Implementing Internet Key Exchange Security Protocol on Cisco ASR 9000 Series Routers

Internet Key Exchange (IKE) is a key management protocol standard that is used in conjunction with the IP Security (IPSec) standard. IPSec is a feature that provides robust authentication and encryption of IP packets.

IKE is a hybrid protocol that implements the Oakley key exchange and the Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

This module describes how to implement IKE on Cisco ASR 9000 Series Aggregation Services Routers.

For a complete description of the IKE commands used in this module, see the Internet Key Exchange Security Protocol Commands on Cisco ASR 9000 Series Routers module of the Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference. To locate documentation of other commands that appear in this module, use the command reference master index, or search online.

Feature History for Implementing Internet Key Exchange Security Protocol on Cisco ASR 9000 Series Routers

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced on the Cisco ASR 9000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites, page SC-20
- Information About Implementing IKE Security Protocol Configurations for IPSec Networks, page SC-20
- IPSec Dead Peer Detection Periodic Message Option, page SC-31
- How to Implement IKE Security Protocol Configurations for IPSec Networks, page SC-31
- How to Configure the ISAKMP Profile for a Service Interface, page SC-49
- How to Configure a Periodic Dead Peer Detection Message, page SC-52
Prerequisites

The following prerequisites are required to implement IKE:

- You must be in a user group associated with a task group that includes the proper task IDs for security commands. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must install and activate the package installation envelope (PIE) for the security software. For detailed information about optional PIE installation, see the Cisco ASR 9000 Series Aggregation Services Router System Management Guide.

Information About Implementing IKE Security Protocol Configurations for IPSec Networks

To implement IKE, you should understand the following concepts:

- Supported Standards, page SC-20
- Concessions for Not Enabling IKE, page SC-22
- IKE Policies, page SC-22
- ISAKMP Identity, page SC-26
- ISAKMP Profile Overview, page SC-26
- Mask Preshared Keys, page SC-27
- Preshared Keys Using an AAA-Method Server, page SC-28
- Internet Key Exchange Mode Configuration, page SC-28
- Internet Key Exchange Extended Authentication, page SC-29
- Call Admission Control, page SC-29
- Information About IP Security VPN Monitoring, page SC-30

Supported Standards

Cisco implements the following standards:

- **IKE**—Internet Key Exchange. A hybrid protocol that implements Oakley and Skeme key exchanges inside the ISAKMP framework. IKE can be used with other protocols, but its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations (SAs).
  IKE is implemented following RFC 2409, *The Internet Key Exchange*.

- **IPSec**—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms
based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec is used to protect one or more data flows between a pair of hosts, a pair of security gateways, or a security gateway and a host.

For more information on IPSec, see the Implementing IPSec Network Security on Cisco ASR 9000 Series Routers module of the Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide.

- **ISAKMP**—Internet Security Association and Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

  ISAKMP is implemented following the latest version of the Internet Security Association and Key Management Protocol (ISAKMP) Internet Draft (RFC 2408).

- **Oakley**—A key exchange protocol that defines how to derive authenticated keying material.

- **Skeme**—A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.

The component technologies implemented for use by IKE include the following:

- **DES**—Data Encryption Standard. An algorithm that is used to encrypt packet data. IKE implements the 56-bit DES-CBC with Explicit IV standard. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet.

  Cisco IOS XR software also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be sent over untrusted networks. It enables customers, particularly in the finance industry, to use network-layer encryption.

- **AES**—Advanced Encryption Standard. Standards of 128-bit, 192-bit, and 256-bit are supported.

  **Note** Cisco IOS XR images that have strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to U.S. government export controls, and have a limited distribution. Images that are to be installed outside the United States require an export license. Customer orders might be denied or subject to delay because of U.S. government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- **Diffie-Hellman**—A public-key cryptography protocol that allows two parties to establish a shared secret over an insecure communications channel. Diffie-Hellman is used within IKE to establish session keys. 768-bit, 1024-bit, and 1536-bit Diffie-Hellman groups are supported.

- **MD5 (HMAC variant)**—Message Digest 5. A hash algorithm used to authenticate packet data. HMAC is a variant that provides an additional level of hashing.

- **SHA (HMAC variant)**—Secure Hash Algorithm. A hash algorithm used to authenticate packet data. HMAC is a variant that provides an additional level of hashing.

- **RSA signatures and RSA encrypted nonces**—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adelman. RSA signatures provide nonrepudiation, and RSA encrypted nonces provide repudiation. (Repudiation and nonrepudiation are associated with traceability.)
IKE interoperates with the X.509v3 certificates standard. It is used with the IKE protocol when authentication requires public keys. This certificate support allows the protected network to scale by providing the equivalent of a digital ID card to each device. When two devices want to communicate, they exchange digital certificates to prove their identity; thus, removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer.

**Concessions for Not Enabling IKE**

IKE is disabled by default in Cisco IOS XR software. If you do not enable IKE, you must make these concessions at the peers:

- You must manually specify all IPSec security associations in the crypto profiles at all peers. (Crypto profile configuration is described in the module Implementing IPSec Network Security on Cisco ASR 9000 Series Routers.)
- The IPSec security associations of the peers never time out for a given IPSec session.
- During IPSec sessions between the peers, the encryption keys never change.
- Anti-replay services are not available between the peers.
- Certification authority (CA) support cannot be used.

**IKE Policies**

You must create IKE policies at each peer. An IKE policy defines a combination of security parameters to be used during the IKE negotiation.

Before you create and configure IKE policies you should understand the following concepts:

- IKE Policy Creation, page SC-22
- Definition of Policy Parameters, page SC-23
- IKE Peer Agreement for Matching Policies, page SC-23
- Limitation of an IKE Peer to a Specific Set of Policies, page SC-24
- Value Selection for Parameters, page SC-24
- Policy Creation, page SC-25
- Additional Configuration Required for IKE Policies, page SC-25

**IKE Policy Creation**

IKE negotiations must be protected, so each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree on a policy, the security parameters of the policy are identified by a security association established at each peer, and these security associations apply to all subsequent IKE traffic during the negotiation.

You can create multiple, prioritized policies at each peer to ensure that at least one policy matches the policy of a remote peer.
Definition of Policy Parameters

Table 1 lists the five parameters to define in each IKE policy.

