CONTENTS

CHAPTER 1

Read Me First  1

CHAPTER 2

Configuring Security for VPNs with IPsec  3

Finding Feature Information  3
Prerequisites for Configuring Security for VPNs with IPsec  4
Restrictions for Configuring Security for VPNs with IPsec  4
Information About Configuring Security for VPNs with IPsec  5
Supported Standards  5
Supported Encapsulation  6
IPsec Functionality Overview  6
IKEv1 Transform Sets  8
IKEv2 Transform Sets  8
Transform Sets: A Combination of Security Protocols and Algorithms  9
About Transform Sets  9
Cisco IOS Suite-B Support for IKE and IPsec Cryptographic Algorithms  13
Suite-B Requirements  14
Where to Find Suite-B Configuration Information  14
How to Configure IPsec VPNs  15
Creating Crypto Access Lists  15
What to Do Next  16
Configuring Transform Sets for IKEv1 and IKEv2 Proposals  16
Restrictions  16
Configuring Transform Sets for IKEv1  17
What to Do Next  18
Configuring Transform Sets for IKEv2  18
Transform Sets for IKEv2 Examples  20
What to Do Next  21
Creating Crypto Map Sets  21
Chapter 3: IPsec Virtual Tunnel Interfaces

Finding Feature Information 39
Restrictions for IPsec Virtual Tunnel Interfaces 40
Information About IPsec Virtual Tunnel Interfaces 41
  Benefits of Using IPsec Virtual Tunnel Interfaces 42
  Static Virtual Tunnel Interfaces 42
  Dynamic Virtual Tunnel Interfaces 43
Traffic Encryption with the IPsec Virtual Tunnel Interface 44
Multi-SA Support for Dynamic Virtual Interfaces 46
Dynamic Virtual Tunnel Interface Life Cycle 46
Routing with IPsec Virtual Tunnel Interfaces 47
FlexVPN Mixed Mode Support 47
Auto Tunnel Mode Support in IPsec 47
IPSec Mixed Mode Support for VTI 47
How to Configure IPsec Virtual Tunnel Interfaces 48
  Configuring Static IPsec Virtual Tunnel Interfaces 48
  Configuring BGP over IPsec Virtual Tunnel Interfaces 50
  Configuring Dynamic IPsec Virtual Tunnel Interfaces 52
  Configuring Multi-SA Support for Dynamic Virtual Tunnel Interfaces Using IKEv1 54
Configuring IPsec Mixed Mode Support for SVTIs 58
Configuring IPsec Mixed Mode Support for Dynamic VTIs 60
Configuration Examples for IPsec Virtual Tunnel Interfaces 62
Example: Static Virtual Tunnel Interface with IPsec 62
  Example: Verifying the Results for the IPsec Static Virtual Tunnel Interface 64
Example: VRF-Aware Static Virtual Tunnel Interface 65
Example: Static Virtual Tunnel Interface with QoS 65
Example: Static Virtual Tunnel Interface with Virtual Firewall 66
Example: Dynamic Virtual Tunnel Interface Easy VPN Server 67
  Example: Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server 68
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under a Virtual Template 68
Example: VRF-Aware IPsec with Dynamic VTI When VRF Is Configured Under a Virtual Template with the Gateway Option in an IPsec Profile 69
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile 70
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile and a Gateway Option in an IPsec Profile 71
Example: VRF-Aware IPsec with a Dynamic VTI When a VRF Is Configured Under Both a Virtual Template and an ISAKMP Profile 71
Example: Dynamic Virtual Tunnel Interface with Virtual Firewall 72
Example: Dynamic Virtual Tunnel Interface with QoS 73
Additional References for IPsec Virtual Tunnel Interface 74
Feature Information for IPsec Virtual Tunnel Interfaces 75

CHAPTER 4

Deleting Crypto Sessions of Revoked Peer Certificates 79
Finding Feature Information 79
Restrictions for Deleting Crypto Sessions of Revoked Peer Certificates 79
Information About Deleting Crypto Sessions of Revoked Peer Certificates 80
  How a Crypto Session is Deleted 80
How to Enable Deletion of Crypto Sessions for Revoked Peer Certificates 80
  Enabling Deletion of Crypto Sessions 80
  Verifying the Delete Crypto Session Capability for a Revoked Peer Certificate 82
Configuration Examples for Deleting Crypto Sessions of Revoked Peer Certificates 82
Additional References 108
Feature Information for IPv6 over IPv4 GRE Tunnel Protection 109

CHAPTER 7
RFC 430x IPsec Support 111
Finding Feature Information 111
Information About RFC 430x IPsec Support 111
   RFC 430x IPsec Support Phase 1 111
   RFC 430x IPsec Support Phase 2 112
How to Configure RFC 430x IPsec Support 112
   Configuring RFC 430x IPsec Support Globally 112
   Configuring RFC 430x IPsec Support Per Crypto Map 113
Configuration Examples for RFC 430x IPsec Support 116
   Example: Configuring RFC 430x IPsec Support Globally 116
   Example: Configuring RFC 430x IPsec Support Per Crypto Map 116
Additional References for RFC 430x IPsec Support 117
Feature Information for RFC 430x IPsec Support 118
Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E (for Catalyst Switching) and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.
CHAPTER 2

Configuring Security for VPNs with IPsec

This module describes how to configure basic IPsec VPNs. IPsec is a framework of open standards developed by the IETF. It provides security for the transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices ("peers"), such as Cisco routers.

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- Finding Feature Information, page 3
- Prerequisites for Configuring Security for VPNs with IPsec, page 4
- Restrictions for Configuring Security for VPNs with IPsec, page 4
- Information About Configuring Security for VPNs with IPsec, page 5
- How to Configure IPsec VPNs, page 15
- Configuration Examples for IPsec VPN, page 33
- Additional References for Configuring Security for VPNs with IPsec, page 34
- Feature Information for Configuring Security for VPNs with IPsec, page 36
- Glossary, page 37

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for Configuring Security for VPNs with IPsec

IKE Configuration
You must configure Internet Key Exchange (IKE) as described in the module Configuring Internet Key Exchange for IPsec VPNs.

Note
If you decide not to use IKE, you must still disable it as described in the module Configuring Internet Key Exchange for IPsec VPNs.

Ensure Access Lists Are Compatible with IPsec
IKE uses UDP port 500. The IPsec encapsulating security payload (ESP) and authentication header (AH) protocols use protocol numbers 50 and 51, respectively. Ensure that your access lists are configured so that traffic from protocol 50, 51, and UDP port 500 are not blocked at interfaces used by IPsec. In some cases, you might need to add a statement to your access lists to explicitly permit this traffic.

Restrictions for Configuring Security for VPNs with IPsec

Cisco IPsec Policy Map MIB
The MIB OID objects are displayed only when an IPsec session is up.

Discontiguous Access Control Lists
Crypto maps using access control lists (ACLs) that have discontiguous masks are not supported.

IPv4 Packets with IP Options Set
The following platforms do not support encrypting IPv4 packets with IP options set:
Cisco ASR1001 and ASR1000 routers with ESP-5, ESP-10, ESP-20, and ESP-40.

Physical Interface and Crypto Map
A crypto map on a physical interface is not supported, if the physical interface is the source interface of a tunnel protection interface.

NAT Configuration
If you use Network Address Translation (NAT), you should configure static NAT so that IPsec works properly. In general, NAT should occur before the router performs IPsec encapsulation; in other words, IPsec should work with global addresses.

Unicast IP Datagram Application Only
IPsec can be applied to unicast IP datagrams only. Because the IPsec Working Group has not yet addressed the issue of group key distribution, IPsec does not currently work with multicasts or broadcast IP datagrams.
Unsupported Interface Types

- Crypto VPNs are not supported on the bridge domain interfaces (BDI).
- Crypto maps are not supported on tunnel interface and port-channel interface.

Information About Configuring Security for VPNs with IPsec

Supported Standards

Cisco implements the following standards with this feature:

- **IPsec**—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; IPsec uses IKE to handle negotiation of protocols and algorithms based on the local policy, and generate the encryption and authentication keys to be used by IPsec. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

  ![Note](Note.png)

  The term IPsec is sometimes used to describe the entire protocol of IPsec data services and IKE security protocols, and is also sometimes used to describe only the data services.

- **IKE (IKEv1 and IKEv2)**—A hybrid protocol that implements Oakley and SKEME key exchanges inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. While IKE is used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of IPsec peers, negotiates IPsec security associations, and establishes IPsec keys.

  ![Note](Note.png)

  Cisco no longer recommends using DES, 3DES, MD5 (including HMAC variant), and Diffie-Hellman (DH) groups 1, 2 and 5; instead, you should use AES, SHA and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- **AES**—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is a privacy transform for IPsec and IKE and has been developed to replace DES. AES is designed to be more secure than DES. AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.

- **DES**—Data Encryption Standard. An algorithm that is used to encrypt packet data. Cisco software implements the mandatory 56-bit DES-CBC with Explicit IV. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet. For backwards compatibility, Cisco IOS IPsec also implements the RFC 1829 version of ESP DES-CBC.
Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Cisco no longer recommends Triple DES (3DES).

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

**Note**

- SHA-2 and SHA-1 family (HMAC variant)—Secure Hash Algorithm (SHA) 1 and 2. Both SHA-1 and SHA-2 are hash algorithms used to authenticate packet data and verify the integrity verification mechanisms for the IKE protocol. HMAC is a variant that provides an additional level of hashing. SHA-2 family adds the SHA-256 bit hash algorithm and SHA-384 bit hash algorithm. This functionality is part of the Suite-B requirements that comprises four user interface suites of cryptographic algorithms for use with IKE and IPsec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support. SHA-2 for ISAKMP is supported in Cisco IOS XE 15.3(3)S and later.

- Diffie-Hellman—A public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. It supports 768-bit (the default), 1024-bit, 1536-bit, 2048-bit, 3072-bit, and 4096-bit DH groups. It also supports a 2048-bit DH group with a 256-bit subgroup, and 256-bit and 384-bit elliptic curve DH (ECDH). Cisco recommends using 2048-bit or larger DH key exchange, or ECDH key exchange.

- MD5 (Hash-based Message Authentication Code (HMAC) variant)—Message digest algorithm 5 (MD5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPsec as implemented in Cisco software supports the following additional standards:

- AH—Authentication Header. A security protocol, which provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).

- ESP—Encapsulating Security Payload. A security protocol, which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

**Supported Encapsulation**

IPsec works with the following serial encapsulations: Frame Relay, High-Level Data-Links Control (HDLC), and PPP.

IPsec also works with Generic Routing Encapsulation (GRE) and IPinIP Layer 3, Data Link Switching+ (DLSw+), and Source Route Bridging (SRB) tunneling protocols; however, multipoint tunnels are not supported. Other Layer 3 tunneling protocols may not be supported for use with IPsec.

**IPsec Functionality Overview**

IPsec provides the following network security services. (In general, the local security policy dictates the use of one or more of these services.)
• Data confidentiality—The IPsec sender can encrypt packets before transmitting them across a network.
• Data integrity—The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
• Data origin authentication—The IPsec receiver can authenticate the source of the sent IPsec packets. This service is dependent upon the data integrity service.
• Anti-replay—The IPsec receiver can detect and reject replayed packets.

IPsec provides secure tunnels between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets by specifying the characteristics of these tunnels. When the IPsec peer recognizes a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer. (The use of the term tunnel in this chapter does not refer to using IPsec in tunnel mode.)

More accurately, these tunnels are sets of security associations (SAs) that are established between two IPsec peers. The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (AH or ESP).

With IPsec, you can define the traffic that needs to be protected between two IPsec peers by configuring access lists and applying these access lists to interfaces using crypto map sets. Therefore, traffic may be selected on the basis of the source and destination address, and optionally the Layer 4 protocol and port. (The access lists used for IPsec are only used to determine the traffic that needs to be protected by IPsec, not the traffic that should be blocked or permitted through the interface. Separate access lists define blocking and permitting at the interface.)

A crypto map set can contain multiple entries, each with a different access list. The crypto map entries are searched in a sequence—the router attempts to match the packet to the access list specified in that entry.

When a packet matches a permit entry in a particular access list, and the corresponding crypto map entry is tagged as cisco, connections are established, if necessary. If the crypto map entry is tagged as ipsec-isakmp, IPsec is triggered. If there is no SA that the IPsec can use to protect this traffic to the peer, IPsec uses IKE to negotiate with the remote peer to set up the necessary IPsec SAs on behalf of the data flow. The negotiation uses information specified in the crypto map entry as well as the data flow information from the specific access list entry. (The behavior is different for dynamic crypto map entries. See the Creating Dynamic Crypto Maps section later in this module.)

If the crypto map entry is tagged as ipsec-manual, IPsec is triggered. If there is no SA that IPsec can use to protect this traffic to the peer, the traffic is dropped. In this case, the SAs are installed via the configuration, without the intervention of IKE. If SAs do not exist, IPsec does not have all the necessary pieces configured.

Once established, the set of SAs (outbound to the peer) is then applied to the triggering packet and to subsequent applicable packets as those packets exit the router. "Applicable" packets are packets that match the same access list criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound SAs are used when processing the incoming traffic from that peer.

Multiple IPsec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams only need to be authenticated, while other data streams must both be encrypted and authenticated.

Access lists associated with IPsec crypto map entries also represent the traffic that the router needs protected by IPsec. Inbound traffic is processed against crypto map entries—if an unprotected packet matches a permit
entry in a particular access list associated with an IPsec crypto map entry, that packet is dropped because it was not sent as an IPsec-protected packet.

Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms, and other settings that can be applied to IPsec-protected traffic. During the IPsec SA negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

### IKEv1 Transform Sets

An Internet Key Exchange version 1 (IKEv1) transform set represents a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

If you change a transform set definition, the change is applied only to crypto map entries that reference the transform set. The change is not applied to existing security associations, but is used in subsequent negotiations to establish new SAs.

### IKEv2 Transform Sets

An Internet Key Exchange version 2 (IKEv2) proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at least an encryption algorithm, an integrity algorithm, and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, then the default proposal is used in the negotiation.

The default proposal is a collection of commonly used algorithms which are as follows:

```plaintext
encryption aes-cbc-128 3des
integrity sha1 md5
group 5 2
```

The transforms shown above translate to the following combinations in the following order of priority:

```plaintext
aes-cbc-128, sha, 5
aes-cbc-128, sha, 2
aes-cbc-128, md5, 5
aes-cbc-128, md5, 2
3des, sha1, 5
3des, sha1, 2
3des, md5, 5
3des, md5, 2
```

Although the `crypto ikev2 proposal` command is similar to the `crypto isakmp policy priority` command, the IKEv2 proposal differs as follows:

- An IKEv2 proposal allows configuration of one or more transforms for each transform type.
- An IKEv2 proposal does not have any associated priority.

