    }
}
```

The following example shows how to define the same algorithms, keys, and security parameter index values for bidirectional processing:

```
[edit security ipsec security-association sa-name manual]
direction bidirectional {
    authentication {
        algorithm hmac-md5-96;
        key ascii-text 123456789012abcd;
    }
    protocol ah;
    spi 20001;
}
```

Configuring the Protocol for a Manual SA

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

NOTE: The AH protocol is supported only on M Series routers.

To configure the IPsec protocol on an ES PIC, include the **protocol** statement at the **edit security ipsec security-association** *sa-name* **manual direction (inbound | outbound | bidirectional)]** hierarchy level and specify the **ah**, **bundle**, or **esp** option:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bi-directional)]
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 on an ES PIC, include the **spi** statement and specify a value (256 through 16,639) at the **[edit security ipsec security-association** *sa-name* **manual direction (inbound | outbound | bi-directional]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bidirectional)]
spi spi-value;
```

Configuring the Auxiliary Security Parameter Index

When you configure the **protocol statement to use the bundle** option, the Junos OS uses the auxiliary SPI for the ESP and the SPI for the AH.

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

To configure the auxiliary SPI, include the **auxiliary-spi** statement at the **[edit security ipsec security-association** *sa-name* **manual direction (inbound | outbound | bi-directional)]** hierarchy level and set the value to an integer between 256 and 16,639:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bidirectional)]
auxiliary-spi auxiliary-spi-value;
```

Configuring the Authentication Algorithm and Key

To configure an authentication algorithm and key, include the **authentication** statement at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound | outbound |
  bidirectional)]
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 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 *key***—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 *key***—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 the Encryption Algorithm and Key

To configure IPsec encryption, include the **encryption** statement and specify an algorithm and key at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound | outbound |
  bi-directional)]
encryption {
  algorithm (des-cbc | 3des-cbc);
  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.

NOTE: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409. For **3des-cbc**, we recommend that the first 8 bytes not be the same as the second 8 bytes, and that the second 8 bytes be the same as the third 8 bytes.

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 IPsec 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 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 configure a dynamic SA, include the **dynamic** statement at the **[edit security ipsec security-association *sa-name*]** hierarchy level. Specify an IPsec policy name, and optionally, a 32-packet or 64-packet replay window size.

```
[edit security ipsec security-association sa-name]
dynamic {
   ipsec-policy policy-name;
   replay-window-size (32 | 64);
}
```

Related Documentation

- Configuring Manual IPsec Security Associations for an ES PIC on page 17

Configuring an IKE Proposal for Dynamic SAs

Dynamic Security Associations (SAs) require IKE configuration. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway.

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

To configure an IKE proposal and define its properties, include the following statements at the **[edit security ike]** hierarchy level:

```
[edit security ike]
proposal ike-proposal-name {
  authentication-algorithm (md5 | sha1);
  authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
  description description;
  dh-group (group1 | group2);
  encryption-algorithm (3des-cbc | des-cbc | ase-128-cbc | ase-192-cbc | ase-256-cbc);
  lifetime-seconds seconds;
}
```

For information about associating an IKE proposal with an IKE policy, see "Configuring an IKE Policy for Preshared Keys" on page 29.

Tasks for configuring the IKE proposal are:

1. Configuring the Authentication Algorithm for an IKE Proposal on page 27
2. Configuring the Authentication Method for an IKE Proposal on page 27
3. Configuring the Description for an IKE Proposal on page 28
4. Configuring the Diffie-Hellman Group for an IKE Proposal on page 28
5. Configuring the Encryption Algorithm for an IKE Proposal on page 28
6. Configuring the Lifetime for an IKE SA on page 29

Configuring the Authentication Algorithm for an IKE Proposal

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

```
[edit security ike proposal ike-proposal-name]
authentication-algorithm (md5 | sha1);
```

The authentication algorithm can be one of the following:

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

Configuring the Authentication Method for an IKE Proposal

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

```
[edit security ike proposal ike-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 (DSA)
- **pre-shared-keys**—Preshared keys; a key derived from an out-of-band mechanism is used to authenticate an exchange
- **rsa-signatures**—Public key algorithm that supports encryption and digital signatures

Configuring the Description for an IKE Proposal

To specify a description for an IKE proposal, include the **description** statement at the **[edit security ike proposal *ike-proposal-name*]** hierarchy level:

```
[edit security ike proposal ike-proposal-name]
description description;
```

Configuring the Diffie-Hellman Group for an IKE Proposal

The Diffie-Hellman key exchange 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 an IKE Diffie-Hellman group, include the **dh-group** statement at the **[edit security ike proposal *ike-proposal-name*]** hierarchy level:

```
[edit security ike proposal ike-proposal-name ]
dh-group (group1 | group2);
```

The group can be one of the following:

- **group1**—Specify that IKE use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group2**—Specify that IKE use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

group2 provides more security but requires more processing time.

Configuring the Encryption Algorithm for an IKE Proposal

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

```
[edit security ike proposal ike-proposal-name ]
encryption-algorithm (3des-cbc | des-cbc);
```

The encryption algorithm can be one of the following:

- **3des-cbc**—Encryption algorithm that has a key size of 24 bytes; its key size is 192 bits long.
- **des-cbc**—Encryption algorithm that has a key size of 8 bytes; its key size is 56 bits long.
- **aes-128-cbc**—Advanced encryption algorithm that has a key size of 16 bytes; its key size is 128 bits long.
- **aes-192-cbc**—Advanced encryption algorithm that has a key size of 24 bytes; its key size is 192 bits long.
- **aes-256-cbc**—Advanced encryption algorithm that has a key size of 32 bytes; its key size is 256 bits long.

Configuring the Lifetime for an IKE SA

The IKE lifetime sets the lifetime of an IKE SA. When the IKE SA expires, it is replaced by a new SA (and SPI) or is terminated. The default value IKE lifetime is 3600 seconds.

To configure the IKE lifetime, include the **lifetime-seconds** statement and specify the number of seconds (180 through 86,400) at the **[edit security ike proposal *ike-proposal-name*]** hierarchy level:

```
[edit security ike proposal ike-proposal-name ]
lifetime-seconds seconds;
```

Example: Configuring an IKE Proposal

The following example shows how to configure an IKE proposal:

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

Related Documentation

- Configuring an IKE Proposal for Dynamic SAs on page 26

Configuring an IKE Policy for Preshared Keys

An IKE policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address, the preshared key for the given peer, and the proposals needed for that connection. 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.

You can create multiple, prioritized proposals at each peer to ensure that at least one proposal will match 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 at the **[edit security ike]** hierarchy level and specify a peer address:

```
[edit security ike]
policy ike-peer-address;
```

NOTE: The IKE policy peer address must be an IPsec tunnel destination address.

Tasks for configuring an IKE policy are:

1. Configuring the Description for an IKE Policy on page 30
2. Configuring the Mode for an IKE Policy on page 30
3. Configuring the Preshared Key for an IKE Policy on page 30
4. Associating Proposals with an IKE Policy on page 31

Configuring the Description for an IKE Policy

To specify a description for an IKE policy, include the **description** statement at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike policy ike-peer-address]
description description;
```

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 key 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.

To configure IKE policy mode, include the **mode** statement and specify **aggressive** or **main** at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike policy ike-peer-address ]
mode (aggressive | main);
```

Configuring the Preshared Key for an IKE Policy

IKE policy preshared keys authenticate peers. 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.

A local certificate is an alternative to the preshared key. A commit operation fails if either a preshared key or a local certificate is not configured.

To configure an IKE policy preshared key, include the **pre-shared-key** statement at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike policy ike-peer-address]
pre-shared-key (ascii-text key | hexadecimal key);
```

Associating Proposals with an IKE Policy

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

To configure an IKE policy proposal, include the **proposals** statement at the **[edit security ike policy *ike-peer-address*]** hierarchy level and specify one or more proposal names:

```
[edit security ike policy ike-peer-address]
proposals [ proposal-names ];
```

Related Documentation

- Example: Configuring an IKE Policy on page 31

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**.

```
[edit security]
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-filename;
      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 System Basics and Services Command Reference*.