Table 1  IKE Policy Parameter Definitions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Accepted Values</th>
<th>Keyword</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption algorithm</td>
<td>56-bit DES-CBC</td>
<td>des</td>
<td>56-bit DES-CBC</td>
</tr>
<tr>
<td></td>
<td>168-bit DES</td>
<td>3des</td>
<td></td>
</tr>
<tr>
<td></td>
<td>128-bit AES</td>
<td>aes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>192-bit AES</td>
<td>aes 192</td>
<td></td>
</tr>
<tr>
<td></td>
<td>256-bit AES</td>
<td>aes 256</td>
<td></td>
</tr>
<tr>
<td>Hash algorithm</td>
<td>SHA-1 (HMAC variant)</td>
<td>sha</td>
<td>SHA-1</td>
</tr>
<tr>
<td></td>
<td>MD5 (HMAC variant)</td>
<td>md5</td>
<td></td>
</tr>
<tr>
<td>Authentication method</td>
<td>RSA signatures</td>
<td>rsa-sig</td>
<td>RSA signatures</td>
</tr>
<tr>
<td></td>
<td>RSA encrypted nonces</td>
<td>rsa-encr</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Preshared keys</td>
<td>pre-share</td>
<td></td>
</tr>
<tr>
<td>Diffie-Hellman group identifier</td>
<td>768-bit Diffie-Hellman</td>
<td>1</td>
<td>768-bit Diffie-Hellman</td>
</tr>
<tr>
<td></td>
<td>1024-bit Diffie-Hellman</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1536-bit Diffie-Hellman</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Lifetime of the security association¹</td>
<td>Any number of seconds</td>
<td>—</td>
<td>86400 seconds (1 day)</td>
</tr>
</tbody>
</table>

¹ For information about this lifetime and how it is used, see the command description for the lifetime command.

These parameters apply to the IKE negotiations when the IKE security association is established.

IKE Peer Agreement for Matching Policies

When the IKE negotiation begins, 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 will try to find a match. The remote peer looks for a match by comparing its own highest priority policy (designated by the lowest priority number) against the policies received from the other peer. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer policy specifies a lifetime that is less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime—from the remote peer’s policy—is used.)

If no acceptable match is found, IKE refuses negotiation and IPSec is not established. (See related information in the “Limitation of an IKE Peer to a Specific Set of Policies” section.)

If a match is found, IKE completes negotiation, and an ISAKMP security association (SA) is created. To establish an ISAKMP SA pre-shared key or certificate, a match must be configured. Without a match, no ISAKMP SA can be established.
Note

Depending on which authentication method is specified in a policy, additional configuration might be required (as described in the “Additional Configuration Required for IKE Policies” section on page SC-25). If a peer’s policy does not have the required companion configuration, the peer does not submit the policy when attempting to find a matching policy with the remote peer.

Limitation of an IKE Peer to a Specific Set of Policies

Cisco VPN clients are preconfigured with all available policies, and propose all of these policies when connecting to the hub. The hub must then select the “first-match” policy. However, some users may have a need to restrict the use of strong encryption algorithms between the local and remote peer when they connect through the IPSec gateway. Because the Cisco VPN client does not allow users to choose which policy (and therefore which encryption algorithm) to use, these users may instead configure policy sets that in effect create such restrictions. Matches between peer and policy set are then restricted or allowed, based on a match with the local IP address (or tunnel source configured at the SVI) identified in the policy set.

For example, an IPSec hub is configured with six policies, but the policy set is configured with only three of these six. When a remote client tries to initiate a tunnel and refers to this SVI tunnel source address, the policy set is matched. IKE looks for a match among the three policies dictated by the policy set, starting from the highest to the lowest priority number (the lower the number, the higher the priority). If no match exists among these three policies, no tunnel can be established.

If a remote peer tries to connect to an SVI, whose local IP address does not restrict it to certain IKE policies, then the default behavior described under “IKE Peer Agreement for Matching Policies” is operational.

You may configure up to five ISAKMP policies within a single policy set.

For information about how to limit an IKE peer to a specific set of policies, see Manually Configuring RSA Keys, page SC-35 of this module.

Value Selection for Parameters

You can select certain values for each parameter, following the IKE standard. But why choose one value over another?

If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the value supported by the other device. Aside from this, a trade-off between security and performance often exists, and many of these parameter values represent such a trade-off. You should evaluate the level of security risks for your network and your tolerance for these risks. Then the following tips might help you select which value to specify for each parameter:

- The encryption algorithm has five options: 56-bit DES-CBC, 168-bit DES, 128-bit AES, 192-bit AES, and 256-bit AES.
- The hash algorithm has two options: SHA-1 and MD5.
  
  MD5 has a smaller digest and is considered to be slightly faster than SHA-1. A demonstrated successful (but extremely difficult) attack has been demonstrated against MD5; however, the HMAC variant used by IKE prevents this attack.
- The authentication method has three options: RSA signatures, RSA encrypted nonces, and preshared keys.
  
  RSA signatures provide nonrepudiation for the IKE negotiation (you can prove to a third party after the fact that you did indeed have an IKE negotiation with the remote peer).
RSA signatures allow the use of a CA. Using a CA can dramatically improve the manageability and scalability of your IPSec network. Additionally, RSA signature-based authentication uses only two public key operations, whereas RAS encryption uses four public key operations, making it costlier in terms of overall performance.

- RSA encrypted nonces provide repudiation for the IKE negotiation (you cannot prove to a third party that you had an IKE negotiation with the remote peer).

  RSA encrypted nonces require that peers possess each other’s public keys but do not use a certification authority. Instead, two ways exist for peers to get each other’s public keys:

  If your local peer has previously used RSA signatures with certificates during a successful IKE negotiation with a remote peer, your local peer already possesses the remote peer’s public key. (The peers’ public keys are exchanged during the RSA-signatures-based IKE negotiations, if certificates are used.)

- Preshared keys are clumsy to use if your secured network is large, and they do not scale well with a growing network. However, they do not require use of a certification authority, as do RSA signatures, and might be easier to set up in a small network with fewer than ten nodes. RSA signatures also can be considered more secure when compared with preshared key authentication.

- The Diffie-Hellman group identifier has three options: 768-bit, 1024-bit Diffie-Hellman, and 1536-bit Diffie Hellman.

  The 1024-bit Diffie-Hellman and 1536-bit Diffie Hellman options are harder to crack but require more CPU time to execute.

- The lifetime of the security association can be set to any value.

  As a general rule, the shorter the lifetime (up to a point), the more secure your IKE negotiations. However, with longer lifetimes, future IPSec security associations can be set up more quickly. For more information about this parameter and how it is used, see the command description for the lifetime command.

**Policy Creation**

You can create multiple IKE policies, each with a different combination of parameter values. For each policy that you create, assign a unique priority (1 through 10,000, with 1 being the highest priority).

You can configure multiple policies on each peer—but at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. (The lifetime parameter need not necessarily be the same; see details in the “IKE Peer Agreement for Matching Policies” section on page SC-23.)

If you do not configure any policies, your router uses the default policy, which is always set to the lowest priority and contains the default value of each parameter.

**Additional Configuration Required for IKE Policies**

Depends on the authentication method you specify in your IKE policies, you must perform certain additional configuration tasks before IKE and IPSec can successfully use the IKE policies.

Each authentication method requires additional companion configuration as follows:

- RSA signatures method. If you specify RSA signatures as the authentication method in a policy, you may configure the peers to obtain certificates from a CA. (The CA must be properly configured to issue the certificates.) Configure this certificate support as described in the module “Implementing Certification Authority Interoperability.”
The certificates are used by each peer to exchange public keys securely. (RSA signatures require that each peer has the public signature key of the remote peer.) When both peers have valid certificates, they automatically exchange public keys with each other as part of any IKE negotiation in which RSA signatures are used.