**Note**

To use IKEv2 proposals in negotiation, they must be attached to IKEv2 policies. If a proposal is not configured, then the default IKEv2 proposal is used with the default IKEv2 policy.

When multiple transforms are configured for a transform type, the order of priority is from left to right.
A proposal with multiple transforms for each transform type translates to all possible combinations of transforms. If only a subset of these combinations is required, then they must be configured as individual proposals.

Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128, aes-cbc-192
Device(config-ikev2-proposal)# integrity sha1, sha256
Device(config-ikev2-proposal)# group 14

For example, the commands shown above translate to the following transform combinations:

- aes-cbc-128, sha1, 14
- aes-cbc-192, sha1, 14
- aes-cbc-128, sha256, 14
- aes-cbc-192, sha256, 14

To configure the first and last transform combinations, use the following commands:

Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128
Device(config-ikev2-proposal)# integrity sha1
Device(config-ikev2-proposal)# group 14
Device(config-ikev2-proposal)# exit
Device(config)# crypto ikev2 proposal proposal-2
Device(config-ikev2-proposal)# encryption aes-cbc-192
Device(config-ikev2-proposal)# integrity sha256
Device(config-ikev2-proposal)# group 14

### Transform Sets: A Combination of Security Protocols and Algorithms

**About Transform Sets**

Transform sets represent a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

You can specify multiple transform sets, and specify one or more of these transform sets in a crypto map entry. The transform set is used in the IPsec SA negotiation to protect the data flows specified by that crypto map entry’s access list.

During IPsec security association negotiations with IKE, peers search for an identical transform set for both peers. When such a transform set is found, it is selected and applied to the protected traffic as part of both peers’ IPsec SAs. (With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.)

If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change is not applied to existing security associations, but is used in subsequent negotiations to establish new SAs.
You cannot specify AH and ESP algorithms on the same transform set in Cisco IOS XE releases.

The table below shows allowed transform combinations.

**Table 1: Allowed Transform Combinations**

<table>
<thead>
<tr>
<th>Transform Type</th>
<th>Transform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AH Transform (Pick only one.)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ah-md5-hmac</td>
<td></td>
<td>AH with the MD5 (Message Digest 5) (an HMAC variant) authentication algorithm. (No longer recommended).</td>
</tr>
<tr>
<td>ah-sha-hmac</td>
<td></td>
<td>AH with the SHA (Secure Hash Algorithm) (an HMAC variant) authentication algorithm. A transform set combination of AH and ESP is not supported in Cisco IOS XE releases.</td>
</tr>
</tbody>
</table>
ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm.

The esp-gcm and esp-gmac transforms are ESPs with either a 128-bit or a 256-bit encryption algorithm. The default for either of these transforms is 128 bits.

Both esp-gcm and esp-gmac transforms cannot be configured together with any other ESP transform within the same crypto IPsec transform set using the `crypto ipsec transform-set` command.

The esp-gcm and esp-gmac combinations are not supported on the Cisco ASR 1001 routers with the following ESPs:

- ESP-5
- ESP-10
- ESP-20
- ESP-40

ESP with the 192-bit AES encryption algorithm.

ESP with the 256-bit AES encryption algorithm.

### Transform Sets: A Combination of Security Protocols and Algorithms

<table>
<thead>
<tr>
<th>Transform Type</th>
<th>Transform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESP Encryption Transform</td>
<td>esp-aes</td>
<td>ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm.</td>
</tr>
</tbody>
</table>
|                        | esp-gcm, esp-gmac | The esp-gcm and esp-gmac transforms are ESPs with either a 128-bit or a 256-bit encryption algorithm. The default for either of these transforms is 128 bits. Both esp-gcm and esp-gmac transforms cannot be configured together with any other ESP transform within the same crypto IPsec transform set using the `crypto ipsec transform-set` command. The esp-gcm and esp-gmac combinations are not supported on the Cisco ASR 1001 routers with the following ESPs:
  - ESP-5
  - ESP-10
  - ESP-20
  - ESP-40 |
|                        | esp-aes 192   | ESP with the 192-bit AES encryption algorithm.                                |
|                        | esp-aes 256   | ESP with the 256-bit AES encryption algorithm.                                |
### Transform Types

<table>
<thead>
<tr>
<th>Transform Type</th>
<th>Transform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>esp-des</td>
<td>ESP with the 56-bit Data Encryption Standard (DES) encryption algorithm. (No longer recommended).</td>
</tr>
<tr>
<td></td>
<td>esp-3des</td>
<td>ESP with the 168-bit DES encryption algorithm (3DES or Triple DES). (No longer recommended).</td>
</tr>
<tr>
<td></td>
<td>esp-null</td>
<td>Null encryption algorithm.</td>
</tr>
</tbody>
</table>

**Note**
- When using DES, 3DES, or GMAC transforms on Cisco ASR 1001-X and Cisco ASR 1002-X routers with ESP-100 or ESP-200, there might be a 30% performance degradation compared to AES and GCM.

<table>
<thead>
<tr>
<th>ESP Authentication Transform (Pick only one.)</th>
<th>Transform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>esp-md5-hmac</td>
<td>ESP with the MD5 (HMAC variant) authentication algorithm. (No longer recommended).</td>
</tr>
<tr>
<td></td>
<td>esp-sha-hmac</td>
<td>ESP with the SHA (HMAC variant) authentication algorithm.</td>
</tr>
</tbody>
</table>

**Note**
- Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

### Tunnel and Transport Modes

Traffic can be encapsulated either in tunnel or transport mode if the traffic has the same IP addresses as IPsec peers. If the traffic to be protected has the same IP address as the IPsec peers and transport mode is specified, during negotiation the router will request transport mode but will accept either transport or tunnel mode. If tunnel mode is specified, the router will request tunnel mode and will accept only tunnel mode.

With tunnel mode, the entire original IP packet is protected (encrypted, authenticated, or both) and is encapsulated by IPsec headers and trailers (an encapsulation security protocol header and trailer, an authentication header, or both). Then, a new IP header is prefixed to the packet, specifying the IPsec endpoints as the source and destination. Tunnel mode can be used with any IP traffic. Tunnel mode must be used if IPsec is protecting traffic from hosts behind the IPsec peers. For example, tunnel mode is used with VPNs where hosts on one protected network send packets to hosts on a different protected network via a pair of...
IPsec peers. With VPNs, the IPsec peers "tunnel" the protected traffic between the peers while the hosts on their protected networks are the session endpoints.

With transport mode, only the payload (data) of the original IP packet is protected (encrypted, authenticated, or both). The payload is encapsulated by IPsec headers and trailers (an ESP header and trailer, an AH header, or both). The original IP headers remain intact and are not protected by IPsec. Use transport mode only when the IP traffic to be protected has IPsec peers as both the source and destination. For example, you could use transport mode to protect router management traffic. Specifying transport mode allows the router to negotiate with the remote peer whether to use transport or tunnel mode.

Cisco IOS Suite-B Support for IKE and IPsec Cryptographic Algorithms

Suite-B adds support for four user interface suites of cryptographic algorithms for use with IKE and IPsec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm.

Suite-B is not supported on the following hardware platforms:

- Cisco ASR1001
- ESP-5
- ESP-10
- ESP-20
- ESP-40

Suite-B has the following cryptographic algorithms:

- Suite-B-GCM-128-Provides ESP integrity protection, confidentiality, and IPsec encryption algorithms that use the 128-bit AES using Galois and Counter Mode (AES-GCM) described in RFC 4106. This suite should be used when ESP integrity protection and encryption are both needed.
- Suite-B-GCM-256-Provides ESP integrity protection and confidentiality using 256-bit AES-GCM described in RFC 4106. This suite should be used when ESP integrity protection and encryption are both needed.
- Suite-B-GMAC-128-Provides ESP integrity protection using 128-bit AES-Galois Message Authentication Code (GMAC) described in RFC 4543, but does not provide confidentiality. This suite should be used only when there is no need for ESP encryption.
- Suite-B-GMAC-256-Provides ESP integrity protection using 256-bit AES-GMAC described in RFC 4543, but does not provide confidentiality. This suite should be used only when there is no need for ESP encryption.

IPSec encryption algorithms use AES-GCM when encryption is required and AES-GMAC for message integrity without encryption.

IKE negotiation uses AES Cipher Block Chaining (CBC) mode to provide encryption and Secure Hash Algorithm (SHA)-2 family containing the SHA-256 and SHA-384 hash algorithms, as defined in RFC 4634, to provide the hash functionality. Diffie-Hellman using Elliptic Curves (ECP), as defined in RFC 4753, is used for key exchange and the Elliptic Curve Digital Signature Algorithm (ECDSA), as defined in RFC 4754, to provide authentication.
Suite-B Requirements

Suite-B imposes the following software crypto engine requirements for IKE and IPsec:

- HMAC-SHA256 and HMAC-SHA384 are used as pseudorandom functions; the integrity check within the IKE protocol is used. Optionally, HMAC-SHA512 can be used.
- Elliptic curve groups 19 (256-bit ECP curve) and 20 (384-bit ECP curve) are used as the Diffie-Hellman group in IKE. Optionally, group 21 (521-bit ECP curve) can be used.
- The Elliptic Curve Digital Signature Algorithm (ECDSA) algorithm (256-bit and 384-bit curves) is used for the signature operation within X.509 certificates.
- GCM (16 byte ICV) and GMAC is used for ESP (128-bit and 256-bit keys). Optionally, 192-bit keys can be used.
- Public Key Infrastructure (PKI) support for validation of X.509 certificates using ECDSA signatures must be used.
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS must be used.
- IKEV2 support for allowing the ECDSA signature (ECDSA-sig) as authentication method must be used.

Where to Find Suite-B Configuration Information

Suite-B configuration support is described in the following documents:

- For more information on SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration, see the Configuring Internet Key Exchange for IPsec VPNs feature module.
- For more information on configuring a transform for an integrity algorithm type, see the “Configuring the IKEv2 Proposal” section in the Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site feature module.
- For more information on configuring the ECDSA-sig to be the authentication method for IKEv2, see the “Configuring IKEv2 Profile (Basic)” section in the Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site feature module.
- For more information on configuring elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation, see the Configuring Internet Key Exchange for IPsec VPNs and Configuring Internet Key Exchange Version 2 and FlexVPN feature modules.

For more information on the Suite-B support for certificate enrollment for a PKI, see the Configuring Certificate Enrollment for a PKI feature module.
How to Configure IPsec VPNs

Creating Crypto Access Lists

SUMMARY STEPS

1. enable
2. configure terminal
3. Do one of the following:
   • access-list access-list-number {deny | permit} protocol source source-wildcard destination destination-wildcard [log]
   • ip access-list extended name
4. Repeat Step 3 for each crypto access list you want to create.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• access-list access-list-number {deny</td>
</tr>
<tr>
<td></td>
<td>• ip access-list extended name</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# access-list 100 permit ip 10.0.68.0 0.0.0.255 10.1.1.0 0.0.0.255</td>
</tr>
<tr>
<td></td>
<td>Tip Cisco recommends that you configure “mirror image” crypto access lists for use by IPsec and that you avoid using the any keyword.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Repeat Step 3 for each crypto access list you want to create.</td>
</tr>
</tbody>
</table>

**What to Do Next**

After at least one crypto access list is created, a transform set needs to be defined as described in the "Configuring Transform Sets for IKEv1 and IKEv2 Proposals" section.

Next the crypto access lists need to be associated to particular interfaces when you configure and apply crypto map sets to the interfaces. (Follow the instructions in the "Creating Crypto Map Sets" and "Applying Crypto Map Sets to Interfaces" sections).

**Configuring Transform Sets for IKEv1 and IKEv2 Proposals**

Perform this task to define a transform set that is to be used by the IPsec peers during IPsec security association negotiations with IKEv1 and IKEv2 proposals.

**Restrictions**

If you are specifying SEAL encryption, note the following restrictions:

- Your router and the other peer must not have a hardware IPsec encryption.
- Your router and the other peer must support IPsec.
- Your router and the other peer must support the k9 subsystem.
- SEAL encryption is available only on Cisco equipment. Therefore, interoperability is not possible.
- Unlike IKEv1, the authentication method and SA lifetime are not negotiable in IKEv2, and because of this, these parameters cannot be configured under the IKEv2 proposal.
Configuring Transform Sets for IKEv1

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]
4. mode [tunnel | transport]
5. end
6. clear crypto sa [peer {ip-address | peer-name} | sa map map-name | sa entry destination-address protocol spi]
7. show crypto ipsec transform-set [tag transform-set-name]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac</td>
</tr>
<tr>
<td></td>
<td>Defines a transform set and enters crypto transform configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• There are complex rules defining the entries that you can use for transform arguments. These rules are explained in the command description for the crypto ipsec transform-set command, and the table in &quot;About Transform Sets&quot; section provides a list of allowed transform combinations.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>mode [tunnel</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(cfg-crypto-tran)# mode transport</td>
</tr>
<tr>
<td></td>
<td>(Optional) Changes the mode associated with the transform set.</td>
</tr>
<tr>
<td></td>
<td>• The mode setting is applicable only to traffic whose source and destination addresses are the IPsec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>end</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(cfg-crypto-tran)# end</td>
</tr>
<tr>
<td></td>
<td>Exits crypto transform configuration mode and enters privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Purpose

**Step 6**

**Command or Action:** `clear crypto sa [peer {ip-address | peer-name} | sa map map-name | sa entry destination-address protocol spi]`

**Purpose:** (Optional) Clears existing IPsec security associations so that any changes to a transform set takes effect on subsequently established security associations.

- Manually established SAs are reestablished immediately.
  - Using the `clear crypto sa` command without parameters clears out the full SA database, which clears out active security sessions.
  - You may also specify the peer, map, or entry keywords to clear out only a subset of the SA database.