Related Documentation

- Configuring an IKE Policy for Preshared Keys on page 29

Configuring an IPsec Proposal for an ES PIC

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

To configure an IPsec proposal and define its properties, include the following statements at the **[edit security ipsec]** hierarchy level:

```
[edit security ipsec]
proposal ipsec-proposal-name {
   authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
   description description ;
   encryption-algorithm (3des-cbc | des-cbc);
   lifetime-seconds seconds;
   protocol (ah | esp | bundle);
}
```

Tasks to configure an IPsec proposal for an ES PIC include:

- Configuring the Authentication Algorithm for an IPsec Proposal on page 32
- Configuring the Description for an IPsec Proposal on page 33
- Configuring the Encryption Algorithm for an IPsec Proposal on page 33
- Configuring the Lifetime for an IPsec SA on page 33
- Configuring the Protocol for a Dynamic IPsec SA on page 34

Configuring the Authentication Algorithm for an IPsec Proposal

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

```
[edit security ipsec proposal ipsec-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 a description for an IPsec proposal, include the **description** statement at the **[edit security ipsec proposal *ipsec-proposal-name*]** hierarchy level:

```
[edit security ike policy ipsec-proposal-name]
description description;
```

Configuring the Encryption Algorithm for an IPsec Proposal

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

```
[edit security ipsec proposal ipsec-proposal-name ]
encryption-algorithm (3des-cbc | des-cbc);
```

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.

NOTE: We recommend that you use the triple DES cipher block chaining (3DES-CBC) encryption algorithm.

Configuring the Lifetime for an IPsec SA

The IPsec lifetime option sets the lifetime of an IPsec SA. When the IPsec SA expires, it is replaced by a new SA (and SPI) or is terminated. A new SA has new authentication and encryption keys, and SPI; however, the algorithms may remain the same if the proposal is not changed. If you do not configure a lifetime and a lifetime is not sent by a responder, the lifetime is 28,800 seconds.

To configure the IPsec lifetime, include the **lifetime-seconds** statement and specify the number of seconds (180 through 86,400) at the **[edit security ipsec proposal *ipsec-proposal-name*]** hierarchy level:

```
[edit security ipsec proposal ipsec-proposal-name]
lifetime-seconds seconds;
```

NOTE: When a dynamic 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 that the SA is about to expire. This allows the key management system to negotiate a new SA before the hard lifetime expires. When you specify the lifetime, you specify a hard lifetime.

Configuring the Protocol for a Dynamic IPsec SA

The **protocol** statement sets the protocol for a dynamic SA. 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 at the **[edit security ipsec proposal *ipsec-proposal-name*]** hierarchy level:

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

Configuring the IPsec Policy for an ES PIC

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 an IPsec 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 will match 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 the proposals in the list by listing them in the order in which the IPsec policy uses them (first to last).

To configure an IPsec policy, include the **policy** statement at the **[edit security ipsec]** hierarchy level, specifying the policy name and one or more proposals you want to associate with this policy:

```
[edit security ipsec]
policy ipsec-policy-name {
   proposals [ proposal-names ];
```

```
}
```

Configuring Perfect Forward Secrecy

PFS provides additional security by means of a Diffie-Hellman key exchange 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 security ipsec policy *ipsec-policy-name*]** hierarchy level:

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

The key can be one of the following:

- **group1**—Specify that IKE use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- **group2**—Specify that IKE use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

group2 provides more security than **group1**, but requires more processing time.

Related Documentation

- Example: Configuring an IPsec Policy on page 35
- IPsec Configuration for an ES PIC Overview on page 4

Example: Configuring an IPsec Policy

The following example shows how to configure an IPsec policy:

```
[edit security 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 ];
}
security-association dynamic-sa1 {
  dynamic {
```

```
        replay-window-size 64;
        ipsec-policy dynamic-policy-1;
    }
}
```

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 System Basics and Services Command Reference*.

Related Documentation

- Configuring the IPsec Policy for an ES PIC on page 34
- IPsec Configuration for an ES PIC Overview on page 4

Configuring Internal IPsec for Junos-FIPS

In a Junos-FIPS environment, routers with two Routing Engines must use IPsec for internal communication between the Routing Engines. You configure internal IPsec after you install Junos-FIPS. You must be a Crypto Officer to configure internal IPsec.

To configure internal IPsec, include the **security-association** statement at the **[edit security]** hierarchy level:

```
[edit security]
ipsec {
    internal {
        security-association {
            manual {
                direction (bidirectional | inbound | outbound) {
                    protocol esp;
                    spi spi-value;
                    encryption {
                        algorithm 3des-cbc;
                        key ascii-text ascii-text-string;
                    }
                }
            }
        }
    }
}
```

Tasks for configuring internal IPsec for Junos-FIPS are:

1. Configuring the SA Direction on page 37
2. Configuring the IPsec SPI on page 37
3. Configuring the IPsec Key on page 38

Configuring the SA Direction

To configure the IPsec SA direction, include the **direction** statement at the **[edit security ipsec internal security-association manual]** hierarchy level:

```
direction (bidirectional | inbound | outbound);
```

The value can be one of the following:

- **bidirectional**—Apply the same SA values in both directions between Routing Engines.
- **inbound**—Apply these SA properties only to the inbound IPsec tunnel.
- **outbound**—Apply these SA properties only to the outbound IPsec tunnel.

If you do not configure the SA to be bidirectional, you must configure SA parameters for IPsec tunnels in both directions. The following example uses an inbound and outbound IPsec tunnel:

```
[edit security]
ipsec {
  internal {
    security-association {
      manual {
        direction inbound {
          protocol esp;
          spi 512;
          encryption {
            algorithm 3des-cbc;
            key ascii-text "$.KL3rngIH7,theOPcn87lxfpe9GJKdme";
          }
        }
        direction outbound {
          protocol esp;
          spi 513;
          encryption {
            algorithm 3des-cbc;
            key ascii-text ".n87lngIH7,thxefpe9GJKdme.KL3rOPc";
          }
        }
      }
    }
  }
}
```

Configuring the IPsec SPI

A security parameter index (SPI) is a 32-bit index identifying a security context between a pair of Routing Engines. To configure the IPsec Security Parameter Index (SPI) value, include the **spi** statement at the **[edit security ipsec internal security-association manual direction]** hierarchy level:

```
spi value;
```

The value must be from **256** through **16639**.

Configuring the IPsec Key

To configure the ASCII text key, include the **key** statement at the **[edit security ipsec internal security-association manual direction encryption]** hierarchy level:

```
key ascii-text ascii-text-string;
```

The value must be from **256** through **16639**. You must enter the key ASCII value twice and the strings entered must match, or the key will not be set. The ASCII text key is never displayed in plain text.

Related Documentation

- Example: Configuring Internal IPsec on page 38

Example: Configuring Internal IPsec

Configure a bidirectional IPsec SA with an SPI value of 512 and a key value conforming to the FIPS 140-2 rules:

```
[edit security]
ipsec {
   internal {
      security-association {
         manual {
            direction bidirectional {
               protocol esp;
               spi 512;
               encryption {
                  algorithm 3des-cbc;
                  key ascii-text “$9$90j.COIek8X7VevbYgoji1rh”;
               }
            }
         }
      }
   }
}
```

Related Documentation

- Configuring Internal IPsec for Junos-FIPS on page 36

CHAPTER 5

Configuring Digital Certificates for ES and AS PICs

Configuration Statements for Setting Up Digital Certificates for an ES PIC

To define the digital certificate configuration for an encryption service interface, include the following statements at the **[edit security certificates]** and **[edit security ike]** hierarchy levels:

```
[edit security]
certificates {
    cache-size bytes;
    cache-timeout-negative seconds;
    certification-authority ca-profile-name {
        ca-name ca-identity;
        crl filename;
        encoding (binary | pem);
        enrollment-url url-name;
        file certificate-filename;
        ldap-url url-name;
    }
    enrollment-retry attempts;
```

```
    local certificate-filename {
      certificate-key-string;
      load-key-file URL key-file-name;
    }
    maximum-certificates number;
    path-length certificate-path-length;
  }
  ike {
    policy ike-peer-address {
      description policy;
      encoding (binary | pem);
      identity identity-name;
      local-certificate certificate-filename;
      local-key-pair private-public-key-file;
      mode (aggressive | main);
      pre-shared-key (ascii-text key | hexadecimal key);
      proposals [ proposal-names ];
    }
  }
```

The statements for configuring digital certificates differ for the AS and MultiServices PICs and the ES PIC.

For information about how to configure the **description** and **mode** statements, see "Configuring the Description for an IKE Policy" on page 30 and "Configuring the Mode for an IKE Policy" on page 30. For information about how to configure the IKE proposal, see "Associating Proposals with an IKE Policy" on page 31

NOTE: For digital certificates, the Junos OS supports only VeriSign CAs for the ES PIC.

Related Documentation

- Digital Certificates Overview on page 9

Obtaining a Certificate from a Certificate Authority for an ES PIC

Certificate authorities (CAs) manage certificate requests and issue certificates to participating IPsec network devices. When you create a certificate request, you need to provide the information about the owner of the certificate. The required information and its format vary across certificate authorities.

Certificates use names in the X.500 format, a directory access protocol that provides both read and update access. The entire name is called a DN (distinguished name). It consists of a set of components, which often includes a CN (common name), an organization (O), an organization unit (OU), a country (C), a locality (L), and so on.

NOTE: For the dynamic registration of digital certificates, the Junos OS supports only the Simple Certificate Enrollment Protocol (SCEP).

Related Documentation
- Digital Certificates Overview on page 9

Requesting a CA Digital Certificate for an ES PIC on an M Series or T Series Router

For an encryption interface on an M Series or T Series router, issue the following command to obtain a public key certificate from a CA. The results are saved in the specified file in the **/var/etc/ikecert** directory. The CA public key verifies certificates from remote peers.

```
user@host> request security certificate enroll filename filename ca-name ca-name
  parameters parameters
```

Related Documentation
- Example: Requesting a CA Digital Certificate on page 41
- Digital Certificates Overview on page 9

Example: Requesting a CA Digital Certificate

Specify a URL to the SCEP server and the name of the certification authority whose certificate you want: **mycompany.com**. **filename 1** is name of the file that stores the result. The output, "Received CA certificate:" provides the signature for the certificate, which allows you to verify (offline) that the certificate is genuine.