- RSA encrypted nonces method. If you specify RSA encrypted nonces as the authentication method in a policy, you must ensure that each peer has the public keys of the other peers.

Unlike RSA signatures, the RSA encrypted nonces method cannot use certificates to exchange public keys. Instead, you ensure that each peer has the others’ public keys by one of the following methods:

- Ensuring that an IKE exchange using RSA signatures with certificates has already occurred between the peers. (The peers’ public keys are exchanged during the RSA-signatures-based IKE negotiations if certificates are used.)

To make this happen, specify two policies: a higher-priority policy with RSA encrypted nonces and a lower-priority policy with RSA signatures. When IKE negotiations occur, RSA signatures are used the first time because the peers do not yet have each other’s public keys. Then future IKE negotiations are able to use RSA encrypted nonces because the public keys will have been exchanged.

This alternative requires that you have certification authority support configured.

- Preshared keys authentication method. If you specify preshared keys as the authentication method in a policy, you must configure these preshared keys as described in the “Configuring ISAKMP Preshared Keys in ISAKMP Keyrings” section on page SC-39.

If RSA encryption is configured and signature mode is negotiated (and certificates are used for signature mode), the peer requests both signature and encryption keys. Basically, the router requests as many keys as the configuration supports. If RSA encryption is not configured, it just requests a signature key.

**ISAKMP Identity**

You should set the ISAKMP identity for each peer that uses preshared keys in an IKE policy.

When two peers use IKE to establish IPSec security associations, each peer sends its identity to the remote peer. Each peer sends either its hostname or its IP address, depending on how you have set the ISAKMP identity of the router.

By default, the ISAKMP identity of a peer is the IP address of the peer. If appropriate, you could change the identity to be the peer’s hostname instead. As a general rule, set the identities of all peers the same way—either all peers should use their IP addresses or all peers should use their host names. If some peers use their host names and some peers use their IP addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a domain name server (DNS) lookup is unable to resolve the identity.

**ISAKMP Profile Overview**

The ISAKMP profile is an enhancement to Internet Security Association and Key Management Protocol (ISAKMP) configurations. It enables modularity of ISAKMP configuration for Phase-1 negotiations. This modularity allows mapping different ISAKMP parameters to different IP Security (IPSec) tunnels, and mapping different IPSec tunnels to different VRF forwarding and routing instances.

Currently, many applications and enhancements use the ISAKMP profile, including quality of service (QoS), router certificate management, and Multiprotocol Label Switching (MPLS) VPN configurations.
An ISAKMP profile is a repository for IKE Phase-1 and IKE Phase-1.5 configuration for a set of peers. An ISAKMP profile applies parameters to an incoming IPSec connection identified uniquely through its concept of match identity criteria. These criteria are based on the IKE identity that is presented by incoming IKE connections and includes IP address, fully qualified domain name (FQDN), and group (the Virtual Private Network [VPN] remote client grouping). The granularity of the match identity criteria imposes the granularity of applying the specified parameters. The ISAKMP profile applies parameters specific to each profile, such as trust points, peer identities, and XAUTH authentication, authorization, and accounting (AAA) list, and so forth. For information about XAUTH, see Internet Key Exchange Mode Configuration, page SC-28.

**ISAKMP Profile Considerations**

Consider the following guidelines on when to use the ISAKMP profile:

- You have a router with two or more IPSec connections that require differing Phase-1 parameters for different sites (for example, when you want to configure site-to-site and remote access on the same router).

- When different custom Internet Key Exchange (IKE) Phase-1 policies may be needed for different peers. One determining factor might be whether you are applying XAUTH to a specific peer, rather than applying it to every connection.

**Mask Preshared Keys**

A mask preshared key lets a group of remote users with the same level of authentication share an IKE preshared key. The preshared key of the remote peer must match the preshared key of the local peer for IKE authentication to occur.

A mask preshared key is usually distributed through a secure out-of-band channel. In a remote peer-to-local peer scenario, any remote peer with the IKE preshared key configured can establish IKE SAs with the local peer.

If you specify a `subnet-address` value with the `crypto keyring` command, it is up to you to use a subnet address, which allows more peers to share the same key. That is, the preshared key is no longer restricted to use between two users.

> **Note**
>
> We do not recommend using 0.0.0.0 as a subnet address, because it encourages group preshared keys, which allow all peers to have the same group key, thereby reducing the security of your user authentication.

Mask preshared keys have the following restrictions:

- A security association (SA) cannot be established between the IPSec peers until all IPSec peers are configured for the same preshared key. (An SA is a description of how two or more entities use security services to communicate securely on behalf of a particular data flow.)

- The mask preshared key must be distinctly different for remote users requiring varying levels of authorization. You must configure a new preshared key for each level of trust and assign keys individually, as appropriate, to each party. Otherwise, an untrusted party may obtain access to protected data.
Preshared Keys Using an AAA-Method Server

Preshared keys do not scale well in a large Virtual Private Network (VPN) unless you use a certification authority (CA). When dynamic IP addressing such as DHCP or PPP dialup is used, the changing IP address can make key lookup difficult or impossible unless you use a mask preshared key. On the other hand, mask preshared keys are not very secure, because then large number of users receive the same secret, thereby reducing security.

Configuring preshared keys using an authentication, authorization, and accounting (AAA) server allows individual users to have their own key, which is stored on an external AAA server. This makes it possible to centrally manage the user database and to link it to an existing AAA database. It also gives each user a unique, more secure preshared key.

To configure this feature, you must perform the following tasks at each peer:

- Configure a dynamic crypto ISAKMP profile. See Manually Configuring RSA Keys.

Preshared keys using an AAA server have the following restrictions:

- The shared secret can be accessed only in aggressive mode. The ID of the IKE exchange is used as the username to query AAA if no local key can be found on the Cisco IOS XR router to which the user is trying to connect. Aggressive mode provides the ID in the first part of the IKE exchange; main mode does not provide the ID until the latter part of the IKE exchange, which is too late for key lookup.
- Only the following ID types can be used:
  - IPv4 address (can be different from the one assigned by the Internet service provider [ISP])
  - FQDN (fully qualified domain name)
  - E-mail address

Internet Key Exchange Mode Configuration

IKE mode configuration, as defined by the Internet Engineering Task Force (IETF), allows a gateway to download an IP address (and other network level configuration) to the client as part of an IKE negotiation. Using this exchange, the gateway gives IP addresses to the IKE client to be used as an “inner” IP address encapsulated under IPSec. This method provides a known IP address for the client that can be matched against IPSec policy.

The client initiation type of IKE mode configuration is supported. The client initiates the configuration mode with the gateway. The gateway responds with an IP address that it has allocated for the client.

Mode configuration must be applied to a crypto ISAKMP profile to be enforced. For instructions on how to apply mode configuration to a crypto ISAKMP profile, see the “Manually Configuring RSA Keys” section on page SC-35.