**Step 7**

**Command or Action:** `show crypto ipsec transform-set [tag transform-set-name]`

**Purpose:** (Optional) Displays the configured transform sets.

**Example:**

```
Device# clear crypto sa

Device# show crypto ipsec transform-set
```

### What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the *Creating Crypto Map Sets* section.

### Configuring Transform Sets for IKEv2

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto ikev2 proposal proposal-name`
4. `encryption transform1 [transform2] ...`
5. `integrity transform1 [transform2] ...`
6. `group transform1 [transform2] ...`
7. `end`
8. `show crypto ikev2 proposal`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto ikev2 proposal <strong>proposal-name</strong></td>
<td>Specifies the name of the proposal and enters crypto IKEv2 proposal configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# crypto ikev2 proposal proposal-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> encryption <strong>transform1</strong> [<strong>transform2</strong>] ...</td>
<td>(Optional) Specifies one or more transforms of the following encryption type:</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-ikev2-proposal)# encryption aes-cbc-128</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> integrity <strong>transform1</strong> [<strong>transform2</strong>] ...</td>
<td>(Optional) Specifies one or more transforms of the following integrity type:</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-ikev2-proposal)# integrity sha1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> group <strong>transform1</strong> [<strong>transform2</strong>] ...</td>
<td>(Optional) Specifies one or more transforms of the possible DH group type:</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-ikev2-proposal)# group 14</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>• 15—Specifies the 3072-bit DH group.</td>
<td></td>
</tr>
<tr>
<td>• 16—Specifies the 4096-bit DH group.</td>
<td></td>
</tr>
<tr>
<td>• 19—Specifies the 256-bit elliptic curve DH (ECDH) group.</td>
<td></td>
</tr>
<tr>
<td>• 20—Specifies the 384-bit ECDH group.</td>
<td></td>
</tr>
<tr>
<td>• 24—Specifies the 2048-bit DH/DSA group.</td>
<td></td>
</tr>
</tbody>
</table>

### Step 7
**end**

Example:
Device(config-ikev2-proposal)# end

Exits crypto IKEv2 proposal configuration mode and returns to privileged EXEC mode.

### Step 8
**show crypto ikev2 proposal**

(Optional) Displays the parameters for each IKEv2 proposal.

Example:
Device# show crypto ikev2 proposal

---

**Transform Sets for IKEv2 Examples**

The following examples show how to configure a proposal:

**IKEv2 Proposal with One Transform for Each Transform Type**

Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128
Device(config-ikev2-proposal)# integrity sha1
Device(config-ikev2-proposal)# group 14

**IKEv2 Proposal with Multiple Transforms for Each Transform Type**

Device(config)# crypto ikev2 proposal proposal-2
Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-192
Device(config-ikev2-proposal)# integrity sha2 sha256
Device(config-ikev2-proposal)# group 14 15

The IKEv2 proposal **proposal-2** translates to the following prioritized list of transform combinations:

- aes-cbc-128, sha1, 14
- aes-cbc-128, sha1, 15
- aes-cbc-128, sha256, 14
- aes-cbc-128, sha256, 15
- aes-cbc-192, sha1, 14
- aes-cbc-192, sha1, 15
- aes-cbc-192, sha256, 14
• aes-cbc-192, sha256, 15

IKEv2 Proposal with Multiple Transforms for Each Transform Type

crypto ikev2 proposal proposal-2
encryption aes-cbc-128 aes-cbc-192
integrity sha1 sha256
group 14 15

For a list of transform combinations, see Configuring Security for VPNs with IPsec.

IKEv2 Proposals on the Initiator and Responder

The proposal of the initiator is as follows:

Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-196
Device(config-ikev2-proposal)# integrity sha1 sha256
Device(config-ikev2-proposal)# group 14 16

The proposal of the responder is as follows:

Device(config)# crypto ikev2 proposal proposal-2
Device(config-ikev2-proposal)# encryption aes-cbc-196 aes-cbc-128
Device(config-ikev2-proposal)# integrity sha256 sha1
Device(config-ikev2-proposal)# group 16 14

In the scenario, the initiator’s choice of algorithms is preferred and the selected algorithms are as follows:

encryption aes-cbc-128
integrity sha1
group 14

What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the Creating Crypto Map Sets section.

Creating Crypto Map Sets

Creating Static Crypto Maps

When IKE is used to establish SAs, the IPsec peers can negotiate the settings they use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

Perform this task to create crypto map entries that use IKE to establish SAs. To create IPv6 crypto map entries, you must use the ipv6 keyword with the crypto map command. For IPv4 crypto maps, use the crypto map command without the ipv6 keyword.

Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto map [ipv6] map-name seq-num [ipsec-isakmp]
4. match address access-list-id
5. set peer {hostname | ip-address}
6. crypto ipsec security-association dummy {pps rate | seconds seconds}
7. set transform-set transform-set-name1 [transform-set-name2...transform-set-name6]
8. set security-association lifetime {seconds seconds | kilobytes kilobytes | kilobytes disable}
9. set security-association level per-host
10. set pfs [group1 | group14 | group15 | group16 | group19 | group2 | group20 | group24 | group5]
11. end
12. show crypto map [interface interface | tag map-name]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto map [ipv6] map-name seq-num [ipsec-isakmp]</td>
<td>Creates or modifies a crypto map entry, and enters crypto map configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# crypto map static-map 1 ipsec-isakmp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> match address access-list-id</td>
<td>Names an extended access list.</td>
</tr>
<tr>
<td>Example: Device(config-crypto-m)# match address vpn-tunnel</td>
<td>This access list determines the traffic that should be protected by IPsec and the traffic that should not be protected by IPsec security in the context of this crypto map entry.</td>
</tr>
<tr>
<td><strong>Step 5</strong> set peer {hostname</td>
<td>ip-address}</td>
</tr>
<tr>
<td>Example: Device(config-crypto-m)# set-peer 192.168.101.1</td>
<td>Repeat for multiple remote peers.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>`crypto ipsec security-association dummy {pps rate</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Device(config-crypto-m)# set security-association dummy seconds 5</td>
</tr>
<tr>
<td>7</td>
<td><code>set transform-set transform-set-name1 [transform-set-name2...transform-set-name6]</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Device(config-crypto-m)# set transform-set areset</td>
</tr>
<tr>
<td>8</td>
<td>`set security-association lifetime {seconds seconds</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Device (config-crypto-m)# set security-association lifetime seconds 2700</td>
</tr>
<tr>
<td>9</td>
<td><code>set security-association level per-host</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Device(config-crypto-m)# set security-association level per-host</td>
</tr>
<tr>
<td>10</td>
<td>`set pfs [group1</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Device(config-crypto-m)# set pfs group14</td>
</tr>
</tbody>
</table>

**Caution**

Use this command with care because multiple streams between given subnets can rapidly consume resources.

Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended).

Group 2 specifies the 1024-bit DH identifier. (No longer recommended).

Group 5 specifies the 1536-bit DH identifier. (No longer recommended)

Group 14 specifies the 2048-bit DH identifier.

Group 15 specifies the 3072-bit DH identifier.

Group 16 specifies the 4096-bit DH identifier.

Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier.

Group 20 specifies the 384-bit ECDH identifier.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Group 24 specifies the 2048-bit DH/DSA identifier</td>
<td></td>
</tr>
<tr>
<td>• By default, PFS is not requested. If no group is specified with this command, group 1 is used as the default.</td>
<td></td>
</tr>
<tr>
<td>Step 11</td>
<td>end</td>
</tr>
<tr>
<td>Example: Device(config-crypto-m)# end</td>
<td></td>
</tr>
<tr>
<td>Exits crypto map configuration mode and returns to privileged EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>Step 12</td>
<td>show crypto map [interface interface</td>
</tr>
<tr>
<td>Example: Device# show crypto map</td>
<td></td>
</tr>
<tr>
<td>Displays your crypto map configuration.</td>
<td></td>
</tr>
</tbody>
</table>

Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPsec traffic.

To clear IPsec SAs, use the clear crypto sa command with appropriate parameters. (Omitting all parameters clears out the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a static crypto map, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the “Applying Crypto Map Sets to Interfaces” section.

Creating Dynamic Crypto Maps

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPsec SAs can be established. A dynamic crypto map entry that does not specify an access list is ignored during traffic filtering. A dynamic crypto map entry with an empty access list causes traffic to be dropped. If there is only one dynamic crypto map entry in the crypto map set, it must specify the acceptable transform sets.

Perform this task to create dynamic crypto map entries that use IKE to establish the SAs.

Note
IPv6 addresses are not supported on dynamic crypto maps.
Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. crypto dynamic-map dynamic-map-name dynamic-seq-num
4. set transform-set transform-set-name1 [transform-set-name2...transform-set-name6]
5. match address access-list-id
6. set peer \{hostname | ip-address\}
7. set security-association lifetime \{seconds seconds | kilobytes kilobytes | kilobytes disable\}
8. set pfs \{group1 | group14 | group15 | group16 | group19 | group2 | group20 | group24 | group5\}
9. exit
10. exit
11. show crypto dynamic-map [tag map-name]
12. configure terminal
13. crypto map map-name seq-num ipsec-isakmp dynamic dynamic-map-name [discover]
14. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto dynamic-map</td>
<td>Creates a dynamic crypto map entry and enters crypto map configuration</td>
</tr>
<tr>
<td>dynamic-map-name</td>
<td>mode.</td>
</tr>
<tr>
<td>dynamic-seq-num</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# crypto dynamic-map</td>
<td></td>
</tr>
<tr>
<td>test-map 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> set transform-set</td>
<td>Specifies the transform sets allowed for the crypto map entry.</td>
</tr>
<tr>
<td>transform-set-name1 [transform-</td>
<td></td>
</tr>
<tr>
<td>set-name2...transform-set-name6]</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Device(config-crypto-m)# set transform-set aesset</code></td>
<td>• List multiple transform sets in the order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.</td>
</tr>
<tr>
<td><strong>Step 5</strong> match address <em>access-list-id</em></td>
<td>(Optional) Specifies the list number or name of an extended access list.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Device(config-crypto-m)# match address 101</code></td>
<td>• This access list determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec security in the context of this crypto map entry.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Although access lists are optional for dynamic crypto maps, they are highly recommended.</td>
</tr>
<tr>
<td></td>
<td>• If an access list is configured, the data flow identity proposed by the IPsec peer must fall within a <em>permit</em> statement for this crypto access list.</td>
</tr>
<tr>
<td></td>
<td>• If an access list is not configured, the device accepts any data flow identity proposed by the IPsec peer. However, if an access list is configured but the specified access list does not exist or is empty, the device drops all packets. This is similar to static crypto maps, which require access lists to be specified.</td>
</tr>
<tr>
<td></td>
<td>• Care must be taken if the <em>any</em> keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation.</td>
</tr>
<tr>
<td></td>
<td>• You must configure a match address; otherwise, the behavior is not secure, and you cannot enable TED because packets are sent in the clear (unencrypted.)</td>
</tr>
<tr>
<td><strong>Step 6</strong> set peer *{hostname</td>
<td>ip-address}*</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Device(config-crypto-m)# set peer 192.168.101.1</code></td>
<td><strong>Note</strong> This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.</td>
</tr>
<tr>
<td><strong>Step 7</strong> set security-association lifetime *{seconds</td>
<td>seconds</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Device(config-crypto-m)# set security-association lifetime seconds 7200</code></td>
<td><strong>Note</strong> To minimize the possibility of packet loss when rekeying in high bandwidth environments, you can disable the rekey request triggered by a volume lifetime expiry.</td>
</tr>
<tr>
<td><strong>Step 8</strong> set pfs *{group1</td>
<td>group14</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Example:** Device(config-crypto-m)# set pfs group14 | • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended).  
• Group 2 specifies the 1024-bit DH identifier. (No longer recommended).  
• Group 5 specifies the 1536-bit DH identifier. (No longer recommended)  
• Group 14 specifies the 2048-bit DH identifier.  
• Group 15 specifies the 3072-bit DH identifier.  
• Group 16 specifies the 4096-bit DH identifier.  
• Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier.  
• Group 20 specifies the 384-bit ECDH identifier.  
• Group 24 specifies the 2048-bit DH/DSA identifier  
• By default, PFS is not requested. If no group is specified with this command, **group1** is used as the default. |

**Step 9** exit  
**Example:** Device(config-crypto-m)# exit  
Exits crypto map configuration mode and returns to global configuration mode.

**Step 10** exit  
**Example:** Device(config)# exit  
Exits global configuration mode.

**Step 11** show crypto dynamic-map [tag map-name]  
**Example:** Device# show crypto dynamic-map  
(Optional) Displays information about dynamic crypto maps.

**Step 12** configure terminal  
**Example:** Device# configure terminal  
Enters global configuration mode.

**Step 13** crypto map map-name seq-num  
**Example:** Device(config)# crypto map static-map  
Optional) Adds a dynamic crypto map to a crypto map set.  
• You should set the crypto map entries referencing dynamic maps to the lowest priority entries in a crypto map set.  

**Note** You must enter the **discover** keyword to enable TED.
### Command or Action

<table>
<thead>
<tr>
<th>Step 14</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exit</td>
<td>Exits global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(config)# exit
```

### Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the entire SA database must be reserved for large-scale changes, or when the router is processing minimal IPsec traffic.

To clear IPsec SAs, use the `clear crypto sa` command with appropriate parameters. (Omitting all parameters clears the full SA database, which clears active security sessions.)

### What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the "Applying Crypto Map Sets to Interfaces" section.

### Creating Crypto Map Entries to Establish Manual SAs

Perform this task to create crypto map entries to establish manual SAs (that is, when IKE is not used to establish the SAs). To create IPv6 crypto maps entries, you must use the `ipv6` keyword with the `crypto map` command. For IPv4 crypto maps, use the `crypto map` command without the `ipv6` keyword.
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto map [ipv6] map-name seq-num [ipsec-manual]
4. match address access-list-id
5. set peer {hostname | ip-address}
6. set transform-set transform-set-name
7. Do one of the following:
   - set session-key inbound ah spi hex-key-string
   - set session-key outbound ah spi hex-key-string
8. Do one of the following:
   - set session-key inbound esp spi cipher hex-key-string [authenticator hex-key-string]
   - set session-key outbound esp spi cipher hex-key-string [authenticator hex-key-string]
9. exit
10. exit
11. show crypto map [interface interface | tag map-name]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.</td>
</tr>
<tr>
<td>crypto map [ipv6]</td>
<td>Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.</td>
</tr>
<tr>
<td>map-name seq-num</td>
<td>Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.</td>
</tr>
<tr>
<td>[ipsec-manual]</td>
<td>Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# crypto map mymap 10 ipsec-manual</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Names an IPsec access list that determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec in the context of this crypto map entry.</td>
</tr>
<tr>
<td>match address</td>
<td>Names an IPsec access list that determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec in the context of this crypto map entry.</td>
</tr>
<tr>
<td>access-list-id</td>
<td>Names an IPsec access list that determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec in the context of this crypto map entry.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-crypto-m)# match address 102</td>
</tr>
</tbody>
</table>
### Creating Crypto Map Sets

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• The access list can specify only one <strong>permit</strong> entry when IKE is not used.</td>
</tr>
</tbody>
</table>

#### Step 5

**set peer** `{hostname | ip-address}`

Specifies the remote IPsec peer. This is the peer to which IPsec protected traffic should be forwarded.

- Only one peer can be specified when IKE is not used.

**Example:**

```
Device(config-crypto-m)# set peer 10.0.0.5
```

#### Step 6

**set transform-set** `transform-set-name`

Specifies which transform set should be used.

- This must be the same transform set that is specified in the remote peer’s corresponding crypto map entry.

**Note**

Only one transform set can be specified when IKE is not used.

**Example:**

```
Device(config-crypto-m)# set transform-set somewset
```

#### Step 7

Do one of the following:

- **set session-key inbound ah** `spi hex-key-string`
- **set session-key outbound ah** `spi hex-key-string`

Sets the AH security parameter indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the AH protocol.

- This manually specifies the AH security association to be used with protected traffic.

**Example:**

```
Device(config-crypto-m)# set session-key inbound ah 256 98765432109876549876543210987654
```

**Example:**

```
Device(config-crypto-m)# set session-key outbound ah 256 fedcbadcfebcfdecbedcbeadce
```

### Step 8

Do one of the following:

- **set session-key inbound esp** `spi cipher hex-key-string [authenticator hex-key-string]`
- **set session-key outbound esp** `spi cipher hex-key-string [authenticator hex-key-string]`

Sets the Encapsulating Security Payload (ESP) Security Parameter Indexes (SPI) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol.

Or

Specifies the cipher keys if the transform set includes an ESP cipher algorithm. Specifies the authenticator keys if the transform set includes an ESP authenticator algorithm.

- This manually specifies the ESP security association to be used with protected traffic.

**Example:**

```
Device(config-crypto-m)# set session-key inbound esp 256 cipher 0123456789012345
```

**Example:**

```
Device(config-crypto-m)# set session-key outbound esp 256 cipher abedefabcdefab56c
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-crypto-m)# exit</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits crypto map configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# exit</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>show crypto map [interface interface</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# show crypto map</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Displays your crypto map configuration.</td>
</tr>
</tbody>
</table>

**Troubleshooting Tips**

For manually established SAs, you must clear and reinitialize the SAs for the changes to take effect. To clear IPsec SAs, use the clear crypto sa command with appropriate parameters. (Omitting all parameters clears the entire SA database, which clears active security sessions.)