```
user@host> request security certificate enroll filename ca_verisign ca-file verisign ca-name
  xyzcompany url
http://hostname/path/filename
URL: http://hostname/path/filename name: juniper.net CA file: verisign Encoding: binary
Certificate enrollment has started. To see the certificate enrollment status, check the key
  management process (kmd) log file at /var/log/kmd. <----------
```

NOTE: Each router is initially manually enrolled with a certificate authority.

Related Documentation
- Requesting a CA Digital Certificate for an ES PIC on an M Series or T Series Router on page 41

Generating a Private and Public Key Pair for Digital Certificates for an ES PIC

To generate a private and public key, issue the following command:

```
user@host> request security key-pair name size key-size type ( rsa | dsa )
```

name specifies the filename in which to store the public and private keys.

key-size can be 512, 1024, 1596, or 2048 bytes. The default key size is 1024 bytes.

type can be **rsa** or **dsa**. The default is RSA.

NOTE: When you use SCEP, the Junos OS only supports RSA.

The following example shows how to generate a private and public key pair:

```
user@host> request security key-pair batt
Generated key pair, key size 1024, file batt Algorithm RSA
```

Related Documentation

- Digital Certificates Overview on page 9

Obtaining a Signed Certificate from the CA for an ES PIC

To obtain a signed certificate from the CA, issue the following command:

```
user@host> request security certificate enroll filename filename subject c=us,o=x
   alternative-subject certificate-ip-address certification-authority certificate-authority
   key-file key-file-name domain-name domain-name
```

The results are saved in a specified file to the **/var/etc/ikecert** directory.

The following example shows how to obtain a CA signed certificate by referencing the configured **certification-authority** statement **local** . This statement is referenced by the **request security certificate enroll filename m subject c=us,0=x alternative subject 1.1.1.1 certification-authority** command.

```
[edit]
security {
   certificates {
      certification-authority local {
         ca-name xyz.company.com;
         file l;
         enrollment-url "http://www.xyzcompany.com";
      }
   }
}
```

To obtain a signed certificate from the CA, issue the following command:

```
user@host> request security certificate enroll filename I subject c=uk,o=london
   alternative-subject 10.50.1.4 certification-authority verisign key-file host-1.prv
   domain-name host.xyzcompany.com
CA name: xyz.company.com CA file: ca_verisign
local pub/private key pair: host.prv
subject: c=uk,o=london domain name: host.juniper.net
alternative subject: 10.50.1.4
Encoding: binary
Certificate enrollment has started. To see the certificate enrollment status, check the key
   management process (kmd) log file at /var/log/kmd. <--------------
```

For information about how to use the operational mode commands to obtain a signed certificate, see the *Junos OS System Basics and Services Command Reference*.

Another way to obtain a signed certificate from the CA is to reference the configured statements such as the URL, CA name, and CA certificate file by means of the **certification-authority** statement:

```
user@host> request security certificate enroll filename m subject c=us ,o=x
   alternative-subject 1.1.1.1 certification-authority local key-file y domain-name
   abc.company.com
```

Related Documentation
- Digital Certificates Overview on page 9

Configuring Digital Certificates for an ES PIC

Digital certificates provide a way of authenticating users through a trusted third party called a certificate authority (CA). The CA validates the identity of a certificate holder and "signs" the certificate to attest that it has not been forged or altered.

To define the digital certificate configuration for an encryption service interface, include the following statements at the **[edit security certificates]** and **[edit security ike]** hierarchy levels:

```
[edit security]
certificates {
    cache-size bytes;
    cache-timeout-negative seconds;
    certification-authority ca-profile-name {
        ca-name ca-identity;
        crl filename;
        encoding (binary | pem);
        enrollment-url url-name;
        file certificate-filename;
        ldap-url url-name;
    }
    enrollment-retry attempts;
    local certificate-filename {
        certificate-key-string;
        load-key-file URL key-file-name;
    }
    maximum-certificates number;
    path-length certificate-path-length;
}
ike {
    policy ike-peer-address {
        description policy;
        encoding (binary | pem);
        identity identity-name;
        local-certificate certificate-filename;
        local-key-pair private-public-key-file;
        mode (aggressive | main);
        pre-shared-key (ascii-text key | hexadecimal key);
        proposals [ proposal-names ];
    }
}
```

Tasks to configure digital certificates for ES PICs are:

- Configuring the Certificate Authority Properties for an ES PIC on page 44
- Configuring the Cache Size on page 46
- Configuring the Negative Cache on page 46
- Configuring the Number of Enrollment Retries on page 46

- Configuring the Maximum Number of Peer Certificates on page 47
- Configuring the Path Length for the Certificate Hierarchy on page 47

Configuring the Certificate Authority Properties for an ES PIC

A CA is a trusted third-party organization that creates, enrolls, validates, and revokes digital certificates.

To configure a certificate authority and its properties for an ES PIC, include the following statements at the **[edit security certificates]** hierarchy level:

```
[edit security certificates]
certification-authority ca-profile-name {
    ca-name ca-identity;
    crl filename;
    encoding (binary | pem);
    enrollment-url url-name;
    file certificate-filename;
    ldap-url url-name;
}
```

ca-profile-name is the CA profile name.

Tasks for configuring the CA properties are:

1. Specifying the Certificate Authority Name on page 44
2. Configuring the Certificate Revocation List on page 44
3. Configuring the Type of Encoding Your CA Supports on page 45
4. Specifying an Enrollment URL on page 45
5. Specifying a File to Read the Digital Certificate on page 45
6. Specifying an LDAP URL on page 45

Specifying the Certificate Authority Name

If you are enrolling with a CA using simple certificate enrollment protocols (SCEP), you need to specify the CA name (CA identity) that is used in the certificate request, in addition to the URL for the SCEP server.

To specify the name of the CA identity, include the **ca-name** statement at the **[edit security certificates certification-authority *ca-profile-name*]** hierarchy level:

```
[edit security certificates certification-authority ca-profile-name]
ca-name ca-identity;
```

ca-identity specifies the CA identity to use in the certificate request. It is typically the CA domain name.

Configuring the Certificate Revocation List

A certificate revocation list (CRL) contains a list of digital certificates that have been canceled 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.

To configure the CA certificate revocation list, include the **crl** statement and specify the file from which to read the CRL at the **[edit security certificates certification-authority *ca-profile-name*]** hierarchy level:

```
[edit security certificates certification-authority ca-profile-name]
crl filename;
```

Configuring the Type of Encoding Your CA Supports

By default, encoding is set to binary. Encoding specifies the file format used for the **local-certificate** and **local-key-pair** statements. By default, the binary (distinguished encoding rules) format is enabled. Privacy-enhanced mail (PEM) is an ASCII base 64 encoded format. Check with your CA to determine which file formats it supports.

To configure the file format that your CA supports, include the **encoding** statement and specify a binary or PEM format at the **[edit security certificates certification-authority *ca-profile-name*]** hierarchy level:

```
[edit security certificates certification-authority ca-profile-name]
encoding (binary | pem);
```

Specifying an Enrollment URL

You specify the CA location where your router or switch sends SCEP-based certificate enrollment requests. To specify the CA location by naming the CA URL, include the **enrollment-url** statement at the **[edit security certificates certification-authority *ca-profile-name*]** hierarchy level:

```
[edit security certificates certification-authority ca-profile-name]
enrollment-url url-name;
```

url-name is the CA location. The format is **http://*ca-name***, where ***ca-name*** is the CA host DNS name or IP address.

Specifying a File to Read the Digital Certificate

To specify the file from which to read the digital certificate, include the **file** statement and specify the certificate filename at the **[edit security certificates certification-authority *ca-profile-name*]** hierarchy level:

```
[edit security certificates certification-authority ca-profile-name]
file certificate-filename;
```

Specifying an LDAP URL

If your CA stores its current CRL at its Lightweight Directory Access Protocol (LDAP) server, you can optionally check your CA CRL list before using a digital certificate. If the digital certificate appears on the CA CRL, your router or switch cannot use it. To access your CA CRL, include the **ldap-url** statement at the **[edit security certificates certification-authority *ca-profile-name*]** hierarchy level:

```
[edit security certificates certification-authority ca-profile-name]
ldap-url url-name;
```

url-name is the certification authority LDAP server name. The format is **ldap://*server-name*,** where ***server-name*** is the CA host DNS name or IP address.

Configuring the Cache Size

By default, the cache size is 2 megabytes (MB). To configure total cache size for digital certificates, include the **cache-size** statement at the **[edit security certificates]** hierarchy level:

```
[edit security certificates]
cache-size bytes;
```

bytes is the cache size for digital certificates. The range can be from 64 through 4,294,967,295 bytes.

NOTE: We recommend that you limit your cache size to 4 MB.

Configuring the Negative Cache

Negative caching stores negative results and reduces the response time for negative answers. It also reduces the number of messages that are sent to the remote server. Maintaining a negative cache state allows the system to quickly return a failure condition when a lookup attempt is retried. Without a negative cache state, a retry would require waiting for the remote server to fail to respond, even though the system already " knows" that remote server is not responding.

By default, the negative cache is 20 seconds. To configure the negative cache, include the **cache-timeout-negative** statement at the **[edit security certificates]** hierarchy level:

```
[edit security certificates]
cache-timeout-negative seconds;
```

seconds is the amount of time for which a failed CA or router certificate is present in the negative cache. While searching for certificates with a matching CA identity (domain name for certificates or CA domain name and serial for CRLs), the negative cache is searched first. If an entry is found in the negative cache, the search fails immediately.

NOTE: Configuring a large negative cache value can make you susceptible to a denial-of-service (DoS) attack.

Configuring the Number of Enrollment Retries

By default, the number of enrollment retries is set to 0, an infinite number of retries. To specify how many times a router or switch will resend a certificate request, include the **enrollment-retry** statement at the **[edit security certificates]** hierarchy level:

```
[edit security certificates]
enrollment-retry attempts;
```

attempts is the number of enrollment retries (0 through 100).

Configuring the Maximum Number of Peer Certificates

By default, the maximum number of peer certificates to be cached is 1024. To configure the maximum number of peer certificates to be cached, include the **maximum-certificates** statement at the **[edit security certificates]** hierarchy statement level:

```
[edit security certificates]
maximum-certificates number;
```

number is the maximum number of peer certificates to be cached. The range is from 64 through 4,294,967,295 peer certificates.

Configuring the Path Length for the Certificate Hierarchy

Certification authorities can issue certificates to other CAs. This creates a tree-like certification hierarchy. The highest trusted CA in the hierarchy is called the *trust anchor*. Sometimes the trust anchor is the root CA, which is usually signed by itself. In the hierarchy, every certificate is signed by the CA immediately above it. An exception is the root CA certificate, which is usually signed by the root CA itself. In general, a chain of multiple certificates may be needed, comprising a certificate of the public key owner (the end entity) signed by one CA, and zero or more additional certificates of CAs signed by other CAs. Such chains, called certification paths, are required because a public key user is only initialized with a limited number of assured CA public keys.

Path length refers to a path of certificates from one certificate to another certificate, based on the relationship of a CA and its "children." When you configure the **path-length** statement, you specify the maximum depth of the hierarchy to validate a certificate from the trusted root CA certificate to the certificate in question. For more information about the certificate hierarchy, see RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

By default, the maximum certificate path length is set to 15. The root anchor is 1.

To configure path length, include the **path-length** statement at the **[edit security certificates]** hierarchy level:

```
[edit security certificates]
path-length certificate-path-length;
```

certificate-path-length is the maximum number certificates for the certificate path length. The range is from 2 through 15 certificates.

Configuring an IKE Policy for Digital Certificates for an ES PIC

An IKE policy for digital certificates 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. 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.

To configure an IKE policy for digital certificates for an ES PIC, include the following statements at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike]
policy ike-peer-address{
   encoding (binary | pem);
   identity identity-name;
   local-certificate certificate-filename;
   local-key-pair private-public-key-file;
}
```

Tasks for configuring an IKE policy for digital certificates are:

1. Configuring the Type of Encoding Your CA Supports on page 48
2. Configuring the Identity to Define the Remote Certificate Name on page 48
3. Specifying the Certificate Filename on page 48
4. Specifying the Private and Public Key File on page 48

Configuring the Type of Encoding Your CA Supports

By default, the encoding is set to binary. Encoding specifies the file format used for the **local-certificate** and **local-key-pair** statements. By default, the binary (distinguished encoding rules) format is enabled. PEM is an ASCII base 64 encoded format. Check with your CA to determine which file formats it supports.

To configure the file format that your CA supports, include the **encoding** statement and specify a binary or PEM format at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike policy ike-peer-address ]
encoding (binary | pem);
```

Configuring the Identity to Define the Remote Certificate Name

To define the remote certificate name, include the **identity** statement at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike policy ike-peer-address]
identity identity-name;
```

identity-name defines the identity of the remote certificate name if the identity cannot be learned through IKE (ID payload or IP address).

Specifying the Certificate Filename

To configure the certificate filename from which to read the local certificate, include the **local-certificate** statement at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike policy ike-peer-address]
local-certificate certificate-filename;
```

certificate-filename specifies the file from which to read the local certificate.

Specifying the Private and Public Key File

To specify the filename from which to read the public and private key, include the **local key-pair** statement at the **[edit security ike policy *ike-peer-address*]** hierarchy level:

```
[edit security ike policy ike-peer-address ]
local-key-pair private-public-key-file;
```

private-public-key-file specifies the file from which to read the pair key.

Associating the Configured Security Association with a Logical Interface

Configuring the ES PIC associates the configured SA with a logical interface. This configuration defines the tunnel itself (logical subunit, tunnel addresses, maximum transmission unit [MTU], optional interface addresses, and the name of the SA to apply to traffic).

The addresses configured as the tunnel source and destination are the addresses in the outer IP header of the tunnel.

NOTE: **The tunnel source address must be configured locally on the router, and the tunnel destination address must be a valid address for the security gateway terminating the tunnel.**

The M5, M10, M20, and M40 routers support the ES PIC.

The SA must be a valid tunnel-mode SA. The interface address and destination address listed are optional. The destination address allows the user to configure a static route to encrypt traffic. If a static route uses that destination address as the next hop, traffic is forwarded through the portion of the tunnel in which encryption occurs.

The following example shows how to configure an IPsec tunnel as a logical interface on the ES PIC. The logical interface specifies the tunnel through which the encrypted traffic travels. The **ipsec-sa** statement associates the security profile with the interface.

```
[edit interfaces]
es-0/0/0 {
  unit 0 {
    tunnel {
      source tunnel 10.5.5.5; # tunnel source address
      destination 10.6.6.6; # tunnel destination address
    }
    family inet {
      ipsec-sa ipsec-sa; # name of security association to apply to packet
      address 10.1.1.8/32 { # local interface address inside local VPN
      destination 10.2.2.254; # destination address inside remote VPN
    }
  }
}
```

Related Documentation

- Configuring Security Associations for IPsec on an ES PIC on page 15

Configuring Digital Certificates for Adaptive Services Interfaces

A digital certificate implementation uses the public key infrastructure (PKI), which requires that you generate a key pair consisting of a public key and a private key. The keys are

created with a random number generator and are used to encrypt and decrypt data. In networks that do not use digital certificates, an IPsec-enabled device encrypts data with the private key and IPsec peers decrypt the data with the public key.

With digital certificates, the key sharing process requires an additional level of complexity. First, you and your IPsec peers request that a certificate authority (CA) send you a CA certificate that contains the public key of the CA. Next you request the CA to assign you a local digital certificate that contains the public key and some additional information. When the CA processes your request, it signs your local certificate with the private key of the CA. Then you install the CA certificate and the local certificate in your router and load the CA in remote devices before you can establish IPsec tunnels with your peers.

NOTE: For digital certificates, the Junos OS supports VeriSign, Entrust, Cisco Systems, and Microsoft Windows CAs for the Adaptive Services (AS) and Multiservices PICs.

To define digital certificates configuration for J Series Services Routers and AS and Multiservices PICs installed on M Series and T Series routers, include the following statements at the **[edit security pki]** hierarchy level:

```
[edit security]
pki {
    ca-profile ca-profile-name {
        ca-identity ca-identity;
        enrollment {
            url-name;
            retry number-of-enrollment-attempts;
            retry-interval seconds;
        }
        revocation-check {
            disable;
            crl {
                disable on-download-failure;
                refresh-interval number-of-hours;
                url {
                    url-name;
                    password;
                }
            }
        }
    }
}
```

The following tasks enable you to implement digital certificates on J Series Services Routers and AS and Multiservices PICs installed on M Series and T Series routers:

1. Configuring the Certificate Authority Properties on page 51
2. Configuring the Certificate Revocation List on page 52
3. Managing Digital Certificates on page 53
4. Configuring the Auto-Reenrollment Properties for Automatic Renewal of the Router Certificate from the CA on page 55

Configuring the Certificate Authority Properties

A CA is a trusted third-party organization that creates, enrolls, validates, and revokes digital certificates.

To configure a certificate authority and its properties for the AS and Multiservices PICs, include the following statements at the **[edit security pki]** hierarchy level:

```
[edit security pki]
ca-profile ca-profile-name {
    ca-identity ca-identity;
    enrollment {
        url url-name;
        retry number-of-attempts;
        retry-interval seconds;
    }
}
```

Tasks for configuring the Certificate Authority properties are:

1. Specifying the CA Profile Name on page 51
2. Specifying an Enrollment URL on page 51
3. Specifying the Enrollment Properties on page 52

Specifying the CA Profile Name

The CA profile contains the name and URL of the CA or RA, as well as some retry-timer settings. CA certificates issued by Entrust, VeriSign, Cisco Systems, and Microsoft are compatible with the J Series Services Routers and AS and Multiservices PICs installed in the M Series and T Series routers.

To specify the CA profile name, include the **ca-profile statement** at the **[edit security pki]** security level:

```
[edit security pki]
ca-profile ca-profile-name;
```

You also need to specify the name of the CA identity used in the certificate request. This name is typically the domain name. To specify the name of the CA identity, include the **ca-identity** statement at the **[edit security pki ca-profile *ca-profile-name*]** level:

```
[edit security pki ca-profile ca-profile-name]
ca-identity ca-identity;
```

Specifying an Enrollment URL

You specify the CA location where your router should send the SCEP-based certificate enrollment requests. To specify the CA location by naming the CA URL, include the **url** statement at the **[edit security pki enrollment]** hierarchy level:

```
[edit security pki ca-profile ca-profile-name enrollment]
url url-name;
```

url-name is the CA location. The format is **http://*CA_name***, where ***CA_name*** is the CA host DNS name or IP address.

Specifying the Enrollment Properties

You can specify the number of times a router will resend a certificate request and the amount of time, in seconds, the router should wait between enrollment attempts.

By default, the number of enrollment retries is set to 0, an infinite number of retries. To specify how many times a router will resend a certificate request, include the **retry** ***number-of-attempts*** statement at the **[edit security pki ca-profile** ***ca-profile-name*** **enrollment]** hierarchy level:

```
[edit security pki ca-profile ca-profile-name enrollment]
retry number-of-attempts;
```

The range for ***number-of-attempts*** is from 0 through 100.

To specify the amount of time, in seconds, that a router should wait between enrollment attempts, include the **retry-interval** ***seconds*** statement at the **[edit security pki ca-profile** ***ca-profile-name*** **enrollment]** hierarchy level:

```
[edit security pki ca-profile ca-profile-name enrollment]
retry-interval seconds;
```

The range for ***seconds*** is from 0 through 3600.

Configuring the Certificate Revocation List

Tasks to configure the certificate revocation list are:

1. Specifying an LDAP URL on page 52
2. Configuring the Interval Between CRL Updates on page 53
3. Overriding Certificate Verification if CRL Download Fails on page 53

Specifying an LDAP URL

You can specify the URL for the Lightweight Directory Access Protocol (LDAP) server where your CA stores its current CRL. If the CA includes the Certificate Distribution Point (CDP) in the digital certificate, you do not need to specify a URL for the LDAP server. The CDP is a field within the certificate that contains information about how to retrieve the CRL for the certificate. The router uses this information to download the CRL automatically.

Configure an LDAP URL if you want to use a different CDP from the one specified in the certificate. Any LDAP URL you configure takes precedence over the CDP included in the certificate.

You can configure up to three URLs for each CA profile.

If the LDAP server requires a password to access the CRL, you need to include the **password** statement.

To configure the router to retrieve the CRL from the LDAP server, include the **url** statement and specify the URL name at the **[edit security pki ca-profile** ***ca-profile-name*** **revocation-check crl]** hierarchy level:

```
[edit security pki ca-profile ca-profile-name revocation-check crl]
url {
  url-name;
}
```

url-name is the certificate authority LDAP server name. The format is **ldap://*server-name***, where ***server-name*** is the CA host DNS name or IP address.

To specify to use a password to access the CRL, include the **password** statement at the **[edit security pki ca-profile *ca-profile-name* revocation-check crl url]** hierarchy level:

```
[edit security pki ca-profile ca-profile-name revocation-check crl url]
password password;
```

password is the secret password that the LDAP server requires for access.

Configuring the Interval Between CRL Updates

By default, the time interval between CRL updates is 24 hours. To configure the amount of time between CRL updates, include the **refresh-interval** statement at the **[edit security pki ca-profile *ca-profile-name* revocation-check crl]** hierarchy level:

```
[edit security pki ca-profile ca-profile-name revocation-check crl]
refresh-interval number-of-hours;
```

The range for number of hours is from 0 through 8784.

Overriding Certificate Verification if CRL Download Fails

By default, if the router either cannot access the 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:

```
[edit security pki ca-profile ca-profile-name revocation-check crl]
disable on-download-failure;
```

Managing Digital Certificates

After you configure the CA profile, you can request a CA certificate from the trusted CA. Next, you must generate a public/private key pair. When the key pair is available, you can generate a local certificate either online or manually.

Tasks to manage digital certificates are:

1. Requesting a CA Digital Certificate for AS and Multiservices PICs installed on M Series and T Series Routers on page 54
2. Generating a Public/Private Key Pair on page 54
3. Generating and Enrolling a Local Digital Certificate on page 54

Requesting a CA Digital Certificate for AS and Multiservices PICs installed on M Series and T Series Routers

For J Series Services Routers and AS and Multiservices PICs installed on M Series and T Series routers, issue the following command to obtain a digital certificate from a CA. Specify a configured ***ca-profile-name*** to request a CA certificate from the trusted CA.

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

For information about how to configure a CA profile, see "Configuring the Certificate Authority Properties" on page 51.

In this example, the certificate is enrolled online and installed into the router automatically.

```
user@host> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
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
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, CN=First Officer
 Fingerprint:46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or Web site download), you can install it with the **request security pki ca-certificate load** command. For more information, see the *Junos OS System Basics and Services Command Reference*.

Generating a Public/Private Key Pair

After obtaining a certificate for an AS PIC or Multiservices PIC, you must generate a public-private key before you can generate a local certificate. The public key is included in the local digital certificate and the private key is used to decrypt data received from peers. To generate a public-private key pair, issue the **request security pki generate-key-pair certificate-id *certificate-id-name*** command.

The following example shows how to generate a public-private key for an AS PIC or Multiservices PIC:

```
user@host>request security pki generate-key-pair certificate-id local-entrust2
Generated key pair local-entrust2, key size 1024 bits
```

Generating and Enrolling a Local Digital Certificate

You can generate and enroll local digital certificates either online or manually. To generate and enroll a local certificate online by using the Simple Certificate Enrollment Protocol (SCEP) for an AS PIC or Multiservices PIC, issue the **request security pki local-certificate enroll** command. To generate a local certificate request manually in the PKCS-10 format, issue the **request security pki generate-certificate-request** command.

If you create the local certificate request manually, you must also load the certificate manually. To manually install a certificate in your router, issue the **request security pki local-certificate load** command.

The following example shows how to generate a local certificate request manually and send it to the CA for processing:

```
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+1BPNbCxNWYvyrnSyVYDbFj8oOXyqog8ACDfVL2JBWrPNBYy7imq/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)
```

The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the router and load the certificate:

```
user@host> request security pki local-certificate load filename /tmp/router2-cert
  certificate-id local-entrust2
Local certificate local-entrust2 loaded successfully
```

NOTE: The name of the file sent to you by the CA might not match the name of the certificate identifier. However, the certificate-id name must always match the name of the key pair you generated for the router.

After the local and CA certificates have been loaded, you can reference them in your IPsec configuration. Using default values in the AS and Multiservices PICs, you do not need to configure an IPsec proposal or an IPsec policy. However, you must configure an IKE proposal that specifies the use of digital certificates, reference the IKE proposal and locate the certificate in an IKE policy, and apply the CA profile to the service set.

Configuring the Auto-Reenrollment Properties for Automatic Renewal of the Router Certificate from the CA

Use the **auto-re-enrollment** statement to configure automatic reenrollment of a specified existing router certificate before its existing expiration date. This function automatically reenrolls the router certificate. The reenrollment process requests the certificate authority (CA) to issue a new router certificate with a new expiration date. The date of auto-reenrollment is determined by the following parameters:

- **re-enroll-trigger-time**—The percentage of the difference between the router certificate start date/time (when the certificate was generated) and the validity period; used to specify how long auto-reenrollment should be initiated before expiration.
- **validity-period**—The number of days after issuance when the router certificate will expire, as set when a certificate is generated.

NOTE: By default, this feature is not enabled unless configured explicitly. This means that a certificate that does not have auto-reenrollment configured will expire on its normal expiration date.

The **ca-profile** statement specifies which CA will be contacted to reenroll the expiring certificate. This is the CA that issued the original router certificate.

The **challenge-password** statement provides the issuing CA with the router certificate's password, as set by the administrator and normally obtained from the SCEP enrollment Web page of the CA. The password is 16 characters in length.

Optionally, the router certificate key pair can be regenerated by using the **re-generate-keypair** statement.

To configure automatic reenrollment properties, include the following statements at the **[edit security pki]** hierarchy level:

```
[edit security pki]
auto-re-enrollment {
   certificate-id {
      ca-profile ca-profile-name;
      challenge-password password;
      re-enroll-trigger-time-percentage percentage;
      re-generate-keypair;
      validity-period days;
   }
}
```

percentage is the percentage for the reenroll trigger time. The range can be from 1 through 99 percent.

days is the number of days for the validity period. The range can be from 1 through 4095.