Interfaces with crypto ISAKMP profiles, which are configured for IKE mode configuration, may experience a slightly longer connection setup time. This longer setup time is true even for IKE peers that refuse to be configured or do not respond to the configuration mode request. In both cases, the gateway initiates the configuration of the client.
Internet Key Exchange Extended Authentication

IKE extended authentication (Xauth) is a draft RFC based on the IKE protocol. Xauth allows all Cisco IOS XR software AAA authentication methods to perform user authentication in a separate phase after the IKE authentication Phase-1 exchange. The AAA configuration list name must match the Xauth configuration list name for user authentication to occur.

Xauth does not replace IKE. IKE allows for device authentication and Xauth allows for user authentication, which occurs after IKE device authentication. Xauth occurs after IKE authentication Phase 1, but before IKE IPSec SA negotiation Phase 2.

To configure Xauth, perform the following tasks:

- Configure AAA (you must set up an authentication list). See the Configuring AAA Services on Cisco ASR 9000 Series Routers module of the Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide.
- Configure a static crypto ISAKMP profile. For configuration details, see the “How to Configure the ISAKMP Profile for a Service Interface” section on page SC-49.
- Configure a dynamic crypto ISAKMP profile (optional). For configuration details, see “How to Configure the ISAKMP Profile for a Service Interface” section on page SC-49.
- Configure ISAKMP policy. For configuration details, see the “Configuring IKE Policies” section on page SC-33.

Call Admission Control

The Call Admission Control (CAC) for Internet Key Exchange (IKE) feature describes the application of CAC to the IKE protocol in Cisco IOS XR software. CAC limits the number of simultaneous IKE security associations (SAs) (that is, calls to CAC) that a router can establish. In addition, there is an option to limit the maximum number of active IKE SAs allowed in the system and the CPU usage that is consumed by the IKE process or global CPU. The main function of CAC is to protect the router from severe resource depletion and to prevent crashes.

IKE Session

You can configure the absolute IKE SA limit by using the crypto isakmp call admission limit command. The router drops new IKE SA requests when the value has been reached.

Security Association Limit

IKE uses SAs to identify the parameters of its connections. IKE can negotiate and establish its own SA. An IKE SA, which is bidirectional, is used only by IKE. An IKE SA cannot limit IPSec.

IKE drops SA requests based on a user-configured SA limit. To configure an IKE SA limit, use the crypto isakmp call admission limit command. When there is a new SA request from a peer router, IKE determines if the number of active IKE SAs being negotiated meets or exceeds the configured SA limit. If the number is greater than or equal to the limit, the new SA request is rejected and a system log is generated. This log contains the source destination IP address of the SA request.
Information About IP Security VPN Monitoring

The IP Security (IPSec) VPN Monitoring feature provides VPN session monitoring enhancements that allow you to troubleshoot the Virtual Private Network (VPN) and monitor the end-user interface. Session monitoring includes the following enhancements:

- Ability to specify an Internet Key Exchange (IKE) peer description in the configuration file.
- Summary listing of crypto session status.
- Ability to clear both IKE and IP Security (IPSec) security associations (SAs) using one command-line interface (CLI).
- Ability to expand the filtering mechanism by using the options from the show crypto session command.

To implement IPSec VPN monitoring, you must understand the following concepts:

- Crypto Sessions Background, page SC-30
- Per-IKE Peer Description, page SC-30
- Summary Listing of Crypto Session Status, page SC-30
- IKE and IPSec Security Exchange Clear Command, page SC-31

Crypto Sessions Background

A crypto session is a set of IPSec connections (flows) between two crypto endpoints. If the two crypto endpoints use IKE as the keying protocol, they are IKE peers to each other. Typically, a crypto session consists of one IKE security association (for control traffic) and at least two IPSec security associations (for data traffic—one per each direction). There may be duplicated IKE security associations (SAs) and IPSec SAs or duplicated IKE SAs or IPSec SAs for the same session during rekeying or because of simultaneous setup requests from both sides.

Per-IKE Peer Description

The Per-IKE Peer Description function allows you to enter a description of your choice for an IKE peer. The unique peer description, which includes up to 80 characters, is used whenever you are referencing that particular IKE peer. To add the peer description, use the description (ISAKMP peer) command.

The primary application of this description field is for monitoring purposes (for example, when using show commands or for logging [syslog messages]). The description field is purely informational.

Summary Listing of Crypto Session Status

You can obtain a list of status information for active crypto sessions by using the show crypto session command. The listing includes the following summary status of the crypto session:

- Interface
- IKE SAs that are associated with the peer by whom the IPSec SAs are created
- IPSec SAs serving the flows of a session

Up to two IKE SAs and multiple IPSec SAs can be established for the same peer (for the same session), in which case IKE peer descriptions are repeated with different values for the IKE SAs that are associated with the peer and for the IPSec SAs that are serving the flows of the session.
In addition, you can use the `show crypto session` command with the `detail` keyword to obtain more detailed information about the sessions.

### IKE and IPSec Security Exchange Clear Command

The `clear crypto session` command allows you to clear both IKE and IPSec. To clear a specific crypto session or a subset of all the sessions (for example, a single tunnel to one remote site), you need to provide session-specific parameters, such as a local or remote IP address, a local or remote port, a front door VPN routing and forwarding (FVRF) name, or an inside VRF (IVRF) name. Typically, the remote IP address is used to specify a single tunnel to be deleted.

If a local IP address is provided as a parameter when you use the `clear crypto session` command, all the sessions (and their IKE SAs and IPSec SAs) that share the IP address as a local crypto endpoint (IKE local address) are cleared. If you do not provide a parameter, all IPSec SAs and IKE SAs that are in the router are deleted.

### IPSec Dead Peer Detection Periodic Message Option

A peer is an IPSec-compliant node capable of establishing IKE channels and negotiating SAs between itself and other peers. Peers can lose their IP connection to other peers due to routing problems, peer reloading, or other situations, resulting in a loss of packet traffic (sometimes called a “black hole”).

The IPSec Dead Peer Detection (DPD) Periodic Message Option feature (defined in RFC 3706) allows you to query the liveliness of the Internet Key Exchange (IKE) peer on your router at regular intervals. The benefit of this configuration approach over that of the default configuration (on-demand dead peer detection) is the earlier detection of dead peers.

### How to Implement IKE Security Protocol Configurations for IPSec Networks

To configure the IKE security protocol for IPSec networks, perform the tasks described in the following sections. The tasks in the first two sections are required; the remaining may be optional, depending on which parameters are configured.

- Enabling or Disabling IKE, page SC-32 (required)
- Configuring IKE Policies, page SC-33 (required)
- Manually Configuring RSA Keys, page SC-35 (optional)
- Configuring ISAKMP Preshared Keys in ISAKMP Keyrings, page SC-39 (optional, depending on IKE parameters)
- Configuring Call Admission Control, page SC-41 (optional)
- Configuring Crypto Keyrings, page SC-45 (required)
Enabling or Disabling IKE

This task enables or disables the Internet Key Exchange protocol.
IKE is disabled by default. IKE need not be enabled for individual interfaces, but it is enabled globally for all interfaces at the router.