**What to Do Next**

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the “Applying Crypto Map Sets to Interfaces” section.

**Applying Crypto Map Sets to Interfaces**

You must apply a crypto map set to each interface through which IPsec traffic flows. Applying the crypto map set to an interface instructs the device to evaluate the interface’s traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by the crypto map.

Perform this task to apply a crypto map to an interface.
### SUMMARY STEPS

1. enable
2. configure terminal
3. interface type/number
4. crypto map map-name
5. exit
6. crypto map map-name local-address interface-id
7. exit
8. show crypto map [interface interface]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong>&lt;br&gt;Example: Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>configure terminal</strong>&lt;br&gt;Example: Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>interface type/number</strong>&lt;br&gt;Example: Device(config)# interface FastEthernet 0/0</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>crypto map map-name</strong>&lt;br&gt;Example: Device(config-if)# crypto map mymap</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>exit</strong>&lt;br&gt;Example: Device(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>crypto map map-name local-address interface-id</strong>&lt;br&gt;Example: Device(config)# crypto map mymap local-address loopback0</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>exit</strong>&lt;br&gt;Example: Device(config)# exit</td>
</tr>
</tbody>
</table>
### Command or Action

**Step 8**

- `show crypto map [interface interface]`

**Example:**

```
Device# show crypto map
```

### Purpose

(Optional) Displays your crypto map configuration

---

## Configuration Examples for IPsec VPN

### Example: Configuring AES-Based Static Crypto Map

This example shows how a static crypto map is configured and how an AES is defined as the encryption method:

```
crypto isakmp policy 10
  encryption aes 256
  authentication pre-share
  group 14
  lifetime 180
crypto isakmp key cisco123 address 10.0.110.1
  !
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
  mode transport
  !
crypto map aesmap 10 ipsec-isakmp
  set peer 10.0.110.1
  set transform-set aesset
  match address 120
  !
  !
voice call carrier capacity active
  !
mta receive maximum-recipients 0
  !
interface FastEthernet0/0
  ip address 10.0.110.2 255.255.255.0
  ip nat outside
  no ip route-cache
  no ip mrout-cache
  duplex auto
  speed auto
crypto map aesmap
interface Serial0/0
  no ip address
  shutdown
interface FastEthernet0/1
  ip address 10.0.110.1 255.255.255.0
  ip nat inside
  no ip route-cache
  no ip mrout-cache
  duplex auto
  speed auto
```
ip nat inside source list 110 interface FastEthernet0/0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 10.5.1.1
ip route 10.0.110.0 255.255.255.0 FastEthernet0/0
ip route 172.18.124.0 255.255.255.0 10.5.1.1
ip route 172.18.125.3 255.255.255.255 10.5.1.1
ip http server
!
access-list 110 deny ip 10.0.110.0 0.0.0.255 10.0.110.0 0.0.0.255
access-list 110 permit ip 10.0.110.0 0.0.0.255 any
access-list 120 permit ip 10.0.110.0 0.0.0.255 10.0.110.0 0.0.0.255
!

Additional References for Configuring Security for VPNs with IPsec

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
</tbody>
</table>
| IKE, IPsec, and PKI configuration commands | • Cisco IOS Security Command Reference Commands A to C  
• Cisco IOS Security Command Reference Commands D to L  
• Cisco IOS Security Command Reference Commands M to R  
• Cisco IOS Security Command Reference Commands S to Z |
<p>| IKE configuration | Configuring Internet Key Exchange for IPsec VPNs |
| Suite-B SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration | Configuring Internet Key Exchange for IPsec VPNs |
| Suite-B Integrity algorithm type transform configuration | Configuring Internet Key Exchange Version 2 (IKEv2) |
| Suite-B Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method configuration for IKEv2 | Configuring Internet Key Exchange Version 2 (IKEv2) |</p>
<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suite-B Elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation</td>
<td>• Configuring Internet Key Exchange for IPsec VPNs</td>
</tr>
<tr>
<td></td>
<td>• Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site</td>
</tr>
<tr>
<td>Suite-B support for certificate enrollment for a PKI</td>
<td>Configuring Certificate Enrollment for a PKI</td>
</tr>
<tr>
<td>Set Up VPN between Cisco ASR 100 Series and Google Cloud Platform</td>
<td>Set Up VPN between Cisco ASR 100 Series and Google Cloud Platform</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

**Standards**

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>—</td>
</tr>
</tbody>
</table>

**MIBs**

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>• CISCO-IPSEC-FLOW-MONITOR-MIB</td>
<td>To locate and download MIBs for selected platforms,</td>
</tr>
<tr>
<td>• CISCO-IPSEC-MIB</td>
<td>Cisco IOS software releases, and feature sets, use</td>
</tr>
<tr>
<td>• CISCO-IPSEC-POLICY-MAP-MIB</td>
<td>Cisco MIB Locator found at the following URL:</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</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>RFCs</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>RFC 2406</td>
<td><em>IP Encapsulating Security Payload (ESP)</em></td>
</tr>
<tr>
<td>RFC 2407</td>
<td><em>The Internet IP Security Domain of Interpretation for ISAKMP</em></td>
</tr>
<tr>
<td>RFC 2408</td>
<td><em>Internet Security Association and Key Management Protocol (ISAKMP)</em></td>
</tr>
</tbody>
</table>

**Technical Assistance**

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

**Feature Information for Configuring Security for VPNs with IPsec**

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.
Table 2: Feature Information for Configuring Security for IPsec VPNs

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Encryption Standard</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature adds support for the new encryption standard AES, which is a privacy transform for IPsec and IKE and has been developed to replace DES. The following commands were modified by this feature: crypto ipsec transform-set, encryption (IKE policy), show crypto ipsec transform-set, show crypto isakmp policy.</td>
</tr>
<tr>
<td>IKE Shared Secret Using AAA Server</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The IKE Shared Secret Using AAA Server feature enables key lookup from a AAA server. In Cisco IOS XE Release 2.1, this feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Suite-B Support in IOS SW Crypto</td>
<td>Cisco IOS XE Release 3.7S</td>
<td>Suite-B adds support for four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. The following command was modified by this feature: crypto ipsec transform-set.</td>
</tr>
</tbody>
</table>

Glossary

anti-replay—Security service where the receiver can reject old or duplicate packets to protect itself against replay attacks. IPsec provides this optional service by use of a sequence number combined with the use of data authentication. Cisco IOS XE IPsec provides this service whenever it provides the data authentication service, except for manually established SAs (that is, SAs established by configuration and not by IKE).

data authentication—Verification of the integrity and origin of the data. Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).
**data confidentiality**—Security service in which the protected data cannot be observed.

**data flow**—Grouping of traffic, identified by a combination of source address or mask, destination address or mask, IP next protocol field, and source and destination ports, where the protocol and port fields can have the values of any. IPsec protection is applied to data flows.

**IKE**—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPsec) that require keys. Before any IPsec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

**IPsec**—IP Security. 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. IPsec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

**peer**—In the context of this module, a “peer” is a router or other device that participates in IPsec.

**PFS**—perfect forward secrecy. Cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

**SA**—security association. Description of how two or more entities use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. The transform and the shared secret keys are used for protecting the traffic.

**SPI**—security parameter index. A number which, together with a destination IP address and security protocol, uniquely identifies a particular security association. Without IKE, the SPI is manually specified for each security association.

**transform**—List of operations performed on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

**tunnel**—In the context of this module, “tunnel” is a secure communication path between two peers, such as two routers. It does not refer to using IPsec in tunnel mode.
Chapter 3

IPsec Virtual Tunnel Interfaces

Last Published Date: March 28, 2014

IPsec virtual tunnel interfaces (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify the configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.

---

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

---

- Finding Feature Information, page 39
- Restrictions for IPsec Virtual Tunnel Interfaces, page 40
- Information About IPsec Virtual Tunnel Interfaces, page 41
- How to Configure IPsec Virtual Tunnel Interfaces, page 48
- Configuration Examples for IPsec Virtual Tunnel Interfaces, page 62
- Additional References for IPsec Virtual Tunnel Interface, page 74
- Feature Information for IPsec Virtual Tunnel Interfaces, page 75

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Restrictions for IPsec Virtual Tunnel Interfaces

IPsec Transform Set
The IPsec transform set must be configured in tunnel mode only.

IKE Security Association
The Internet Key Exchange (IKE) security association (SA) is bound to the VTI. Therefore, the same IKE SA cannot be used for a crypto map.

IPsec SA Traffic Selectors
Static VTIs (SVTIs) support only a single IPsec SA that is attached to the VTI interface. The traffic selector for the IPsec SA is always "IP any any."
A dynamic VTI (DVTIs) is also a point-to-point interface that can support multiple IPsec SAs. The DVTI can accept the multiple IPsec selectors that are proposed by the initiator.

IPv4 and IPv6 Packets
This feature supports SVTIs that are configured to encapsulate IPv4 packets or IPv6 packets, but IPv4 packets cannot carry IPv6 packets and IPv6 packets cannot carry IPv4 packets.

Proxy
SVTIs support only the "IP any any" proxy.
DVTIs support multiple proxies, but DVTIs do not allow mixing "any any" proxies with non-"any any" proxies.
DVTIs permit only one type of proxy at a time, either a single "any any" proxy or multiple "no any any" proxies.

Stateful Failover
IPsec stateful failover is not supported with IPsec VTIs.

Static VTIs Versus GRE Tunnels
The IPsec VTI is limited to the IP unicast and multicast traffic only, as opposed to Generic Routing Encapsulation (GRE) tunnels, which have a wider application for IPsec implementation.

Single Template Model
In the single template model, the VPN routing and forwarding (VRF) is configured in the ISAKMP profile. In this model, each virtual access that is created belongs to the internal VRF (IVRF) specified in the ISAKMP profile. But because the IP address of the virtual access is derived from the interface to which the virtual access is unnumbered to, the IP address of the interface will not be available in the virtual access routing table. This happens because the unnumbered interface does not belong to the IVRF routing table of the virtual access. In such cases, a ping to the virtual access IP address fails.

Tunnel Protection
Do not configure the shared keyword when using the tunnel mode ipsec ipv4 command for IPsec IPv4 mode.
Traceroute

The traceroute function with crypto offload on VTIs is not supported.

Virtual Template Lock

Effective with CSCtt26236, virtual template lock allows you to modify or delete a virtual template of type tunnel only when the virtual template is not associated with any cloned virtual access interfaces. The virtual template lock prevents dynamic command updates from virtual templates to the cloned virtual access interfaces, which can cause instability in some scenarios.

If you try to modify or delete an active virtual template of type tunnel, the following error message appears:

Device(config)# interface virtual-template 1
% Virtual-template config is locked, active vaccess present

Although the virtual template cannot be modified when the virtual template is associated with a virtual access interface, perform the following steps to modify an existing virtual template configuration:

1. Configure a new virtual template interface. For more information, see "Configuring Dynamic IPsec Virtual Tunnel Interfaces."

2. Associate the new virtual template to the IKEv2 profile. For more information, see the Configuring IKEv2 Profile (Basic) module.

3. Clear the active sessions using the clear crypto session command or wait for session termination.

The new session will use the new virtual template.

VRF-Aware IPsec Configuration

VPN routing and forwarding (VRF) must not be configured in the Internet Security Association and Key Management Protocol (ISAKMP) profile in VRF-aware IPsec configurations with either SVTIs or DVTIs. Instead, the VRF must be configured on the tunnel interface for SVTIs. For DVTIs, you must apply the VRF to the virtual template using the ip vrf forwarding command.

You must include the VRF in the local-address command when using the local address with VRF in the ISAKMP profile and keyring.

IPsec Mixed mode support for VTspag-rel318ls

Mixed mode is not supported with tunnel mode auto.

Mixed mode is not supported with tunnel protection ipsec [shared].

Information About IPsec Virtual Tunnel Interfaces

The use of IPsec VTIs can simplify the configuration process when you need to provide protection for remote access and it provides an alternative to using generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP) tunnels for encapsulation. A benefit of using IPsec VTIs is that the configuration does not require static mapping of IPsec sessions to a physical interface. The IPsec tunnel endpoint is associated with an actual (virtual) interface. Because there is a routable interface at the tunnel endpoint, many common interface capabilities can be applied to the IPsec tunnel.

The IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface, such as in the case of multiple paths. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using IP routing to
forward the traffic to the tunnel interface simplifies the IPsec VPN configuration compared to the more complex process of using access control lists (ACLs) with the crypto map in native IPsec configurations. Because DVTIs function like any other real interface you can apply quality of service (QoS), firewall, and other security services as soon as the tunnel is active.