Tasks to configure automatic reenrollment of certificates are:

1. Specify the Certificate ID on page 57
2. Specify the CA Profile on page 57
3. Specify the Challenge Password on page 57
4. Specify the Reenroll Trigger Time on page 57
5. Specify the Regenerate Key Pair on page 57
6. Specify the Validity Period on page 58

Specify the Certificate ID

Use the **certificate-id** statement to specify the name of the router certificate to configure for auto-reenrollment. To specify the certificate ID, include the statement at the **[edit security pki auto-re-enrollment]** hierarchy level:

```
[edit security pki auto-re-enrollment]
certificate-id certificate-name;
```

Specify the CA Profile

Use the **ca-profile** statement to specify the name of the CA profile from the router certificate previously specified by certificate ID. To specify the CA profile, include the statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
ca-profile ca-profile-name;
```

NOTE: The referenced **ca-profile** must have an enrollment URL configured at the **[edit security pki ca-profile *ca-profile-name* enrollment url]** hierarchy level.

Specify the Challenge Password

The challenge password is used by the CA specified by the PKI certificate ID for reenrollment and revocation. To specify the challenge password, include the following statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
challenge-password password;
```

Specify the Reenroll Trigger Time

Use the **re-enroll-trigger-time** statement to set the percentage of the validity period before expiration at which reenrollment occurs. To specify the reenroll trigger time, include the following statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
re-enroll-trigger-time percentage;
```

percentage is the percentage for the reenroll trigger time. The range can be from 1 through 99 percent.

Specify the Regenerate Key Pair

When a regenerate key pair is configured, a new key pair is generated during reenrollment. On successful reenrollment, a new key pair and new certificate replace the old certificate and key pair. To generate a new key pair, include the following statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
re-generate-keypair;
```

Specify the Validity Period

The **validity-period** statement specifies the router certificate validity period, in number of days, that the specified router certificate remains valid. To specify the validity period, include the statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
validity-period days;
```

days is the number of days for the validity period. The range can be from 1 through 4095.

Related Documentation

- Digital Certificates Overview on page 9
- Configuring Digital Certificates for an ES PIC on page 43

Configuring the Auto-Reenrollment Properties for Automatic Renewal of the Router Certificate from the CA

Use the **auto-re-enrollment** statement to configure automatic reenrollment of a specified existing router certificate before its existing expiration date. This function automatically reenrolls the router certificate. The reenrollment process requests the certificate authority (CA) to issue a new router certificate with a new expiration date. The date of auto-reenrollment is determined by the following parameters:

- **re-enroll-trigger-time**—The percentage of the difference between the router certificate start date/time (when the certificate was generated) and the validity period; used to specify how long auto-reenrollment should be initiated before expiration.
- **validity-period**—The number of days after issuance when the router certificate will expire, as set when a certificate is generated.

NOTE: **By default, this feature is not enabled unless configured explicitly. This means that a certificate that does not have auto-reenrollment configured will expire on its normal expiration date.**

The **ca-profile** statement specifies which CA will be contacted to reenroll the expiring certificate. This is the CA that issued the original router certificate.

The **challenge-password** statement provides the issuing CA with the router certificate's password, as set by the administrator and normally obtained from the SCEP enrollment Web page of the CA. The password is 16 characters in length.

Optionally, the router certificate key pair can be regenerated by using the **re-generate-keypair** statement.

To configure automatic reenrollment properties, include the following statements at the **[edit security pki]** hierarchy level:

```
[edit security pki]
auto-re-enrollment {
  certificate-id {
    ca-profile ca-profile-name;
    challenge-password password;
    re-enroll-trigger-time-percentage percentage;
    re-generate-keypair;
    validity-period days;
  }
}
```

percentage is the percentage for the reenroll trigger time. The range can be from 1 through 99 percent.

days is the number of days for the validity period. The range can be from 1 through 4095.

Tasks to configure automatic reenrollment of certificates are:

1. Specify the Certificate ID on page 59
2. Specify the CA Profile on page 59
3. Specify the Challenge Password on page 60
4. Specify the Reenroll Trigger Time on page 60
5. Specify the Regenerate Key Pair on page 60
6. Specify the Validity Period on page 60

Specify the Certificate ID

Use the **certificate-id** statement to specify the name of the router certificate to configure for auto-reenrollment. To specify the certificate ID, include the statement at the **[edit security pki auto-re-enrollment]** hierarchy level:

```
[edit security pki auto-re-enrollment]
certificate-id certificate-name;
```

Specify the CA Profile

Use the **ca-profile** statement to specify the name of the CA profile from the router certificate previously specified by certificate ID. To specify the CA profile, include the statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
ca-profile ca-profile-name;
```

NOTE: The referenced **ca-profile** must have an enrollment URL configured at the **[edit security pki ca-profile *ca-profile-name* enrollment url]** hierarchy level.

Specify the Challenge Password

The challenge password is used by the CA specified by the PKI certificate ID for reenrollment and revocation. To specify the challenge password, include the following statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
challenge-password password;
```

Specify the Reenroll Trigger Time

Use the **re-enroll-trigger-time** statement to set the percentage of the validity period before expiration at which reenrollment occurs. To specify the reenroll trigger time, include the following statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
re-enroll-trigger-time percentage;
```

percentage is the percentage for the reenroll trigger time. The range can be from 1 through 99 percent.

Specify the Regenerate Key Pair

When a regenerate key pair is configured, a new key pair is generated during reenrollment. On successful reenrollment, a new key pair and new certificate replace the old certificate and key pair. To generate a new key pair, include the following statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
re-generate-keypair;
```

Specify the Validity Period

The **validity-period** statement specifies the router certificate validity period, in number of days, that the specified router certificate remains valid. To specify the validity period, include the statement at the **[edit security pki auto-re-enrollment certificate-id *certificate-name*]** hierarchy level:

```
[edit security pki auto-re-enrollment certificate-id certificate-name]
validity-period days;
```

days is the number of days for the validity period. The range can be from 1 through 4095.

CHAPTER 6

Configuring Traffic Filters and Tracing Operations

Example: Configuring an Outbound Traffic Filter

Firewall filters for outbound traffic direct the traffic through the desired IPsec tunnel and ensure that the tunneled traffic goes out the appropriate interface (see Figure 1 on page 5). Here, an outbound firewall filter is created on security Gateway A; it identifies the traffic to be encrypted and adds it to the input side of the interface that carries the internal VPN traffic:

```
[edit firewall]
filter ipsec-encrypt-policy-filter {
   term term1 {
      from {
         source-address { # local network
         10.1.1.0/24;
      }
      destination-address { # remote network
      10.2.2.0/24;
   }
}
then ipsec-sa manual-sa1; # apply SA name to packet
term default {
   then accept;
}
```

NOTE: The source address, port, and protocol on the outbound traffic filter must match the destination address, port, and protocol on the inbound traffic filter. The destination address, port, and protocol on the outbound traffic filter must match the source address, port, and protocol on the inbound traffic filter.

Related Documentation

- Example: Applying an Outbound Traffic Filter on page 62
- IPsec Tunnel Traffic Configuration Overview on page 4

Example: Applying an Outbound Traffic Filter

After you have configured the outbound firewall filter, you apply it:

```
[edit interfaces]
fe-0/0/1 {
  unit 0 {
    family inet {
      filter {
        input ipsec-encrypt-policy-filter;
      }
      address 10.1.1.254/24;
    }
  }
}
```

The outbound filter is applied on the Fast Ethernet interface at the **[edit interfaces fe-0/0/1 unit 0 family inet]** hierarchy level. Any packet matching the IPsec action term (**term 1**) on the input filter (**ipsec-encrypt-policy-filter**), configured on the Fast Ethernet interface, is directed to the ES PIC interface at the **[edit interfaces es-0/1/0 unit 0 family inet]** hierarchy level. If a packet arrives from the source address **10.1.1.0/24** and goes to the destination address **10.2.2.0/24**, the Packet Forwarding Engine directs the packet to the ES PIC interface, which is configured with the **manual-sa1** SA. The ES PIC receives the packet, applies the **manual-sa1** SA, and sends the packet through the tunnel.

The router must have a route to the tunnel endpoint; add a static route if necessary.

Related Documentation

- IPsec Tunnel Traffic Configuration Overview on page 4

Example: Configuring an Inbound Traffic Filter for a Policy Check

Here, an inbound firewall filter, which performs the final IPsec policy check, is created on security Gateway A. This check ensures that only packets that match the traffic configured for this tunnel are accepted.

```
filter ipsec-decrypt-policy-filter {
  term term1 { # perform policy check
  from {
    source-address { # remote network
```

```
    10.2.2.0/24;
  }
  destination-address { # local network
  10.1.1.0/24;
}
then accept;
```

Related Documentation

- IPsec Tunnel Traffic Configuration Overview on page 4

Example: Applying an Inbound Traffic Filter to an ES PIC for a Policy Check

After you create the inbound firewall filter, apply it to the ES PIC. Here, the inbound firewall filter (**ipsec-decrypt-policy-filter**) is applied on the decrypted packet to perform the final policy check. The IPsec **manual-sa1** SA is referenced at the **[edit interfaces es-1/2/0 unit 0 family inet]** hierarchy level and decrypts the incoming packet.