**SUMMARY STEPS**

1. configure
2. crypto isakmp
3. no crypto isakmp
4. end
   or
   commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

<table>
<thead>
<tr>
<th>Step 2 crypto isakmp</th>
<th>Globally enables IKE at the peer router.</th>
</tr>
</thead>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# crypto isakmp
```
Configuring IKE Policies

This task configures IKE policies.

**SUMMARY STEPS**

1. configure
2. crypto isakmp policy *priority*
3. encryption {192-aes | 256-aes | 3des | aes | des}
4. hash {sha | md5}
5. authentication {pre-share | rsa-sig | rsa-encr}
6. group {1 | 2 | 5}
7. lifetime seconds
8. end
   or
   commit
9. show crypto isakmp policy
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
configure | Enters global configuration mode. |
| **Example:**
RP/0/RSP0/CPU0:router# configure | |
| **Step 2**
crypto isakmp policy priority | Identifies the policy to create. 
Each policy is uniquely identified by the priority number you assign, which can be from 1 to 10000. This command places the router in ISAKMP policy configuration mode. |
| **Example:**
RP/0/RSP0/CPU0:router(config)# crypto isakmp policy 5 | |
| **Step 3**
encryption (192-aes | 256-aes | 3des | aes | des) | Specifies the encryption algorithm:
• **192-aes**—Advanced Encryption Standard, 192-bit keys
• **256-aes**—Advanced Encryption Standard, 256-bit keys
• **3des**—Three-key triple Data Encryption Standard
• **aes**—Advanced Encryption Standard, 128-bit keys
• **des**—Data Encryption Standard, 56-bit keys |
| **Example:**
RP/0/RSP0/CPU0:router(config-isakmp)# encryption aes | |
| **Step 4**
hash (sha | md5) | Specifies the hash algorithm.
• **sha**—Secure-hash-algorithm
• **md5**—Message-digest-5 |
| **Example:**
RP/0/RSP0/CPU0:router(config-isakmp)# hash md5 | |
| **Step 5**
authentication (pre-share | rsa-sig | rsa-encr) | Specifies the authentication method for this policy as either a pre-shared key, an RSA-encryption, or an RSA signature. |
| **Example:**
RP/0/RSP0/CPU0:router(config-isakmp)# authentication rsa-sig | |
| **Step 6**
group (1 | 2 | 5) | Specifies the Diffie-Hellman group identifier. |
| **Example:**
RP/0/RSP0/CPU0:router(config-isakmp)# group 5 | |
| **Step 7**
lifetime seconds | Specifies the lifetime of the security association. The range, in seconds, is from 60 to 86400. |
| **Example:**
RP/0/RSP0/CPU0:router(config-isakmp)# lifetime 50000 | |
Manually Configuring RSA Keys

Manually configure RSA keys when you specify RSA encrypted nonces as the authentication method in an IKE policy and you are not using a CA.

To manually configure RSA keys, perform these tasks at each IPSec peer that uses RSA encrypted nonces in an IKE policy:

- Generating RSA Keys, page SC-35
- Configuring ISAKMP Identity, page SC-36
- Configuring RSA Public Keys of All the Other Peers, page SC-37

Generating RSA Keys

For instructions on how to generate RSA keys, see the “Generating an RSA Key Pair” section on page SC-7 in this publication.
Configuring ISAKMP Identity

This task configures the ISAKMP identity of a peer.

Remember to repeat these tasks at each peer that uses preshared keys in an IKE policy.

SUMMARY STEPS

1. configure
2. crypto isakmp identity {address | hostname}
3. host hostname address1 [address2...address8]
4. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2 crypto isakmp identity {address</td>
<td>hostname}</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# crypto isakmp identity address</td>
</tr>
</tbody>
</table>
Configuring RSA Public Keys of All the Other Peers

This task configures the RSA public keys of all the other peers.

Remember to repeat these tasks at each peer that uses RSA encrypted nonces in an IKE policy.

SUMMARY STEPS

1. configure
2. crypto keyring keyring-name [vrf fvrf-name]
3. rsa-pubkey [address address | name fqdn] [encryption | signature]
4. address ip-address
5. key-string key-string
6. quit
7. end
   or
   commit
8. show crypto key pubkey-chain rsa [name key-name | address key-address]
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto keyring keyring-name [vrf fvrf-name]</td>
<td>Defines a crypto keyring during IKE authentication</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# crypto keyring vpnkeyring</td>
<td>• Enters keyring configuration mode.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-keyring)#</td>
<td>• Use the keyring-name argument to specify the name of the crypto keyring.</td>
</tr>
<tr>
<td></td>
<td>• (Optional) Use the vrf keyword to specify that the front door virtual routing and forwarding (FVRF) name is the keyring that is referenced.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> rsa-pubkey (address address</td>
<td>Defines the Rivest, Shamir, and Adelman (RSA) manual key to be used for encryption or signature during Internet Key Exchange (IKE) authentication.</td>
</tr>
<tr>
<td></td>
<td>name fqdn) [encryption</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>encryption</td>
</tr>
<tr>
<td></td>
<td>signature]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-keyring)# rsa-pubkey name host.vpn.com</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pubkey)</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address ip-address</td>
<td>Specifies the remote peer’s IP address.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pubkey)# address 10.5.5.1</td>
<td>• You can use this command if you used a fully qualified domain name to name the remote peer.</td>
</tr>
<tr>
<td><strong>Step 5</strong> key-string key-string</td>
<td>Specifies the remote peer’s RSA public key.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pubkey)# key-string 005C300D 06092A86 4886F70D 01010105</td>
<td>• This is the key previously displayed by the remote peer’s administrator when the remote router’s RSA keys were generated.</td>
</tr>
<tr>
<td>...</td>
<td>• To avoid mistakes, you should cut and paste the key data (instead of attempting to enter in the data).</td>
</tr>
<tr>
<td></td>
<td>• Enter a key on each line. You must enter the key-string command before the key.</td>
</tr>
<tr>
<td></td>
<td>• When you have finished specifying the remote peer’s RSA key, return to global configuration mode by entering quit at the public key configuration prompt.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6</td>
<td><code>quit</code></td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-pubkey)# quit</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><code>end</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# end</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# commit</code></td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>`show crypto pubkey-chain rsa [name key-name</td>
<td>address key-address]`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# show crypto pubkey-chain rsa</code></td>
<td></td>
</tr>
</tbody>
</table>

### Configuring ISAKMP Preshared Keys in ISAKMP Keyrings

This task configures ISAKMP preshared keys in ISAKMP keyrings.

### Prerequisites

To configure ISAKMP preshared keys in ISAKMP keyrings, perform these tasks at each peer that uses preshared keys in an IKE policy:

- Set the ISAKMP identity of each peer. Each peer's identity should be set either to its hostname or by its IP address. By default, a peer's identity is set to its IP address. Setting ISAKMP identities is described in the “Configuring ISAKMP Identity” section on page SC-36.

- Specify the shared keys at each peer. Note that a given preshared key is shared between two peers. At a given peer you could specify the same key to share with multiple remote peers; however, a more secure approach is to specify different keys to share between different pairs of peers.

You must specify the support for masked preshared keys. Remember to repeat these tasks at each peer that uses preshared keys in an IKE policy.
SUMMARY STEPS

1. configure
2. crypto keyring keyring-name [vrf fvrf-name]
3. pre-shared-key {address address [mask] | hostname hostname} key key
4. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
</tbody>
</table>

**Step 2** crypto keyring keyring-name [vrf fvrf-name]

| Example: RP/0/RSP0/CPU0:router(config)# crypto keyring vpnkeyring |
| RP/0/RSP0/CPU0:router(config-keyring)# |