The following sections provide details about the IPSec VTI:

**Benefits of Using IPsec Virtual Tunnel Interfaces**

IPsec VTIs allow you to configure a virtual interface to which you can apply features. Features for clear-text packets are configured on the VTI. Features for encrypted packets are applied on the physical outside interface. When IPsec VTIs are used, you can separate the application of features such as Network Address Translation (NAT), ACLs, and QoS and apply them to clear-text, or encrypted text, or both. When crypto maps are used, there is no simple way to apply extra features to the IPsec tunnel.

There are two types of VTI interfaces: static VTIs (SVTIs) and dynamic VTIs (DVTIs).

**Static Virtual Tunnel Interfaces**

SVTI configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites.

The advantage of using SVTIs as opposed to crypto map configurations is that users can enable dynamic routing protocols on the tunnel interface without the extra 24 bytes required for GRE headers, thus reducing the bandwidth for sending encrypted data.

Additionally, multiple Cisco IOS software features can be configured directly on the tunnel interface and on the physical egress interface of the tunnel interface. This direct configuration allows users to have solid control on the application of the features in the pre- or post-encryption path.

The figure below illustrates how a SVTI is used.

*Figure 1: IPsec SVTI*

The IPsec VTI supports native IPsec tunneling and exhibits most of the properties of a physical interface.
When configuring IPsec SVTI with high availability (HA), the standby router reload does not affect the existing security associations.

Dynamic Virtual Tunnel Interfaces

DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels.

You can configure DVTIs with IKEv1 or IKEv2. The legacy crypto map based configuration supports DVTIs with IKEv1 only. A DVTI configuration with IKEv2 is supported only in FlexVPN.

DVTIs can be used for both the server and the remote configuration. The tunnels provide an on-demand separate virtual access interface for each VPN session. The configuration of the virtual access interfaces is cloned from a virtual template configuration, which includes the IPsec configuration and any Cisco IOS software feature configured on the virtual template interface, such as QoS, NetFlow, or ACLs.

DVTIs function like any other real interface, so you can apply QoS, firewall, or other security services as soon as the tunnel is active. QoS features can be used to improve the performance of various applications across the network. Any combination of QoS features offered in Cisco IOS software can be used to support voice, video, or data applications.

DVTIs provide efficiency in the use of IP addresses and provide secure connectivity. DVTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using an extended authentication (Xauth) User or Unity group, or can be derived from a certificate. DVTIs are standards based, so interoperability in a multiple-vendor environment is supported. IPsec DVTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco Architecture for Voice, Video, and Integrated Data (AVVID) to deliver converged voice, video, and data over IP networks. The DVTI simplifies VPN routing and forwarding-(VRF-) aware IPsec deployment. The VRF is configured on the interface.

A DVTI requires minimal configuration on the router. A single virtual template can be configured and cloned. The DVTI creates an interface for IPsec sessions and uses the virtual template infrastructure for dynamic instantiation and management of dynamic IPsec VTIs. The virtual template infrastructure is extended to create dynamic virtual-access tunnel interfaces. DVTIs are used in hub-and-spoke configurations. A single DVTI can support several static VTIs.
The figure below illustrates the DVTI authentication path.

Figure 2: Dynamic IPsec VTI

The authentication shown in the figure above follows this path:

1 User 1 calls the router.
2 Router 1 authenticates User 1.
3 IPsec clones the virtual access interface from the virtual template interface.

Traffic Encryption with the IPsec Virtual Tunnel Interface

When an IPsec VTI is configured, encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static routing can be used to route traffic to the SVTI. DVTI uses reverse route injection to further simplify the routing configurations. Using IP routing to forward the traffic to encryption simplifies the IPsec VPN configuration because the use of ACLs with a crypto map in native IPsec configurations is not required. The IPsec virtual tunnel also allows you to encrypt multicast traffic with IPsec.
IPsec packet flow into the IPSec tunnel is illustrated in the figure below.

**Figure 3: Packet Flow into the IPsec Tunnel**

1. Clear-text IP packets enter the router.
2. Packets are routed to virtual interface.
3. Encrypted packets are forwarded to the forwarding engine.
4. Encrypted packets are passed out of the physical outside interface.

After packets arrive on the inside interface, the forwarding engine switches the packets to the VTI, where they are encrypted. The encrypted packets are handed back to the forwarding engine, where they are switched through the outside interface.

The figure below shows the packet flow out of the IPsec tunnel.

**Figure 4: Packet Flow out of the IPsec Tunnel**

1. IPsec encrypted packets enter the router.
2. Forwarding engine determines it is a packet for us and sends it to IPsec for decryption.
3. IPsec decrypts the packets and associates to the VTI based on the SA information.
4. Clear text packets are passed through physical interface.
Multi-SA Support for Dynamic Virtual Interfaces

DVTI supports multiple IPsec SAs. The DVTI can accept multiple IPsec selectors that are proposed by the initiator.

The DVTIs allow per-peer features to be applied on a dedicated interface. You can order features in such way that all features that are applied on the virtual access interfaces are applied before applying crypto. Additionally, all the features that are applied on the physical interfaces are applied after applying crypto. Clean routing is available across all VRFs so that there are no traffic leaks from one VRF to another before encrypting.

Multi-SA VTIs ensure interoperability with third-party devices and provide a flexible, clean, and modular feature-set.

Multi-SA VTIs enable a clean Cisco IOS XE infrastructure, even when the Cisco IOS XE software interoperates with the third-party devices that only implement crypto maps.

VRF and Scalability of Baseline Configuration:

Virtual access instances inherit the Inside-VRF (IVRF) from the template configuration. Users must configure several templates to enforce an appropriate IVRF for each customer. The number of templates must be equal to the number of customers connecting to the headend. Such a configuration is cumbersome and undesirable and also affects performance because each template declaration consumes one Interface Descriptor Block (IDB).

This complication can be avoided by allowing the IKE profile to override the virtual access VRF with the VRF configured on the IKE profile. A better solution is to allow the IKE profile to override the virtual access VRF using AAA, but this method is supported only for IKEv2.

The VRF configured in the ISAKMP profile is applied to the virtual access first. Then the configuration from virtual template is applied to the virtual access. If your virtual template contains `ip vrf forwarding` command configuration, the VRF from the template overrides the VRF from the ISAKMP profile.

Rules for Initial Configuration of a VRF:

The following rules must be applied during the initial configuration of VRF:

- If you configure IVRF in the IKE profile without configuring it in the virtual template, then you must apply the VRF from the IKE profile on each virtual access derived from this IKE profile.
- If you configure VRF in an IKE profile and virtual template, then the virtual template IVRF gets precedence.

Rules for Changing the VRF:

If you change the VRF configured in an IKE profile, all the IKE SAs, IPsec SAs, and the virtual access identifier derived from this profile will get deleted. The same rule applies when the VRF is configured on the IKE profile for the first time.

Dynamic Virtual Tunnel Interface Life Cycle

IPsec profiles define the policy for DVTIs. The dynamic interface is created at the end of IKE Phase 1 and IKE Phase 1.5. The interface is deleted when the IPsec session to the peer is closed. The IPsec session is closed when both IKE and IPsec SAs to the peer are deleted.
Routing with IPsec Virtual Tunnel Interfaces

Because VTIs are routable interfaces, routing plays an important role in the encryption process. Traffic is encrypted only if it is forwarded out of the VTI, and traffic arriving on the VTI is decrypted and routed accordingly. VTIs allow you to establish an encryption tunnel using a real interface as the tunnel endpoint. You can route to the interface or apply services such as QoS, firewalls, network address translation (NAT), and NetFlow statistics as you would to any other interface. You can monitor the interface and route to it, and the interface has an advantage over crypto maps because it is a real interface and provides benefits similar to other Cisco IOS interface.

FlexVPN Mixed Mode Support

The FlexVPN Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.

This feature is only supported for Remote Access VPN with IKEv2 and Dynamic VTI.

The FlexVPN Mixed Mode feature provides support for carrying IPv6 traffic over IPsec IPv4 transport from Cisco IOS XE Everest 16.4.1.

Auto Tunnel Mode Support in IPsec

When configuring a VPN headend in a multiple vendor scenario, you must be aware of the technical details of the peer or responder. For example, some devices may use IPsec tunnels while others may use generic routing encapsulation (GRE) or IPsec tunnel, and sometimes, a tunnel may be IPv4 or IPv6. In the last case, you must configure an Internet Key Exchange (IKE) profile and a virtual template.

The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder’s details. This feature automatically applies the tunnelling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. This feature is useful on dual stack hubs aggregating multivendor remote access, such as Cisco AnyConnect VPN Client, Microsoft Windows7 Client, and so on.

Note

The Tunnel Mode Auto Selection feature eases the configuration for a responder only. The tunnel must be statically configured for an initiator.

IPSec Mixed Mode Support for VTI

The IPSec Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.

This feature is supported for SVTI as well as DVTI and IKEv1 as well as IKEv2.
How to Configure IPsec Virtual Tunnel Interfaces

Configuring Static IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto IPsec profile profile-name
4. set transform-set transform-set-name [transform-set-name2...transform-set-name6]
5. exit
6. interface type number
7. ip address address mask
8. tunnel mode ipsec ipv4
9. tunnel source interface-type interface-number
10. tunnel destination ip-address
11. tunnel protection IPsec profile profile-name
12. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto IPsec profile profile-name</td>
<td>Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# crypto IPsec profile PROF</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies which transform sets can be used with the crypto map entry.</td>
</tr>
<tr>
<td><code>set transform-set transform-set-name [transform-set-name2...transform-set-name6]</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(ipsec-profile)# set transform-set tset</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits IPsec profile configuration mode, and enters global configuration mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(ipsec-profile)# exit</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Specifies the interface on which the tunnel will be configured and enters interface configuration mode.</td>
</tr>
<tr>
<td><code>interface type number</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# interface tunnel 0</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Specifies the IP address and mask.</td>
</tr>
<tr>
<td><code>ip address address mask</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# ip address 10.1.1.1 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Defines the mode for the tunnel.</td>
</tr>
<tr>
<td><code>tunnel mode ipsec ipv4</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel mode ipsec ipv4</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Specifies the tunnel source as a loopback interface.</td>
</tr>
<tr>
<td><code>tunnel source interface-type interface-number</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel source loopback 0</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Identifies the IP address of the tunnel destination.</td>
</tr>
<tr>
<td><code>tunnel destination ip-address</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel destination 172.16.1.1</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Associates a tunnel interface with an IPsec profile.</td>
</tr>
<tr>
<td><code>tunnel protection IPsec profile profile-name</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel protection IPsec profile PROF</td>
</tr>
</tbody>
</table>
## Configuring BGP over IPsec Virtual Tunnel Interfaces

Perform this task to optionally configure BGP over the virtual tunnel interfaces of two routers.

### Before You Begin

Perform steps in Configuring Static IPsec Virtual Tunnel Interfaces, on page 48.

### SUMMARY STEPS

1. `router bgp autonomous-system-number`
2. `neighbor ip-address remote-as autonomous-system-number`
3. `network network-ip-address mask subnet-mask`
4. `exit`
5. Enter the following commands on the second router.
6. `router bgp autonomous-system-number`
7. `neighbor ip-address remote-as autonomous-system-number`
8. `network network-ip-address mask subnet-mask`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Enters router configuration mode and creates a BGP routing process. <em>autonomous-system-number</em> — Number of an autonomous system that identifies the router to other BGP routers and tags the routing information that is passed along. Number in the range from 1 to 65535. In the example, the first router in this procedure is identified as &quot;65510&quot;.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Device(config)# router bgp 65510</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>neighbor ip-address remote-as autonomous-system-number</code></td>
<td><em>ip-address</em> — IP address of the adjacent router’s tunnel interface. <em>autonomous-system-number</em> — Number of an autonomous system that identifies the router of the second router. Number in the range from 1 to 65535.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Device(config-router)# neighbor 10.1.1.2 remote-as 65511</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>network network-ip-address mask subnet-mask</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router)# network 2.2.2.0 mask 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>network-ip-address</strong>—IP address of the network advertised in BGP. For example, the IP address of a loopback interface.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>subnet-mask</strong>—subnet mask of the network advertised in BGP.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The BGP network command and mask <em>must</em> exactly match a route that is already in the routing table for it to be brought into BGP and advertised to BGP neighbors. This is different from EIGRP, OSPF where the network statement just has to &quot;cover&quot; an interface network and it will pick up the network with mask from the interface.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>exit</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enter the following commands on the second router.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>router bgp autonomous-system-number</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# router bgp 65511</td>
<td></td>
</tr>
<tr>
<td><strong>autonomous-system-number</strong>—Number of an autonomous system that identifies the router to other BGP routers and tags the routing information that is passed along. Number in the range from 1 to 65535.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In the example, the second router in this procedure is identified as &quot;65511&quot;.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>neighbor ip-address remote-as autonomous-system-number</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router)# neighbor 10.1.1.1 remote-as 65510</td>
<td></td>
</tr>
<tr>
<td><strong>ip-address</strong>—IP address of the adjacent router's tunnel interface.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>network network-ip-address mask subnet-mask</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router)# network 1.1.1.0 mask 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>network-ip-address</strong>—IP address of the network advertised in BGP. For example, the IP address of a loopback interface.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>subnet-mask</strong>—subnet mask of the network advertised in BGP.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Use the exact network IP address and subnet mask.</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Dynamic IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto ipsec profile profile-name
4. set transform-set transform-set-name [transform-set-name2...transform-set-name6]
5. exit
6. interface virtual-template number type tunnel
7. tunnel mode ipsec ipv4
8. tunnel protection IPsec profile profile-name
9. exit
10. crypto isakamp profile profile-name
11. match identity address ip-address mask
12. virtual template template-number
13. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto ipsec profile profile-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# crypto ipsec profile PROF</td>
</tr>
<tr>
<td></td>
<td>Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>set transform-set transform-set-name [transform-set-name2...transform-set-name6]</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(ipsec-profile)# set transform-set tset</td>
</tr>
<tr>
<td></td>
<td>Specifies which transform sets can be used with the crypto map entry.</td>
</tr>
</tbody>
</table>
## Configuring Dynamic IPsec Virtual Tunnel Interfaces