```
[edit interfaces]
es-1/2/0 {
  unit 0 {
    tunnel {
      source 10.5.5.5; # tunnel source address
      destination 10.6.6.6; # tunnel destination address
    }
    family inet {
      filter {
        input ipsec-decrypt-policy-filter;
      }
    ipsec-sa manual-sa1; # SA name applied to packet
      address 10.1.1.8/32 { # local interface address inside local VPN
        destination 10.2.2.254; # destination address inside remote VPN
      }
  }
}
```

The Packet Forwarding Engine directs IPsec packets to the ES PIC. It uses the packet's SPI, protocol, and destination address to look up the SA configured on one of the ES interfaces. The IPsec **manual-sa1** SA is referenced at the **[edit interfaces es-1/2/0 unit 0 family inet]** hierarchy level and is used to decrypt the incoming packet. When the packets are processed (decrypted, authenticated, or both), the input firewall filter (**ipsec-decrypt-policy-filter**) is applied on the decrypted packet to perform the final policy check. Term1 defines the decrypted (and verified) traffic and performs the required policy check.

NOTE: The inbound traffic filter is applied after the ES PIC has processed the packet, so the decrypted traffic is defined as any traffic that the remote gateway is encrypting and sending to this router. IKE uses this filter to determine the policy required for a tunnel. This policy is used during the negotiation with the remote gateway to find the matching SA configuration.

Related Documentation
- IPsec Tunnel Traffic Configuration Overview on page 4

Configuring Tracing Operations for Security Services

To configure trace options for security services, specify flags using the **traceoptions** statement:

```
[edit security]
traceoptions {
   file filename <files number> <size size>;
   flag all;
   flag database;
   flag general;
   flag ike;
   flag parse;
   flag policy-manager;
   flag routing-socket;
   flag timer;
}
```

You can include these statements at the following hierarchy levels:

- **[edit security]**
- **[edit services ipsec-vpn]**

You can specify one or more of the following security tracing flags:

- **all**—Trace all security events
- **database**—Trace database events
- **general**—Trace general events
- **ike**—Trace IKE module processing
- **parse**—Trace configuration processing
- **policy-manager**—Trace policy manager processing
- **routing-socket**—Trace routing socket messages
- **timer**—Trace internal timer events

Related Documentation
- Configuring Tracing Operations for IPsec Events for Adaptive Services PICs on page 64
- Security Associations Overview on page 7

Configuring Tracing Operations for IPsec Events for Adaptive Services PICs

To configure trace options to trace IPsec events for Adaptive Services PICs, include the following statements at the **[edit services ipsec-vpn]** hierarchy level:

```
[edit services ipsec-vpn]
traceoptions {
```

```
    file filename <files number> <size size>;
    flag all;
    flag database;
    flag general;
    flag ike;
    flag parse;
    flag policy-manager;
    flag routing-socket;
    flag timer;
}
```

Trace option output is recorded in the **/var/log/kmd** file.

You can specify one or more of the following security tracing flags:

- **all**—Trace all security events
- **database**—Trace database events
- **general**—Trace general events
- **ike**—Trace IKE module processing
- **parse**—Trace configuration processing
- **policy-manager**—Trace policy manager processing
- **routing-socket**—Trace routing socket messages
- **timer**—Trace internal timer events

Related Documentation

- Configuring Tracing Operations for Security Services on page 64

CHAPTER 7

Configuring Authentication Key Updates

Configuring the Authentication Key Update Mechanism for BGP and LDP Routing Protocols

You can configure an authentication key update mechanism for the Border Gateway Protocol (BGP) and Label Distribution Protocol (LDP) routing protocols. This mechanism allows you to update authentication keys without interrupting associated routing and signaling protocols such as Open Shortest Path First (OSPF) and Resource Reservation Setup Protocol (RSVP).

To configure this feature, include the **authentication-key-chains** statement at the **[edit security]** level, and include the **authentication-key-chain** statement for the BGP or LDP routing protocols at the **[edit protocols]** level.

The following topics provide more details about configuring authentication key updates for BGP and LDP Routing Protocols:

Configuring Authentication Key Updates

To configure the authentication key update mechanism, include the **key-chain** statement at the **[edit security authentication-key-chains]** hierarchy level, and specify the **key** option to create a keychain consisting of several authentication keys.

```
[edit security authentication-key-chains]
key-chain key-chain-name {
    key key {
        secret secret-data;
        start-time yyyy-mm-dd.hh:mm:ss;
    }
}
```

key-chain—Assigns a name to the keychain mechanism. This name is also configured at the **[edit protocols bgp]** or the **[edit protocols ldp]** hierarchy levels to associate unique **authentication key-chain** attributes as specified using the following options:

- **key**—Each key within a keychain is identified by a unique integer value. The range is from 0 through 63.
- **secret**—Each key must specify a secret in encrypted text or plain text format. Even if you enter the secret data in plain-text format, the secret always appears in encrypted format.
- **start-time**—Start times for authentication key updates are specified in UTC (Coordinated Universal Time), and must be unique within the keychain.

Configuring BGP and LDP for Authentication Key Updates

To configure the authentication key update mechanism for the BGP and LDP routing protocols, include the **authentication-key-chain** statement at the **[edit protocols (bgp | ldp)]** hierarchy level to associate each routing protocol with the **[edit security authentication-key-chains]** authentication keys.

```
[edit protocols (bgp | ldp)]
group group-name {
   neighbor address {
      authentication-key-chain key-chain-name;
   }
}
```

NOTE: When configuring the authentication key update mechanism for BGP, you cannot commit the 0.0.0.0/allow statement with authentication keys or key chains. The CLI issues a warning and fails to commit such configurations.

For information about the BGP protocol, see the *Junos OS Routing Protocols Configuration Guide*.

Related Documentation

- Example: Configuring the BGP and IS-IS Routing Protocols

CHAPTER 8

Configuring Keys for SSH and SSL

Configuring SSH Host Keys for Secure Copying of Data

Secure Shell (SSH) uses encryption algorithms to generate a host, server, and session key system that ensures secure data transfer. You can configure SSH host keys to support secure copy (SCP) as an alternative to FTP for the background transfer of data such as configuration archives and event logs. To configure SSH support for SCP, you must complete the following tasks:

- Specify SSH known hosts by including hostnames and host key information in the Routing Engine configuration hierarchy.
- Set an SCP URL to specify the host from which to receive data. Setting this attribute automatically retrieves SSH host key information from the SCP server.
- Verify that the host key is authentic.
- Accept the secure connection. Accepting this connection automatically stores host key information in the local host key database. Storing host key information in the configuration hierarchy automates the secure handshake and allows background data transfer using SCP.

Tasks to configure SSH host keys for secure copying of data are:

1. Configuring SSH Known Hosts on page 69
2. Configuring Support for SCP File Transfer on page 70
3. Updating SSH Host Key Information on page 70

Configuring SSH Known Hosts

To configure SSH known hosts, include the **host** statement, and specify hostname and host key options for trusted servers at the **[edit security ssh-known-hosts]** hierarchy level:

```
[edit security ssh-known-hosts]
host corporate-archive-server, ip-address {
    dsa-key key;
}
host archive-server-url {
```

```
    rsa-key key;
}
host server-with-ssh-version-1, ip-address {
    rsa1-key key;
}
```

Host keys are one of the following:

- **dsa-key**—Base64 encoded Digital Signature Algorithm (DSA) key.
- **rsa-key**—Base64 encoded RSA public key algorithm, which supports encryption and digital signatures.
- **rsa1-key**—Base64 encoded RSA public key algorithm, which supports encryption and digital signatures for SSH version 1 and SSH version 2.

Configuring Support for SCP File Transfer

To configure a known host to support background SCP file transfers, include the **archive-sites** statement at the **[edit system archival configuration]** hierarchy level.

```
[edit system archival configuration]
archive-sites {
    scp://username<:password>@host<:port>/url-path;
}
```

NOTE: When specifying a URL in a Junos OS statement using an IPv6 host address, you must enclose the entire URL in quotation marks (" ") and enclose the IPv6 host address in brackets ([ ]). For example, **"scp://username<:password>@[host]<:port>/url-path";**

Setting the **archive-sites** statement to point to an SCP URL triggers automatic host key retrieval. At this point, Junos OS connects to the SCP host to fetch the SSH public key, displays the host key message digest or fingerprint as output to the console, and terminates the connection to the server.

```
user@switch# set system archival configuration archive-sites "<scp-url-path>"
The authenticity of host <my-archive-server (<server-ip-address>)> can't be established.
    RSA key fingerprint is <ascii-text key>. Are you sure you want to continue connecting
    (yes/no)?
```

To verify that the host key is authentic, compare this fingerprint with a fingerprint that you obtain from the same host using a trusted source. If the fingerprints are identical, accept the host key by entering **yes** at the prompt. The host key information is then stored in the Routing Engine configuration and supports background data transfers using SCP.

Updating SSH Host Key Information

Typically, SSH host key information is automatically retrieved when you set a URL attribute for SCP using the **archival configuration archive-sites** statement at the **[edit system]**

hierarchy level. However, if you need to manually update the host key database, use one of the following methods.

1. Retrieving Host Key Information Manually on page 71
2. Importing Host Key Information from a File on page 71

Retrieving Host Key Information Manually

To manually retrieve SSH public host key information, use the **fetch-from-server** option with the **set security ssh-known-hosts** command. You must include a hostname attribute with the **set security ssh-known-hosts fetch-from-server** command to specify the host from which to retrieve the SSH public key.

```
user@switch# set security ssh-known-hosts fetch-from-server <hostname>
```

Importing Host Key Information from a File

To manually import SSH host key information from the known-hosts file located at **/var/tmp/known-hosts** on the server, include the **load-key-file** option with the **set security ssh-known-hosts** command. You must include the path to the **known-hosts** file with the **set security ssh-known-hosts load-key-file** command to specify the location from which to import host key information.