Defines a crypto keyring during IKE authentication.
- Use the keyring-name argument to specify the name of the crypto keyring.
- (Optional) Use the vrf keyword to specify that the front door virtual routing and forwarding (FVRF) name is the keyring that is referenced.

**Step 3** pre-shared-key {address address [mask] | hostname hostname} key key

| Example: RP/0/RSP0/CPU0:router(config-keyring)# |
| pre-shared-key address 10.72.23.11 key vpnkey |
| RP/0/RSP0/CPU0:router(config-keyring)# |
| pre-shared-key hostname mycisco.com key vpnkey |

Defines a preshared key for IKE authentication.
- Use the address keyword to specify the IP address of the remote peer or a subnet and mask.
- (Optional) Use the mask argument to match the range of the address.
- Use the hostname keyword to specify the fully qualified domain name (FQDN) of the peer.
- Use the key keyword to specify the key.
Configuring Call Admission Control

These tasks are used to configure Call Admission Control (CAC):

- Configuring the IKE Security Association Limit, page SC-41
- Configuring the System Resource Limit, page SC-43

Configuring the IKE Security Association Limit

This task configures the IKE security admission limit.

SUMMARY STEPS

1. configure
2. crypto isakmp call admission limit {in-negotiation-sa number | sa number}
3. end
   or commit
4. show crypto isakmp call admission statistics

Command or Action | Purpose
--- | ---
end or commit | Saves configuration changes.

- When you issue the end command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?
  [cancel];

  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
## Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto isakmp call admission limit {in-negotiation-sa number</td>
<td>sa number}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Specifies the maximum number of IKE SAs that the router can establish before IKE begins rejecting new SA requests.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# crypto isakmp call admission limit sa 25</td>
<td>• Use the <strong>in-negotiation-sa</strong> keyword to specify the maximum number of in-negotiation (embryonic) IKE security associations (SAs) that the router can establish before IKE begins rejecting new SA requests. The range for the number argument is from 1 to 100000.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>sa</strong> keyword to specify the maximum number of active IKE SAs that the router can establish. The range for the <strong>number</strong> argument is from 1 to 100000.</td>
</tr>
</tbody>
</table>
## Configuring the System Resource Limit

This task configures the system resource limit.

### SUMMARY STEPS

1. `configure`
2. `crypto isakmp call admission limit {cpu {total percent | ike percent}}`
3. `end` or `commit`
4. `show crypto isakmp call admission statistics`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto isakmp call admission limit {cpu (total percent</td>
<td>ike percent)}</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# crypto isakmp call admission limit cpu total 90</td>
<td>• Use the cpu keyword to specify the total resource limit for the CPU usage.</td>
</tr>
<tr>
<td></td>
<td>• Use the total keyword to specify the maximum total CPU usage to accept new calls. The range for the percent argument is from 1 to 100.</td>
</tr>
<tr>
<td></td>
<td>• Use the ike keyword to specify the maximum IKE CPU usage to accept new calls. The range for the percent argument is from 1 to 100.</td>
</tr>
<tr>
<td><strong>Step 3</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# end or RP/0/RSP0/CPU0:router(config)# commit</td>
<td>• When you issue the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>— Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>— Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>— Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
<tr>
<td><strong>Step 4</strong> show crypto isakmp call admission statistics</td>
<td>Monitors crypto CAC statistics.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# show crypto isakmp call admission statistics</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Crypto Keyrings

A crypto keyring is a repository of preshared and Rivest, Shamir, and Adelman (RSA) public keys. The router can have zero or more keyrings. Each keyring optionally allows the specification of a VRF in which the keys defined in the keyring belong.

This task configures crypto keyrings.

Crypto Keyrings Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring crypto keyrings:

- The VRF associated with a crypto keyring cannot be changed. A different keyring must be configured with the new VRF value.
- Address overlapping in a keyring is not allowed and must be enforced during configuration.
- A crypto keyring is attached to one or more ISAKMP profiles and cannot be deleted while in use.

SUMMARY STEPS

1. configure
2. crypto keyring keyring-name [vrf fvrf-name]
3. description string
4. local-address ip-address
5. pre-shared-key {address address [mask] | hostname hostname} key key
6. rsa-pubkey {address address | name fqdn} [encryption | signature]
7. key-string key-string
8. quit
9. end
   or
   commit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto keyring keyring-name [vrf fvrf-name]</td>
<td>Defines a crypto keyring to be used during IKE authentication.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# crypto keyring vpnkey</td>
<td>- Use the <code>keyring-name</code> argument as the name of the crypto keyring.</td>
</tr>
<tr>
<td></td>
<td>- Use the <code>vrf</code> keyword to specify that the front door virtual routing and forwarding (FVRF) name is the keyring that is referenced. The <code>fvrf-name</code> argument must match the FVRF name that was defined during a VRF configuration.</td>
</tr>
<tr>
<td><strong>Step 3</strong> description string</td>
<td>Creates a one-line description for a keyring.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-keyring)# description this is a sample keyring</td>
<td>- Use the <code>string</code> argument to specify the character string that describes the keyring.</td>
</tr>
<tr>
<td><strong>Step 4</strong> local-address ip-address</td>
<td>Limits the scope of an ISAKMP keyring configuration to a local termination address or interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-keyring)# local-address 10.40.1.1</td>
<td>- Use the <code>ip-address</code> argument to specify the IP address to which to bind.</td>
</tr>
<tr>
<td><strong>Step 5</strong> pre-shared-key {address address [mask]</td>
<td>hostname hostname} key key</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-keyring)# pre-shared-key address 10.72.23.11 key vpnkey</td>
<td>- Use the <code>address</code> keyword to specify the IP address of the remote peer or a subnet and mask. The <code>mask</code> argument is optional.</td>
</tr>
<tr>
<td></td>
<td>- Use the <code>hostname</code> keyword to specify the fully qualified domain name (FQDN) of the peer.</td>
</tr>
<tr>
<td></td>
<td>- Use the <code>key</code> keyword to specify the secret.</td>
</tr>
</tbody>
</table>
### Command or Action

| Step 6 | rsa-pubkey {address address | name fqdn} [encryption | signature] |
|--------|-------------------------------------------------------------|

**Example:**

```
RP/0/RSP0/CPU0:router(config-keyring)# rsa-pubkey
name host.vpn.com
```

**Purpose:**

Defines a Rivest, Shamir, and Adelman (RSA) public key by address or hostname.

- Use the `address` keyword to specify the IP address of the RSA public key of the remote peer. The `address` argument is the IP address of the remote RSA public key of the remote peer that you manually configure.
- Use the `name` keyword to specify the FQDN of the peer.
- (Optional) Use the `encryption` keyword to specify that the key is used for encryption.
- (Optional) Use the `signature` keyword to specify that the key is used for a signature. The `signature` keyword is the default.

<table>
<thead>
<tr>
<th>Step 7</th>
<th>key-string key-string</th>
</tr>
</thead>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-pubkey)# key-string
005C300D 06092A86 4886F70D 01010105
```

**Purpose:**

Manually specifies the RSA public key of a remote peer.