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(ipsec-profile)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits ipsec profile configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>interface virtual-template number type tunnel</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# interface virtual-template 2 type tunnel</td>
</tr>
<tr>
<td></td>
<td>Defines a virtual-template tunnel interface and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>tunnel mode ipsec ipv4</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel mode ipsec ipv4</td>
</tr>
<tr>
<td></td>
<td>Defines the mode for the tunnel.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>tunnel protection IPsec profile profile-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel protection ipsec profile PROF</td>
</tr>
<tr>
<td></td>
<td>Associates a tunnel interface with an IPsec profile.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>crypto isakamp profile profile-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# crypto isakamp profile profile1</td>
</tr>
<tr>
<td></td>
<td>Defines the ISAKMP profile to be used for the virtual template.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>match identity address ip-address mask</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0</td>
</tr>
<tr>
<td></td>
<td>Matches an identity from the ISAKMP profile and enters isakmp-profile configuration mode.</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>virtual template template-number</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# virtual-template 1</td>
</tr>
<tr>
<td></td>
<td>Specifies the virtual template attached to the ISAKMP profile.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>end</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# end</td>
</tr>
<tr>
<td></td>
<td>Exits global configuration mode and enters privileged EXEC mode.</td>
</tr>
</tbody>
</table>
Configuring Multi-SA Support for Dynamic Virtual Tunnel Interfaces Using IKEv1

Note
Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

SUMMARY STEPS

1. enable
2. configure terminal
3. ip vrf vrf-name
4. rd route-distinguisher
5. exit
6. crypto keyring keyring-name
7. pre-shared-key address key key
8. exit
9. crypto isakmp profile profile-name
10. keyring keyring-name
11. match identity address mask
12. virtual-template template-number
13. exit
14. crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3]
15. exit
16. crypto ipsec profile name
17. set security-policy limit maximum-limit
18. set transform-set transform-set-name [transform-set-name2 .... transform-set-name6]
19. exit
20. interface virtual-template number type tunnel
21. ip vrf forwarding vrf-name
22. ip unnumbered type number
23. tunnel mode ipsec ipv4
24. tunnel protection profile ipsec profile-name
25. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>ip vrf vrf-name</td>
<td>Defines the VRF instance and enters VRF configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# ip vrf VRF-100-1</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>rd route-distinguisher</td>
<td>Creates routing and forwarding tables for a VRF.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-vrf)# rd 100:21</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>exit</td>
<td>Exits VRF configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-vrf)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>crypto keyring keyring-name</td>
<td>Defines a crypto key ring and enters key ring configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# crypto keyring cisco-100-1</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>pre-shared-key address key key</td>
<td>Defines the preshared key to be used for Internet Key Exchange (IKE) authentication.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-keyring)# pre-shared-key address 10.1.1.1 key cisco-100-1</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>exit</td>
<td>Exits keyring configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-keyring)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>crypto isakmp profile profile-name</td>
<td>Defines an ISAKMP profile and enters ISAKMP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# crypto isakmp profile cisco-isakmp-profile-100-1</td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>keyring keyring-name</td>
<td>Configures a key ring in ISAKMP mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(conf-isa-prof)# keyring cisco-100-1</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>match identity address mask</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-isa-prof)# match identity address 10.1.1.0 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>virtual-template template-number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-isa-prof)# virtual-template 101</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-isa-prof)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3]</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# crypto ipsec transform-set cisco esp-aes esp-sha-hmac</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-crypto-trans)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>crypto ipsec profile name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# crypto ipsec profile cisco-ipsec-profile-101</td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td>set security-policy limit maximum-limit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(ipsec-profile)# set security-policy limit 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong></td>
<td>set transform-set transform-set-name [transform-set-name2 .... transform-set-name6]</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(ipsec-profile)# set transform-set cisco</td>
<td></td>
</tr>
<tr>
<td><strong>Step 19</strong></td>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(ipsec-profile)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 20</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>interface virtual-template number type tunnel</td>
<td>Creates a virtual template interface that can be configured interface and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# interface virtual-template 101 type tunnel</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 21</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ip vrf forwarding vrf-name</td>
<td>Associates a VRF instance with a virtual-template interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# ip vrf forwarding VRF-100-1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 22</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ip unnumbered type number</td>
<td>Enables IP processing on an interface without assigning an explicit IP address to the interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# ip unnumbered GigabitEthernet 0.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 23</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tunnel mode ipsec ipv4</td>
<td>Defines the mode for the tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel mode ipsec ipv4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 24</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tunnel protection profile ipsec profile-name</td>
<td>Associates a tunnel interface with an IPsec profile.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# tunnel protection ipsec profile PROF</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 25</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>end</td>
<td>Exits interface configuration mode, and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>
Configuring IPsec Mixed Mode Support for SVTIs

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto IPsec profile profile-name
4. set transform-set transform-set-name [transform-set-name2...transform-set-name6]
5. exit
6. interface type number
7. ip address address mask
8. Do one of the following:
   • tunnel mode ipsec ipv4 v6-overlay
   • tunnel mode ipsec ipv6 v4-overlay
9. tunnel source interface-type interface-type
10. tunnel destination ip-address
11. tunnel protection IPsec profile profile-name
12. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td>* Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>crypto IPsec profile profile-name</td>
<td>Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# crypto IPsec profile PROF</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies which transform sets can be used with the crypto map entry.</td>
<td></td>
</tr>
<tr>
<td><strong>set transform-set</strong> transform-set-name [transform-set-name2...transform-set-name6]</td>
<td>Specifies which transform sets can be used with the crypto map entry.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(ipsec-profile)# set transform-set tset</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits IPsec profile configuration mode, and enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>exit</strong></td>
<td>Exits IPsec profile configuration mode, and enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(ipsec-profile)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Specifies the interface on which the tunnel will be configured and enters interface configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>interface type number</strong></td>
<td>Specifies the interface on which the tunnel will be configured and enters interface configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# interface tunnel 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Specifies the IP address and mask.</td>
<td></td>
</tr>
<tr>
<td><strong>ip address address mask</strong></td>
<td>Specifies the IP address and mask.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-if)# ip address 10.1.1.1 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Defines the mode for the tunnel.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Do one of the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- tunnel mode ipsec ipv4 v6-overlay</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- tunnel mode ipsec ipv6 v4-overlay</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-if)# tunnel mode ipsec ipv4 v6-overlay</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Specifies the tunnel source as a loopback interface.</td>
<td></td>
</tr>
<tr>
<td><strong>tunnel source interface-type interface-type</strong></td>
<td>Specifies the tunnel source as a loopback interface.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-if)# tunnel source loopback 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Identifies the IP address of the tunnel destination.</td>
<td></td>
</tr>
<tr>
<td><strong>tunnel destination ip-address</strong></td>
<td>Identifies the IP address of the tunnel destination.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-if)# tunnel destination 172.16.1.1</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring IPsec Mixed Mode Support for Dynamic VTIs

#### SUMMARY STEPS

1. enable
2. configure terminal
3. crypto ipsec profile *profile-name*
4. set mixed mode
5. set transform-set *transform-set-name* [transform-set-name2...transform-set-name6]
6. exit
7. interface virtual-template *number* type tunnel
8. tunnel mode ipsec ipv4
9. tunnel protection IPsec profile *profile-name*
10. exit
11. crypto isakamp profile *profile-name*
12. match identity address *ip-address* mask
13. virtual template *template-number*
14. end

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>enable</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td>configure terminal</td>
</tr>
</tbody>
</table>

**Example:**

```
Device# configure terminal
```

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>crypto ipsec profile profile-name</td>
<td>Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(config)# crypto ipsec profile PROF
```

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>set mixed mode</td>
<td>Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(config)# set mixed mode
```

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>set transform-set transform-set-name transform-set-name2...transform-set-name6</td>
<td>Specifies which transform sets can be used with the crypto map entry.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(ipsec-profile)# set transform-set tset
```

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exit</td>
<td>Exits ipsec profile configuration mode and enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(ipsec-profile)# exit
```

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>interface virtual-template number type tunnel</td>
<td>Defines a virtual-template tunnel interface and enters interface configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(config)# interface virtual-template 2 type tunnel
```

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tunnel mode ipsec ipv4</td>
<td>Defines the mode for the tunnel.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(config-if)# tunnel mode ipsec ipv4
```

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tunnel protection IPsec profile profile-name</td>
<td>Associates a tunnel interface with an IPsec profile.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(config-if)# tunnel protection ipsec profile PROF
```

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exit</td>
<td>Exits interface configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Device(config-if)# exit
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong></td>
<td>crypto isakamp profile <em>profile-name</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# crypto isakamp profile profile1</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>match identity address <em>ip-address mask</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>virtual template <em>template-number</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# virtual-template 1</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>end</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# end</td>
</tr>
</tbody>
</table>

### Configuration Examples for IPsec Virtual Tunnel Interfaces

**Example: Static Virtual Tunnel Interface with IPsec**

The following example configuration uses a preshared key for authentication between peers. VPN traffic is forwarded to the IPsec VTI for encryption and then sent out the physical interface. The tunnel on subnet 10 checks packets for the IPsec policy and passes them to the Crypto Engine (CE) for IPsec encapsulation. The figure below illustrates the IPsec VTI configuration.

*Figure 5: VTI with IPsec*

![Figure 5: VTI with IPsec](image)

**Cisco 7206 Router Configuration**

```
version 12.3
service timestamps debug datetime
```
service timestamps log datetime
hostname 7200-3
no aaa new-model
ip subnet-zero
ip cef
controller ISA 6/1
!
crypto isakmp policy 1
  encr aes
  authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-aes esp-sha-hmac
crypto ipsec profile P1
  set transform-set T1
!
interface Tunnel0
  ip address 10.0.51.203 255.255.255.0
  ip ospf mtu-ignore
  load-interval 30
  tunnel source 10.0.149.203
  tunnel destination 10.0.149.217
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile P1
!
interface Ethernet3/0
  ip address 10.0.149.203 255.255.255.0
duplex full
!
interface Ethernet3/3
  ip address 10.0.35.203 255.255.255.0
duplex full
  !
ip classless
  ip route 10.0.36.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
del

Cisco 1750 Router Configuration

version 12.3
hostname c1750-17
no aaa new-model
ip subnet-zero
ip cef
crypto isakmp policy 1
  encr aes
  authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-aes esp-sha-hmac
crypto ipsec profile P1
  set transform-set T1
!
interface Tunnel0
  ip address 10.0.51.203 255.255.255.0
  ip ospf mtu-ignore
  tunnel source 10.0.149.203
  tunnel destination 10.0.149.217
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile P1
!
interface FastEthernet0/0
  ip address 10.0.149.203 255.255.255.0
  speed 100
  full-duplex
!
interface Ethernet1/0
  ip address 10.0.35.203 255.255.255.0
Example: Verifying the Results for the IPsec Static Virtual Tunnel Interface

This section provides information that you can use to confirm that your configuration is working properly. In this display, Tunnel 0 is "up," and the line protocol is "up." If the line protocol is "down," the session is not active.

Verifying the Cisco 7206 Status IPsec Static Virtual Tunnel Interface

Router# show interface tunnel 0
Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport ipsec/ip, key disabled, sequencing disabled
Tunnel TTL 255
Checksumming of packets disabled, fast tunneling enabled
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPsec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue: 0/0 (size/max/drops/flushes); Total output drops: 0
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
59968 packets output, 15369696 bytes, 0 underruns
59968 packets output, 15369696 bytes, 0 underruns
0 output buffer failures, 0 output buffers swapped out
Router# show crypto session
Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPsec FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAS: 4, origin: crypto map
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
L1 - IS-IS level-1, L2 - IS-IS level-2
L1/L2 - IS-IS level-1/level-2
I - IS-IS inter area, * - candidate default, U - per-user static route
O - ODR, P - periodic downloaded static route
Gateway of last resort is not set
Example: VRF-Aware Static Virtual Tunnel Interface

To add the VRF to the static VTI example, include the `ipvrf` and `ip vrf forwarding` commands to the configuration as shown in the following example.

**Cisco 7206 Router Configuration**

```plaintext
hostname cisco 7206

ip vrf sample-vti
  rd 1:1
  route-target export 1:1
  route-target import 1:1

interface Tunnel0
  ip vrf forwarding sample-vti
  ip address 10.0.51.217 255.255.255.0
  tunnel source 10.0.149.217
  tunnel destination 10.0.149.203
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile P1

end
```

Example: Static Virtual Tunnel Interface with QoS

You can apply any QoS policy to the tunnel endpoint by including the `service-policy` statement under the tunnel interface. The following example shows how to police traffic out the tunnel interface.

**Cisco 7206 Router Configuration**

```plaintext
hostname cisco 7206

class-map match-all VTI
  match any

policy-map VTI
  class VTI
    police cir 2000000
    conform-action transmit
    exceed-action drop

interface Tunnel0
  ip address 10.0.51.217 255.255.255.0
  tunnel source 10.0.149.217
  tunnel destination 10.0.149.203
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile P1
```
Example: Static Virtual Tunnel Interface with Virtual Firewall

Applying the virtual firewall to the SVTI tunnel allows traffic from the spoke to pass through the hub to reach the Internet. The figure below illustrates an SVTI with the spoke protected inherently by the corporate firewall.

**Figure 6: Static VTI with Virtual Firewall**

The basic SVTI configuration has been modified to include the virtual firewall definition:

**Cisco 7206 Router Configuration**