```
user@switch# set security ssh-known-hosts load-key-file /var/tmp/known-hosts
```

Importing SSL Certificates for Junos XML Protocol Support

A Junos XML protocol client application can use one of four protocols to connect to the Junos XML protocol server on a router or switch: clear-text (a Junos XML protocol-specific protocol for sending unencrypted text over a TCP connection), SSH, SSL, or Telnet. For clients to use the SSL protocol, you must copy an X.509 authentication certificate onto the router or switch, as described in this topic. You must also include the **xnm-ssl** statement at the **[edit system services]** hierarchy level.

NOTE: The xnm-ssl statement does not apply to standard IPsec services.

After obtaining an X.509 authentication certificate and private key, copy it to the router or switch by including the **local** statement at the **[edit security certificates]** hierarchy level:

```
[edit security certificates]
local certificate-name {
   load-key-file (filename | url);
}
```

certificate-name is a name you choose to identify the certificate uniquely (for example, **Junos XML protocol-ssl-client-*hostname***, where ***hostname*** is the computer where the client application runs).

filename is the pathname of the file on the local disk that contains the paired certificate and private key (assuming you have already used another method to copy them to the router's or switch's local disk).

url is the URL to the file that contains a paired certificate and private key (for instance, on the computer where the Junos XML protocol client application runs).

NOTE: The CLI expects the private key in the *URL-or-path* file to be unencrypted. If the key is encrypted, the CLI prompts you for the passphrase associated with it, decrypts it, and stores the unencrypted version.

The load-key-file statement acts as a directive that copies the contents of the certificate file into the configuration. When you view the configuration, the CLI displays the string of characters that constitute the private key and certificate, marking them as SECRET-DATA. The load-key-file keyword is not recorded in the configuration.

Related Documentation

- Configuring SSH Host Keys for Secure Copying of Data on page 69
- Configuring clear-text or SSL Service for Junos XML Protocol Client Applications

CHAPTER 9

Configuration Statements

Security Services Configuration Statements

To configure security services, you can include the following configuration statements at the **[edit security]** hierarchy level:

```
[edit security]
authentication-key-chains {
    key-chain key-chain-name {
        key key {
            secret secret-data;
            start-time yyyy-mm-dd.hh:mm:ss;
        }
    }
}
certificates {
    cache-size bytes;
    cache-timeout-negative seconds;
    certification-authority ca-profile-name {
        ca-name ca-identity;
        crl file-name;
        encoding (binary | pem);
        enrollment-url url-name;
        file certificate-filename;
        ldap-url url-name;
    }
    enrollment-retry attempts;
    local certificate-filename {
        certificate-key-string;
        load-key-file URL key-filename;
    }
    maximum-certificates number;
    path-length certificate-path-length;
}
ike {
    proposal ike-proposal-name {
        authentication-algorithm (md5 | sha1);
        authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
        description description;
```

```
      dh-group (group1 | group2);
      encryption-algorithm (3des-cbc | des-cbc | ase-128-cbc | ase-192-cbc | ase-256-cbc);
      lifetime-seconds seconds;
    }
    policy ike-peer-address {
      description description;
      encoding (binary | pem);
      identity identity-name;
      local-certificate certificate-filename;
      local-key-pair private-public-key-file;
      mode (aggressive | main);
      pre-shared-key (ascii-text key | hexadecimal key);
      proposals [ proposal-names ];
    }
  }
  ipsec {
    security-association {
      manual {
        direction (bidirectional | inbound | outbound) {
          protocol esp;
          spi spi-value;
          encryption {
            algorithm 3des-cbc;
            key ascii-text ascii-text-string;
          }
        }
      }
    }
    proposal ipsec-proposal-name {
      authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
      description description;
      encryption-algorithm (3des-cbc | des-cbc);
      lifetime-seconds seconds;
      protocol (ah | esp | bundle);
    }
    policy ipsec-policy-name {
      description description;
      perfect-forward-secrecy {
        keys (group1 | group2);
      }
      proposals [ proposal-names ];
    }
    security-association sa-name {
      description description;
      dynamic {
        ipsec-policy policy-name;
        replay-window-size (32 | 64);
      }
      manual {
        direction (inbound | outbound | bidirectional) {
          authentication {
            algorithm (hmac-md5-96 | hmac-sha1-96);
            key (ascii-text key | hexadecimal key);
          }
          auxiliary-spi auxiliary-spi;
          encryption {
```

```
            algorithm (des-cbc | 3des-cbc);
            key (ascii-text key | hexadecimal key);
          }
          protocol (ah | esp | bundle);
          spi spi-value;
        }
      }
      mode (tunnel | transport);
    }
  }
  pki {
    auto-re-enrollment {
      certificate-id {
        ca-profile ca-profile-name;
        challenge-password password;
        re-enroll-trigger-time-percentage percentage;
        re-generate-keypair;
        validity-period days;
      }
    }
    ca-profile ca-profile-name {
      ca-identity ca-identity;
      enrollment {
        url url-name;
        retry number-of-attempts;
        retry-interval seconds;
      }
      revocation-check {
        disable;
        crl {
          disable on-download-failure;
          refresh-interval number-of-hours;
          url {
            url-name;
            password;
          }
        }
      }
    }
    traceoptions {
      file filename <files number> <match regular-expression> <size maximum-file-size>
        <world-readable | no-world-readable>;
      flag flag;
    }
  }
  ssh-known-hosts {
    host {
      dsa-key key;
      rsa-key key;
      rsa1-key key;
    }
  }
  traceoptions {
    file filename <files number> < size size>;
    flag all;
    flag database;
```

```
    flag general;
    flag ike;
    flag parse;
    flag policy-manager;
    flag routing-socket;
    flag timer;
}
```

NOTE: Most of the configuration statements do not have default values. If you do not specify an identifier for a statement that does not have a default value, you cannot commit the configuration.

For information about IP Security (IPsec) monitoring and troubleshooting, see the *Junos OS System Basics and Services Command Reference*.

Related Documentation

- Security Services Configuration Statements on page 76

Security Services Configuration Statements

The following table lists the security services configuration statements available at the **[edit security]** hierarchy level:

Table 3: Security Services Configuration Statements

A-C	D-G	H-M	N-R	S-Z
algorithm (Authentication Keychain)	description (Authentication Keychain)	identity	options	secret
algorithm (Junos FIPS)	description (IKE policy)	ike	path-length	security-association (Junos OS)
authentication	dh-group	internal	perfect-forward-secrecy	security-association (Junos-FIPS Software)
authentication-algorithm (IKE)	direction (Junos OS)	ipsec	pki	spi (Junos OS)
authentication-algorithm (IPsec)	direction (Junos-FIPS Software)	key (Authentication Keychain)	policy (IKE)	spi (Junos-FIPS Software)
authentication-key-chains	dynamic	key (Junos FIPS)	policy (IPsec)	ssh-known-hosts
authentication-method	encoding	key-chain	pre-shared-key	start-time
auto-re-enrollment	encryption (Junos OS)	ldap-url	proposal (IKE)	tolerance

Table 3: Security Services Configuration Statements *(continued)*

A-C	D-G	H-M	N-R	S-Z
auxiliary-spi	encryption (Junos-FIPS Software)	lifetime-seconds	proposal (IPsec)	traceoptions
ca-identity	encryption-algorithm	local	proposals	url
ca-name	enrollment	local-certificate	protocol (Junos OS)	validity-period
ca-profile	enrollment-retry	local-key-pair	protocol (Junos-FIPS Software)	
cache-size	enrollment-url	manual (Junos OS)	re-enroll-trigger-time-percentage	
cache-timeout-negative	file	manual (Junos-FIPS Software)	re-generate-keypair	
certificate-id		maximum-certificates	refresh-interval	
certificates		mode (IKE)	retry	
certification-authority		mode (IPsec)	retry-interval	
challenge-password			revocation-check	
crl (Adaptive Services Interface)				
crl (Encryption Interface)				

Related Documentation

- Security Services Configuration Statements on page 73

PART 3

Administration

CHAPTER 10

IPsec Administrative Commands

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

List of Sample Output request security pki ca-certificate enroll on page 82

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

List of Sample Output request security pki ca-certificate load on page 83

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 85

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

List of Sample Output

request security pki generate-certificate-request on page 87

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+1BPNbCxNWYvyrnSyVYDbFj8oOXyqog8ACDfVL2JBWrPNBYy7imq/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 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

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 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

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 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

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
```

CHAPTER 11

IPsec Monitoring 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 82
- request security pki ca-certificate load on page 83
- show security pki ca-certificate

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

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 89
- show security pki local-certificate

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 89
- show security pki local-certificate

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.

certificate-cache-entry *number*—(Optional) Delete digital certificates matching a specified cache entry number. To view the certificate cache entry numbers, issue the **show services ipsec-vpn certificates** command.

Required Privilege Level clear

Related Documentation

- show services ipsec-vpn certificates

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 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

List of Sample Output
clear services ipsec-vpn ipsec statistics on page 100

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 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

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 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

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
```

PART 4

Index

Index

Symbols

A

B

C

D

R

S

T