<table>
<thead>
<tr>
<th>Step 8</th>
<th>end or commit</th>
</tr>
</thead>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# end
or
RP/0/RSP0/CPU0:router(config)# commit
```

**Purpose:**

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring IP Security VPN Monitoring

The following sections describe how to configure IP Security (IPSec) VPN monitoring:

- Adding the Description of an IKE Peer, page SC-48 (optional)
- Clearing a Crypto Session, page SC-49 (optional)

Adding the Description of an IKE Peer

This task allows you to add the description of an IKE peer to an IPSec VPN session.

**SUMMARY STEPS**

1. `configure`
2. `crypto isakmp peer {address ip-address | hostname hostname} [description string | vrf fvrf-name]`
3. `description string`
4. `end` or `commit`
5. `show crypto isakmp peers [ip-address | vrf fvrf-name]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> `crypto isakmp peer {address ip-address</td>
<td>hostname hostname} [description string</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# crypto isakmp peer address 10.40.40.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isakmp-peer)</td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>description string</code></td>
<td>Adds a text string description for an IKE peer.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isakmp-peer)# description citeA</td>
<td>• Description of peer may be up to 80 characters.</td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4**

- `end`
  - or
  - `commit`

**Example:**

RP/0/RSP0/CPU0:router(config-isakmp-peer)# end
RP/0/RSP0/CPU0:router(config-isakmp-peer)# commit

### Purpose

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.

### Step 5

**show crypto isakmp peers [ip-address | vrf vrf-name]**

**Example:**

RP/0/RSP0/CPU0:router# show crypto isakmp peers

**Purpose**

Displays peer descriptions.

---

### Clearing a Crypto Session

Use the `clear crypto session` command in EXEC mode to delete the crypto sessions (IP Security [IPSec] and Internet Key Exchange [IKE] security associations [SAs]) for users and groups.

### How to Configure the ISAKMP Profile for a Service Interface

This task configures the ISAKMP profile (which is a repository of commands for a set of peers) for a service interface.

### SUMMARY STEPS

1. `configure`
2. `crypto isakmp profile [local profile-name]`
3. `description string`
4. `keepalive disable`
5. `self-identity [address | fqdn | user-fqdn user-fqdn]`
### How to Configure the ISAKMP Profile for a Service Interface

6. **keyring** keyring-name
7. **match identity** {group group-name | address address [mask] vrf [vrf] | host hostname | host domain domain-name | user username | user domain domain-name}
8. **set interface tunnel-ipsec** intf-index
9. **set ipsec-profile** profile-name
10. **end**
    or
    **commit**

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto isakmp profile {local profile-name}</td>
<td>Defines an ISAKMP profile and audits IPSec user sessions.</td>
</tr>
</tbody>
</table>
| **Example:** RP/0/RSP0/CPU0:router(config)# crypto isakmp profile local vpnprofile RP/0/RSP0/CPU0:router(config-isa-prof)# | - (Required) Use the **local** keyword to specify that the profile is used for locally sourced or terminated traffic.  
    **Note** The **local** keyword is required for use of the **set ipsec-profile** command later in this procedure. |
| **Step 3** description string | Creates a description for a keyring. |
| **Example:** RP/0/RSP0/CPU0:router(config-isa-prof)# description this is a sample profile | - Use the **string** argument to specify the character string that describes the keyring. |
| **Step 4** keepalive disable | Lets the gateway send DPD messages to the Cisco IOS XR peer. |
| **Example:** RP/0/RSP0/CPU0:router(config-isa-prof)# keepalive disable | - Use the **disable** keyword to disable the keepalive global declarations. |
| **Step 5** self-identity {address | fqdn | user-fqdn user-fqdn} | Defines the identity that the local IKE uses to identify itself to the remote peer. |
| **Example:** RP/0/RSP0/CPU0:router(config-isa-prof)# self-identity user-fqdn user@vpn.com | - Use the **address** keyword to specify the IP address of the local endpoint.  
- Use the **fqdn** keyword to specify the fully qualified domain name (FQDN) of the host.  
- Use the **user-fqdn** keyword to specify that the user FQDN was sent to the remote endpoint. |
### Command or Action

**Step 6**
```
keyring keyring-name
```

**Example:**
```
RP/0/RSP0/CPU0:router(config-isa-prof)# keyring vpnkeyring
```

**Step 7**
```
match identity {group group-name | address address [mask] vrf [fvrf] | host hostname | host domain domain-name | user username | user domain domain-name}
```

**Example:**
```
RP/0/RSP0/CPU0:router(config-isa-prof)# match identity group vpnpgrp
```

**Step 8**
```
set interface tunnel-ipsec intf-index
```

**Example:**
```
RP/0/RSP0/CPU0:router(config-isa-prof-match)# set interface tunnel-ipsec 50
```

### Purpose

**Step 6**
- Configures a keyring with an ISAKMP profile.
  - Use the `keyring-name` argument to specify the keyring name, which must match the keyring name that was defined in the global configuration.

**Step 7**
- Matches the identity from a peer in an ISAKMP profile.
  - Use the `group` keyword to specify a Unity group that matches identification (ID) type ID_KEY_ID. If RSA signatures are used, the `group-name` argument matches the organizational unit (OU) field of the distinguished name (DN).
  - Use the `address` keyword to match the address argument with the ID type ID_IPV4_ADDR.
  - Use the `mask` argument to specify a range of addresses.
  - Use the `vrf` keyword to specify the front door VPN routing and forwarding (VRF) of the peer.
  - Use the `fvrf` argument to match the address in the front door virtual router forwarding (FVRF) Virtual Private Network (VPN) space.
  - Use the `host` keyword to specify an identity that matches the type ID_FQDN, whose fully qualified domain name (FQDN) ends with the domain name.
  - Use the `host domain` keyword to specify an identity that matches type ID_USER_FQDN. When the `user domain` keyword is present, all users having identities of the type ID_USER_FQDN and ending with `domain-name` are matched.
  - Use the `user` keyword to specify an identity that matches the FQDN.

**Step 8**
- Predetermines the interface instance when IKE negotiates for IPSec service associations (SAs) for the traffic that is locally sourced or terminated and the local endpoint is the IKE responder.
  - Use the `intf-index` argument to set the range from 0 to 4294967295.
### How to Configure a Periodic Dead Peer Detection Message

This task configures a periodic dead peer detection (DPD) message.

**SUMMARY STEPS**

1. configure
2. crypto isakmp keepalive seconds retry-seconds [periodic | on-demand]
3. end
   or
   commit

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> set ipsec-profile profile-name</td>
<td>Predefines the IPSec profile instance when IKE negotiates for IPSec service associations (SAs) for the traffic that is locally sourced or terminated and the local endpoint is the IKE responder.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isa-prof-match)# set ipsec-profile myprofile</td>
<td>• Use the <em>profile-name</em> argument to set the name of the IPSec profile.</td>
</tr>
<tr>
<td><strong>Step 10</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isa-prof-match)# end or RP/0/RSP0/CPU0:router(config-isa-prof-match)# commit</td>
<td>• When you issue the <em>end</em> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Uses the IKE security association (SA) feature to provide a mechanism to detect loss of connectivity between two IP Security (IPSec) peers.</td>
</tr>
<tr>
<td>`crypto isakmp keepalive seconds retry-seconds [periodic</td>
<td>on-demand]`</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# crypto isakmp keepalive 20 20 on-demand</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# end or RP/0/RSP0/CPU0:router(config)# commit</td>
<td></td>
</tr>
</tbody>
</table>

- When you issue the `end` command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.
  
  - Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuration Examples for Implementing IKE Security Protocol

This section provides the following configuration examples:

- Creating IKE Policies: Example, page SC-54
- Limiting an IKE Peer to a Particular Policy Set Based on Local IP Address: Example, page SC-55

Creating IKE Policies: Example

This example shows how to create two IKE policies with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority.

crypto isakmp policy 15
    encryption 3des
    hash md5
    authentication rsa-sig
    group 2
    lifetime 5000

crypto isakmp policy 20
    authentication pre-share
    lifetime 10000

In the example, the encryption des of policy 20 would not appear in the written configuration because this is the default value for the encryption algorithm parameter.

If the show crypto isakmp policy command is issued with this configuration, the output is as follows:

Protection suite priority 15
    encryption algorithm: 3DES - Data Encryption Standard (168 bit keys)
    hash algorithm: Message Digest 5
    authentication method: Rivest-Shamir-Adelman Signature
    Diffie-Hellman group: #2 (1024 bit)
    lifetime: 5000 seconds, no volume limit

Protection suite priority 20
    encryption algorithm: DES - Data Encryption Standard (56 bit keys)
    hash algorithm: Secure Hash Standard
    authentication method: preshared Key
    Diffie-Hellman group: #1 (768 bit)
    lifetime: 10000 seconds, no volume limit

Default protection suite
    encryption algorithm: DES - Data Encryption Standard (56 bit keys)
    hash algorithm: Secure Hash Standard
    authentication method: Rivest-Shamir-Adelman Signature
    Diffie-Hellman group: #1 (768 bit)
    lifetime: 86400 seconds, no volume limit

Note: Although the output shows “no volume limit” for the lifetimes, you can configure only a time lifetime (such as 86,400 seconds); volume-limit lifetimes are not configurable.
Limiting an IKE Peer to a Particular Policy Set Based on Local IP Address: Example

The first part consists of selecting an ISAKMP policy related to the encryption method and identifying the SVI tunnel source. Users connecting to IP address 1.1.1.1 in the following example experience DES as the ISAKMP policy. However, users connecting to IP address 2.2.2.2 experience only AES as the ISAKMP policy.

More than one ISAKMP policy, or more than one IP address, can be used for matches. The rest of configuration remains the same; in other words, the configuration of the ISAKMP profile that matches a group name set to an SVI.

In this particular example, two policies have been configured in the policy set (policy 10 and 20).

Note that the SVI1 and SVI2 tunnel sources are respectively identified in bold as local-address 10.10.1.1 and local-address 10.10.2.2 in the example.