```
hostname cisco 7206

ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp

interface GigabitEthernet0/1
description Internet Connection
ip address 172.18.143.246 255.255.255.0
ip access-group 100 in
ip nat outside

interface Tunnel0
ip address 10.0.51.217 255.255.255.0
ip nat inside
ip inspect IOSFW1 in
tunnel source 10.0.149.217
tunnel destination 10.0.149.203
tunnel mode ipsec ipv4
```
tunnel protection ipsec profile P1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny esp any any
access-list 110 deny udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny udp any eq non500-isakmp any
!
end

Example: Dynamic Virtual Tunnel Interface Easy VPN Server

The following example illustrates the use of the DVTI Easy VPN server, which serves as an IPsec remote access aggregator. The client can be a home user running a Cisco VPN client or a Cisco IOS router configured as an Easy VPN client.

Cisco 7206 Router Configuration

cisco 7206
!
controller ISA 1/1
!
crypto isakmp policy 1
encr aes
authentication pre-share

group 14
!
crypto isakmp client configuration group group1
key cisco123
pool group1pool
!
crypto isakmp profile vpn1-ra
match identity group group1
client authentication list local_list
isakmp authorization list local_list
client configuration address respond
virtual-template 1
!
crypto ipsec transform-set VTI-TS esp-aes esp-sha-hmac
!
crypto ipsec profile test-vti1
set transform-set VTI-TS
!
interface GigabitEthernet0/1
Example: Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server

The following examples show that a DVTI has been configured for an Easy VPN server.

Router# show running-config interface Virtual-Access2
Building configuration...
Current configuration : 250 bytes

interface Virtual-Access2
  ip unnumbered GigabitEthernet0/1
  ip virtual-reassembly
tunnel source 172.18.143.246
tunnel destination 172.18.143.208
tunnel mode ipsec ipv4
tunnel protection ipsec profile test-vti1
no tunnel protection ipsec initiate
end

Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
* - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.2.1.10 to network 0.0.0.0
172.18.0.0/24 is subnetted, 1 subnets
  C 172.18.143.0 is directly connected, GigabitEthernet0/1
192.168.1.0/32 is subnetworked, 1 subnets
  S 192.168.1.1 [1/0] via 0.0.0.0, Virtual-Access2
10.0.0.0/24 is subnetworked, 1 subnets
  C 10.2.1.0 is directly connected, GigabitEthernet0/2
  S* 0.0.0.0/0 [1/0] via 172.18.143.1

Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under a Virtual Template

The following example shows how to configure VRF-aware IPsec under a virtual template to take advantage of the DVTI:

hostname cisco 7206
!
ip vrf VRF-100-1

Security for VPNs with IPsec Configuration Guide, Cisco IOS XE Fuji 16.7.x
Example: VRF-Aware IPsec with Dynamic VTI When VRF Is Configured Under a Virtual Template with the Gateway Option in an IPsec Profile

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI, when the VRF is configured under a virtual template with the gateway option in an IPsec profile.

hostname ASR 1000
!
ip vrf VRF-100-1
   rd 1:1!
ip vrf VRF-100-2
   rd 1:1
!
!
crypto keyring cisco-100-1
   pre-shared-key address 10.1.1.1 key cisco-100-1

crypto keyring cisco-100-2
   pre-shared-key address 10.1.2.1 key cisco-100-2

crypto isakmp profile cisco-isakmp-profile-100-1
   keyring cisco-100-1
   match identity address 10.1.1.0 255.255.255.0
   virtual-template 101

crypto isakmp profile cisco-isakmp-profile-100-2
   keyring cisco-100-2
   match identity address 10.1.2.0 255.255.255.0
   virtual-template 102
!
!
crypto ipsec transform-set cisco esp-aes esp-sha-hmac
!
crypto ipsec profile cisco-ipsec-profile-101
   set security-policy limit 3
   set transform-set cisco
!
crypto ipsec profile cisco-ipsec-profile-102
   set security-policy limit 5
   set transform-set Cisco
!
interface Virtual-Template101 type tunnel
   ip vrf forwarding VRF-100-1
   ip unnumbered Ethernet 0/0
   tunnel mode ipsec ipv4
   tunnel protection ipsec profile cisco-ipsec-profile-101
!
interface Virtual-Template102 type tunnel
   ip vrf forwarding VRF-100-2
   ip unnumbered Ethernet 0/0
   tunnel mode ipsec ipv4
   tunnel protection ipsec profile cisco-ipsec-profile-102
!
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile

crypto isakmp profile cisco-isakmp-profile-100-2
  keyring cisco-100-2
  match identity address 10.1.2.0 255.255.255.0
  virtual-template 102
!
crypto ipsec transform-set cisco esp-3des esp-sha-hmac
!
crypto ipsec profile cisco-ipsec-profile-101
  set security-policy limit 3
  set transform-set cisco
  set reverse-route gateway 172.16.0.1
!
crypto ipsec profile cisco-ipsec-profile-102
  set security-policy limit 5
  set transform-set cisco
  set reverse-route gateway 172.16.0.1
!
interface Virtual-Template101 type tunnel
  ip vrf forwarding VRF-100-1
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile-101
!
interface Virtual-Template102 type tunnel
  ip vrf forwarding VRF-100-2
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile-102
!
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile

hostname cisco 7206
!
ip vrf VRF-100-1
  rd 1:1
!
ip vrf VRF-100-2
  rd 1:1
!
crypto keyring cisco-100-1
  pre-shared-key address 10.1.1.1 key cisco-100-1
!
crypto keyring cisco-100-2
  pre-shared-key address 10.1.2.1 key cisco-100-2
!
crypto isakmp profile cisco-isakmp-profile-100-1
  vrf VRF-100-1
  keyring cisco-100-1
  match identity address 10.1.1.0 255.255.255.0
  virtual-template 1
!
crypto isakmp profile cisco-isakmp-profile-100-2
  vrf VRF-100-2
  keyring cisco-100-2
  match identity address 10.1.2.0 255.255.255.0
  virtual-template 1
!
crypto ipsec transform-set cisco esp-aes esp-sha-hmac
!
crypto ipsec profile cisco-ipsec-profile
  set security-policy limit 3
  set transform-set cisco
!
interface Virtual-Template 1 type tunnel
  ip unnumbered ethernet 0/0
  tunnel mode ipsec ipv4
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile and a Gateway Option in an IPsec Profile

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI, when the VRF is configured under an ISAKMP profile and a gateway option in an IPsec profile:

```
hostname ASR 1000
!
ip vrf VRF-100-1
  rd 1:1
!
ip vrf VRF-100-2
  rd 1:1
!
crypto keyring cisco-100-1
  pre-shared-key address 10.1.1.1 key cisco-100-1

crypto keyring cisco-100-2
  pre-shared-key address 10.1.2.1 key cisco-100-2

crypto isakmp profile cisco-isakmp-profile-100-1
  vrf VRF-100-1
  keyring cisco-100-1
  match identity address 10.1.1.0 255.255.255.0
  virtual-template 1


crypto isakmp profile cisco-isakmp-profile-100-2
  vrf VRF-100-2
  keyring cisco-100-2
  match identity address 10.1.2.0 255.255.255.0
  virtual-template 1
!
crypto ipsec transform-set cisco esp-3des esp-sha-hmac

crypto ipsec profile cisco-ipsec-profile
  set security-policy limit 3
  set transform-set cisco
  set reverse-route gateway 172.16.0.1
!
!
interface Virtual-Template1 type tunnel
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile
!
```

Example: VRF-Aware IPsec with a Dynamic VTI When a VRF Is Configured Under Both a Virtual Template and an ISAKMP Profile

Note: When separate VRFs are configured under an ISAKMP profile and a virtual template, the VRF configured under the virtual template takes precedence. This configuration is not recommended.
The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI when the VRF is configured under both a virtual template and an ISAKMP profile:

```cisco
hostname ASR 1000
.
.
.
ip vrf test-vti2
 rd 1:2
 route-target export 1:1
 route-target import 1:1
.
.
.
ip vrf test-vti1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
.
.
.
crypto isakmp profile cisco-isakmp-profile
crypto vrf test-vti2
crypto keyring key
match identity address 10.1.1.0 255.255.255.0
.
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback 0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
.
.
.
end
```

Example: Dynamic Virtual Tunnel Interface with Virtual Firewall

The DVTI Easy VPN server can be configured behind a virtual firewall. Behind-the-firewall configuration allows users to enter the network, while the network firewall is protected from unauthorized access. The virtual firewall uses Context-Based Access Control (CBAC) and NAT applied to the Internet interface as well as to the virtual template.

```cisco
hostname Cisco 7206
.
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
.
.
interface GigabitEthernet0/1
description Internet Connection
ip address 172.18.143.246 255.255.255.0
```
Example: Dynamic Virtual Tunnel Interface with QoS

You can add QoS to the DVTI tunnel by applying the service policy to the virtual template. When the template is cloned to make the virtual access interface, the service policy will also be applied to the virtual access interface. The following example shows the basic DVTI configuration with QoS added.