```
RP/0/0/CPU0:router: crypto isakmp policy 10
RP/0/0/CPU0:router(config-isakmp)# encryption des << restricts use to DES only
RP/0/0/CPU0:router(config-isakmp)# group 2
RP/0/0/CPU0:router(config-isakmp)# authentication pre-share

RP/0/0/CPU0:router(config)# crypto isakmp policy 20
RP/0/0/CPU0:router(config-isakmp)# encryption aes << restricts use to AES only
RP/0/0/CPU0:router(config-isakmp)# group 2
RP/0/0/CPU0:router(config-isakmp)# authentication pre-share

RP/0/0/CPU0:router(config)# crypto isakmp policy-set policy_1 << match ID
RP/0/0/CPU0:router(config-isakmp-pol-set)# policy 10 << routing priority
RP/0/0/CPU0:router(config-isakmp-pol-set)# match identity local-address 10.10.1.1

RP/0/0/CPU0:router(config)# crypto isakmp policy-set policy_2 << match ID
RP/0/0/CPU0:router(config-isakmp-pol-set)# policy 20
RP/0/0/CPU0:router(config-isakmp-pol-set)# match identity local-address 10.10.2.2
RP/0/0/CPU0:router(config-isakmp-pol-set)# commit
RP/0/0/CPU0:router(config-isakmp-pol-set)# exit
RP/0/0/CPU0:router(config-isakmp)#
```
Additional References

The following sections provide references related to implementing the IKE security protocol.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKE security protocol commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference</td>
</tr>
<tr>
<td>IPSec-related object tracking commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference</td>
</tr>
<tr>
<td>Object tracking configuration procedures, including examples</td>
<td>Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide</td>
</tr>
<tr>
<td>IPSec network security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>IPSec Network Security Commands on Cisco ASR 9000 Series Routers module in Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>
RFCS

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2401</td>
<td>Security Architecture for the Internet Protocol</td>
</tr>
<tr>
<td>RFC 2402</td>
<td>IP Authentication Header</td>
</tr>
<tr>
<td>RFC 2403</td>
<td>The Use of HMAC-MD5-96 within ESP and AH</td>
</tr>
<tr>
<td>RFC 2404</td>
<td>The Use of HMAC-SHA-1-96 within ESP and AH</td>
</tr>
<tr>
<td>RFC 2405</td>
<td>The ESP DES-CBC Cipher Algorithm With Explicit IV</td>
</tr>
<tr>
<td>RFC 2406</td>
<td>IP Encapsulating Security Payload (ESP)</td>
</tr>
<tr>
<td>RFC 2407</td>
<td>The Internet IP Security Domain of Interpretation for ISAKMP</td>
</tr>
<tr>
<td>RFC 2408</td>
<td>Internet Security Association and Key Management Protocol (ISAKMP)</td>
</tr>
<tr>
<td>RFC 2409</td>
<td>The Internet Key Exchange (IKE)</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>