```plaintext
hostname cisco 7206
.
.
class-map match-all VTI
  match any
.
policy-map VTI
  class VTI
    police cir 2000000
    conform-action transmit
    exceed-action drop
  .
.
interface Virtual-Template1 type tunnel
  ip vrf forwarding test-vti1
  ip unnumbered Loopback0
  ip virtual-reassembly
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile test-vti1
  service-policy output VTI
  .
  .
end
```
# Additional References for IPsec Virtual Tunnel Interface

## Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>Security commands</td>
<td>• Cisco IOS Security Command Reference Commands A to C</td>
</tr>
<tr>
<td></td>
<td>• Cisco IOS Security Command Reference Commands D to L</td>
</tr>
<tr>
<td></td>
<td>• Cisco IOS Security Command Reference Commands M to R</td>
</tr>
<tr>
<td></td>
<td>• Cisco IOS Security Command Reference Commands S to Z</td>
</tr>
<tr>
<td>IPsec configuration</td>
<td>Configuring Security for VPNs with IPsec</td>
</tr>
<tr>
<td>QoS configuration</td>
<td>Cisco IOS Quality of Service Solutions Configuration Guide</td>
</tr>
<tr>
<td>EasyVPN configuration</td>
<td>• Cisco Easy VPN Remote</td>
</tr>
<tr>
<td></td>
<td>• Easy VPN Server</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

## Standards and RFCs

<table>
<thead>
<tr>
<th>Standard/RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2401</td>
<td>Security Architecture for the Internet Protocol</td>
</tr>
<tr>
<td>RFC 2408</td>
<td>Internet Security Association and Key Management Protocol</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 Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

Feature Information for IPsec Virtual Tunnel Interfaces

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic IPsec VTIs</td>
<td>Cisco IOS XE Release 2.1</td>
<td>Dynamic VTIs enable efficient use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. IPsec dynamic VTIs allow you to create highly secure connectivity for remote access VPNs. The dynamic VTI simplifies VRF-aware IPsec deployment. The following commands were introduced or modified: <code>crypto isakmp profile</code>, <code>interface virtual-template</code>, <code>show vtemplate</code>, <code>tunnel mode</code>, <code>virtual-template</code>.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FlexVPN Mixed Mode Support</td>
<td>Cisco IOS XE Release 3.10S</td>
<td>The FlexVPN Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic. This feature is only supported for Remote Access VPN with IKEv2 and Dynamic VTI.</td>
</tr>
<tr>
<td>Multi-SA for Dynamic VTIs</td>
<td>Cisco IOS XE Release 3.2S</td>
<td>The DVTI can accept multiple IPsec selectors that are proposed by the initiator. The following commands were introduced or modified: set security-policy limit, set reverse-route.</td>
</tr>
<tr>
<td>Static IPsec VTIs</td>
<td>Cisco IOS XE Release 2.1</td>
<td>IPsec VTIs provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.</td>
</tr>
<tr>
<td>Tunnel Mode Auto Selection</td>
<td>Cisco IOS XE Release 3.12S</td>
<td>The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. The following command was introduced or modified: virtual-template</td>
</tr>
</tbody>
</table>
The FlexVPN Mixed Mode v6 over v4 Transport feature provides support for carrying IPv6 traffic over IPsec IPv4 transport. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlexVPN Mixed Mode v6 over v4 Transport</td>
<td>Cisco IOS XE Everest 16.4.1</td>
<td>The FlexVPN Mixed Mode v6 over v4 Transport feature provides support for carrying IPv6 traffic over IPsec IPv4 transport. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.</td>
</tr>
</tbody>
</table>
CHAPTER 4

Deleting Crypto Sessions of Revoked Peer Certificates

The Delete Crypto Sessions of Revoked Peer Certificates on CRL Download feature deletes an active crypto session with a peer if its certificate is found to be revoked when downloading a new CRL.

- Finding Feature Information, page 79
- Restrictions for Deleting Crypto Sessions of Revoked Peer Certificates, page 79
- Information About Deleting Crypto Sessions of Revoked Peer Certificates, page 80
- How to Enable Deletion of Crypto Sessions for Revoked Peer Certificates, page 80
- Configuration Examples for Deleting Crypto Sessions of Revoked Peer Certificates, page 82
- Additional References for Deleting Crypto Sessions of Revoked Peers, page 83
- Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates, page 84

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Deleting Crypto Sessions of Revoked Peer Certificates

- If revocation check is turned off and this feature is enabled, the IKE database is not populated with the number of sessions. The show outputs do not display information about the deleted sessions.
Information About Deleting Crypto Sessions of Revoked Peer Certificates

How a Crypto Session is Deleted

1. When negotiating via certificate authentication, the peer sends the CERT payload to the device, which parses each certificate to store information about serial number and the issuer names. This information forms the list of serial numbers issued by the corresponding CA server and is passed to PKI for revocation check.

2. If the revocation-check crl command is configured for a trustpoint, PKI informs IKE about the revocation check thereby disabling IKE from unnecessarily storing unwanted peer certification information.

3. After a successful CRL download, PKI sends IKE a notification, which contains the "issuer-name." The CRL signature and content is verified. If there is no change in CRL content, PKI does not notify IKE.

4. If PKI notifies IKE containing the issuer name, IKE prepares a list of serial numbers for an issuer name and passes this list to PKI to verify if the serial numbers in the list are revoked.

5. PKI performs revocation check on the serial number list received from the IKE and checks the list against the downloaded CRL. The revoked serial number list is returned to IKE.

6. On a notification from PKI containing the list of revoked serial numbers, IKE identifies and deletes sessions pertaining to those serial numbers those sessions.

How to Enable Deletion of Crypto Sessions for Revoked Peer Certificates

Enabling Deletion of Crypto Sessions

Perform this task to enable the deletion of crypto sessions for revoked certificates.
Deleting Crypto Sessions of Revoked Peer Certificates

SUMMARY STEPS

1. enable
2. clear crypto session
3. configure terminal
4. Do one of the following:
   • crypto isakmp disconnect-revoked-peers
   • crypto ikev2 disconnect-revoked-peers
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td><strong>Step 2</strong> clear crypto session (Optional) Deletes IPsec crypto sessions and IKE and security associations.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Note</strong> Use this command to enable the feature for previously established sessions, else the feature is enabled for new sessions only.</td>
</tr>
<tr>
<td><strong>Step 3</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Step 4</strong> Do one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> crypto isakmp disconnect-revoked-peers</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>crypto ikev2 disconnect-revoked-peers</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Step 5</strong> end (Disconnects IKE or IKEv2 crypto sessions with peers having revoked certificates. For this command to take effect, reconnected the existing sessions.)</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

Security for VPNs with IPsec Configuration Guide, Cisco IOS XE Fuji 16.7.x
Verifying the Delete Crypto Session Capability for a Revoked Peer Certificate

Perform this task to verify if the delete crypto session capability is displayed in the show output.

SUMMARY STEPS

1. enable
2. show crypto isakmp peers
3. show crypto ikev2 session detail

DETAILED STEPS

Step 1  enable

Example:
Device> enable
Enables privileged EXEC mode.
• Enter your password if prompted.

Step 2  show crypto isakmp peers

Example:
Device# show crypto isakmp peers
Displays Internet Security Association and Key Management Protocol (ISAKMP) peer descriptions.

Step 3  show crypto ikev2 session detail

Example:
Device# show crypto ikev2 session detail
Displays the status of active Internet Key Exchange Version 2 (IKEv2) sessions.

Configuration Examples for Deleting Crypto Sessions of Revoked Peer Certificates

Example: Enabling Deletion of Crypto Sessions for an IKE Session

Device> enable
Device# clear crypto session
Device# configure terminal
Example: Enabling Deletion of Crypto Sessions for an IKEv2 Session

Device> enable
Device# clear crypto session
Device# configure terminal
Device(config)# crypto ikev2 disconnect-revoked-peers
Device# show crypto ikev2 session detail

Session-id:1, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id  Local  Remote  fvr/ivrf  Status
 1 10.0.0.1/500 10.0.0.2/500 (none)/(none) READY
Encc: 3DES, Hash: SHA96, DH Grp:2, Auth: PSK
Life/Remaining/Active Time: 86400/86157/248 sec
CE id: 0, Session-id: 1, MIB-id: 1
Status Description: Negotiation done
Local spi: 750CBE827434A245  Remote spi: 4353FEBABE8F24C
Local req mess id: 0  Remote req mess id: 0
Local next mess id: 0  Remote next mess id: 2
Local req queued: 0  Remote req queued: 0
Local window: 5  Remote window: 5
DPD configured for 0 seconds
NAT-T is not detected
Disconnect Revoked Peer: Enabled
Child sa: local selector 10.0.0.1/0 - 10.0.0.1/65535
remote selector 10.0.0.2/0 - 10.0.0.2/65535
ESP spi in/out: 0x9360A95/0x6C340600
CPI in/out: 0x9FE5/0xC776
AH spi in/out: 0x0/0x0
Encr: AES CBC, keysizes: 128, esp_hmac: SHA96
ah_hmac: Unknown - 0, comp: IPCOMP_LZS, mode tunnel

Additional References for Deleting Crypto Sessions of Revoked Peers

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
</tbody>
</table>
Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
### Table 4: Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete crypto session(s) of revoked peer cert(s) on CRL download</td>
<td>Cisco IOS XE Release 3.13S</td>
<td>The Delete Crypto Sessions of Revoked Peer Certificates on CRL Download feature deletes an active crypto session with a peer if its certificate is found to be revoked when downloading a new CRL. In Cisco IOS XE Release 3.13S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: <code>crypto ikev2 disconnect-revoked-peers</code>, <code>crypto isakmp disconnect-revoked-peers</code>, <code>show crypto isakmp peers</code>, <code>show crypto ikev2 session detail</code>.</td>
</tr>
</tbody>
</table>
Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates
Crypto Conditional Debug Support

The Crypto Conditional Debug Support feature introduces new debug commands that allow users to debug an IP Security (IPsec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPsec operations and reducing the amount of debug output, users can better troubleshoot a router with a large number of tunnels.

- Finding Feature Information, page 87
- Prerequisites for Crypto Conditional Debug Support, page 87
- Restrictions for Crypto Conditional Debug Support, page 88
- Information About Crypto Conditional Debug Support, page 88
- How to Enable Crypto Conditional Debug Support, page 90
- Configuration Examples for the Crypto Conditional Debug CLIs, page 92
- Additional References, page 93
- Feature Information for Crypto Conditional Debug Support, page 95

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Crypto Conditional Debug Support

To use the new crypto CLIs, you must be using the k9 subsystem crypto image in Cisco IOS XE releases.
Restrictions for Crypto Conditional Debug Support

- Although conditional debugging is useful for troubleshooting peer-specific or functionality related Internet Key Exchange (IKE) and IPSec problems, conditional debugging may not be able to define and check large numbers of debug conditions. Because extra space is needed to store the debug condition values, additional processing overhead is added to the CPU and memory usage is increased. Thus, enabling crypto conditional debugging on a router with heavy traffic should be used with caution.

Information About Crypto Conditional Debug Support

Supported Condition Types

The new crypto conditional debug CLIs—`debug crypto condition`, `debug crypto condition unmatched`, and `show crypto debug-condition`—allow you to specify conditions (filter values) in which to generate and display debug messages related only to the specified conditions. The table below lists the supported condition types.

Note

The `debug crypto condition peer` command with the `ipv4` or `ipv6` keyword can provide the hardware platform specific debugging output. The rest of the condition filters do not provide platform specific debugging output.

<table>
<thead>
<tr>
<th>Condition Type (Keyword)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>connid</code></td>
<td>An integer between 1-32766. Relevant debug messages will be shown if the current IPSec operation uses this value as the connection ID to interface with the crypto engine.</td>
</tr>
<tr>
<td><code>FVRF</code></td>
<td>The name string of a virtual private network (VPN) routing and forwarding (VRF) instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its front-door VRF (FVRF).</td>
</tr>
<tr>
<td><code>ikev2</code></td>
<td>The name string for an IKEv2 profile. Relevant debug messages will be shown if the IKEv2 profile name is specified.</td>
</tr>
<tr>
<td><code>isakmp</code></td>
<td>The name string for an ISAKMP profile. Relevant debug messages will be shown if the ISAKMP profile name is specified.</td>
</tr>
<tr>
<td>Condition Type (Keyword)</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IVRF</td>
<td>The name string of a VRF instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its inside VRF (IVRF).</td>
</tr>
<tr>
<td>local</td>
<td>The name string of an IPv4 or IPv6 local address.</td>
</tr>
<tr>
<td>peer group</td>
<td>A Unity group-name string. Relevant debug messages will be shown if the peer is using this group name as its identity.</td>
</tr>
<tr>
<td>peer hostname</td>
<td>A fully qualified domain name (FQDN) string. Relevant debug messages will be shown if the peer is using this string as its identity; for example, if the peer is enabling IKE Xauth with this FQDN string.</td>
</tr>
<tr>
<td>peer ipv4 or peer ipv6</td>
<td>A single IP address. Relevant debug messages will be shown if the current IPsec operation is related to the IP address of this peer.</td>
</tr>
<tr>
<td>peer subnet</td>
<td>A subnet and a subnet mask that specify a range of peer IP addresses. Relevant debug messages will be shown if the IP address of the current IPsec peer falls into the specified subnet range.</td>
</tr>
<tr>
<td>peer username</td>
<td>A username string. Relevant debug messages will be shown if the peer is using this username as its identity; for example, if the peer is enabling IKE Extended Authentication (Xauth) with this username.</td>
</tr>
<tr>
<td>session</td>
<td>Provides information about crypto sessions.</td>
</tr>
<tr>
<td>SPI</td>
<td>A 32-bit unsigned integer. Relevant debug messages will be shown if the current IPSec operation uses this value as the SPI.</td>
</tr>
<tr>
<td>unmatched</td>
<td>Provides debug messages when context information is unavailable.</td>
</tr>
</tbody>
</table>

---

1. If an IPSec connid, flowid, or SPI is used as a debug condition, the debug messages for a related IPSec flow are generated. An IPSec flow has two connids, flowids, and SPIs—one inbound and one outbound. Both two connids, flowids, and SPIs can be used as the debug condition that triggers debug messages for the IPSec flow.
How to Enable Crypto Conditional Debug Support

Enabling Crypto Conditional Debug Messages

Performance Considerations

• Before enabling crypto conditional debugging, you must decide what debug condition types (also known as debug filters) and values will be used. The volume of debug messages is dependent on the number of conditions you define.

Note: Specifying numerous debug conditions may consume CPU cycles and negatively affect router performance.

• Your router will perform conditional debugging only after at least one of the global crypto debug commands--`debug crypto isakmp`, `debug crypto ipsec`, and `debug crypto engine`--has been enabled. This requirement helps to ensure that the performance of the router will not be impacted when conditional debugging is not being used.

Disable Crypto Debug Conditions

If you choose to disable crypto conditional debugging, you must first disable any crypto global debug CLIs you have issued; thereafter, you can disable conditional debugging.

Note: The reset keyword can be used to disable all configured conditions at one time.

SUMMARY STEPS

1. `enable`
2. `debug crypto condition [connid integer engine-id integer] [flowid integer engine-id integer] [fvrf string] [ivrf string] [peer [group string] [hostname string] [ipv4 ipaddress] [subnet subnet mask] [username string]] [spi integer] [reset]`
3. `show crypto debug-condition {[peer] [connid] [spi] [fvrf] [ivrf] [unmatched]}`
4. `debug crypto isakmp`
5. `debug crypto ipsec`
6. `debug crypto engine`
7. `debug crypto condition unmatched [isakmp | ipsec | engine]`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
<td>Enables privileged EXEC mode.  &lt;br&gt;  - Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>debug crypto condition [connid integer</td>
<td>Defines conditional debug filters.</td>
</tr>
<tr>
<td></td>
<td>engine-id integer] [flowid integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>engine-id integer] [fvrf string] [ivrf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>string] [peer [group string] [hostname</td>
<td></td>
</tr>
<tr>
<td></td>
<td>string] [ipv4 ipaddress] [subnet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subnet mask] [username string]] [spi</td>
<td></td>
</tr>
<tr>
<td></td>
<td>integer] [reset]</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router# debug crypto condition connid 2000 engine-id 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>show crypto debug-condition {[peer] [connid] [spi] [fvrf] [ivrf] [unmatched]}</td>
<td>Displays crypto debug conditions that have already been enabled in the router.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show crypto debug-condition spi</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>debug crypto isakmp</td>
<td>Enables global IKE debugging.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# debug crypto isakmp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>debug crypto ipsec</td>
<td>Enables global IPSec debugging.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# debug crypto ipsec</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>debug crypto engine</td>
<td>Enables global crypto engine debugging.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# debug crypto engine</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>debug crypto condition unmatched [isakmp [ipsec</td>
<td>(Optional) Displays debug conditional crypto messages when no context information is available to check against debug conditions.  &lt;br&gt;  If none of the optional keywords are specified, all crypto-related information will be shown.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# debug crypto condition unmatched ipsec</td>
<td></td>
</tr>
</tbody>
</table>
Enabling Crypto Error Debug Messages

To enable crypto error debug messages, you must perform the following tasks.

debug crypto error CLI

Enabling the `debug crypto error` command displays only error-related debug messages, thereby, allowing you to easily determine why a crypto operation, such as an IKE negotiation, has failed within your system.

Note When enabling this command, ensure that global crypto debug commands are not enabled; otherwise, the global commands will override any possible error-related debug messages.

SUMMARY STEPS

1. enable
2. debug crypto isakmp | ipsec | engine} error

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> debug crypto isakmp</td>
<td>ipsec</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# debug crypto ipsec error</td>
<td></td>
</tr>
</tbody>
</table>

Configuration Examples for the Crypto Conditional Debug CLIs

Enabling Crypto Conditional Debugging Example

The following example shows how to display debug messages when the peer IP address is 10.1.1.1, 10.1.1.2, or 10.1.1.3, and when the connection-ID 2000 of crypto engine 0 is used. This example also shows how to
enable global debug crypto CLIs and enable the `show crypto debug-condition` command to verify conditional settings.

```
Router# debug crypto condition conid 2000 engine-id 1
Router# debug crypto condition peer ipv4 10.1.1.1
Router# debug crypto condition peer ipv4 10.1.1.2
Router# debug crypto condition peer ipv4 10.1.1.3
Router# debug crypto condition unmatched
! Verify crypto conditional settings.
Router# show crypto debug-condition
Crypto conditional debug currently is turned ON
IKE debug context unmatched flag:ON
IPsec debug context unmatched flag:ON
Crypto Engine debug context unmatched flag:ON
IKE peer IP address filters:
10.1.1.1 10.1.1.2 10.1.1.3
Connection-id filters:[conid:engine_id]2000:1,
! Enable global crypto CLIs to start conditional debugging.
Router# debug crypto isakmp
Router# debug crypto ipsec
Router# debug crypto engine
```

### Disabling Crypto Conditional Debugging Example

The following example shows how to disable all crypto conditional settings and verify that those settings have been disabled:

```
Router# debug crypto condition reset
! Verify that all crypto conditional settings have been disabled.
Router# show crypto debug-condition
Crypto conditional debug currently is turned OFF
IKE debug context unmatched flag:OFF
IPsec debug context unmatched flag:OFF
Crypto Engine debug context unmatched flag:OFF
```

### Additional References

The following sections provide references to the Crypto Conditional Debug Support feature.

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPSec and IKE configuration tasks</td>
<td>“Internet Key Exchange for IPsec VPNs” module in the Cisco IOS XE Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>IPSec and IKE commands</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
</tbody>
</table>
Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
Feature Information for Crypto Conditional Debug Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for Crypto Conditional Debug Support

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crypto Conditional Debug Support</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The Crypto Conditional Debug Support feature introduces three new command-line interfaces (CLIs) that allow users to debug an IP Security (IPSec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI).</td>
</tr>
</tbody>
</table>
CHAPTER 6

IPv6 over IPv4 GRE Tunnel Protection

The IPv6 over IPv4 GRE Tunnel Protection feature allows both IPv6 unicast and multicast traffic to pass through a protected generic routing encapsulation (GRE) tunnel.

- Finding Feature Information, page 97
- Prerequisites for IPv6 over IPv4 GRE Tunnel Protection, page 97
- Restrictions for IPv6 over IPv4 GRE Tunnel Protection, page 98
- Information About IPv6 over IPv4 GRE Tunnel Protection, page 98
- How to Configure IPv6 over IPv4 GRE Tunnel Protection, page 100
- Configuration Examples for IPv6 over IPv4 GRE Tunnel Protection, page 107
- Additional References, page 108
- Feature Information for IPv6 over IPv4 GRE Tunnel Protection, page 109

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for IPv6 over IPv4 GRE Tunnel Protection

- To enable this feature, you must configure IPsec tunnel protection on an IPv4 GRE tunnel.
- To enable IPv6 multicast, you must configure IPv6 multicast routing.
Restrictions for IPv6 over IPv4 GRE Tunnel Protection

The IPv6 over IPv4 GRE Tunnel Protection feature supports IPv6 over IPv4 point-to-point GRE tunnel protection and not IPv6 over IPv4 mGRE tunnel protection.

Information About IPv6 over IPv4 GRE Tunnel Protection

GRE Tunnels with IPsec

Generic routing encapsulation (GRE) tunnels sometimes are combined with IPSec, because IPSec does not support IPv6 multicast packets. This function prevents dynamic routing protocols from running successfully over an IPSec VPN network. Because GRE tunnels do support IPv6 multicast, a dynamic routing protocol can be run over a GRE tunnel. Once a dynamic routing protocol is configured over a GRE tunnel, you can encrypt the GRE IPv6 multicast packets using IPSec.

IPSec can encrypt GRE packets using a crypto map or tunnel protection. Both methods specify that IPSec encryption is performed after GRE encapsulation is configured. When a crypto map is used, encryption is applied to the outbound physical interfaces for the GRE tunnel packets. When tunnel protection is used, encryption is configured on the GRE tunnel interface.

The following figure shows encrypted packets that enter a router through a GRE tunnel interface using a crypto map on the physical interface. Once the packets are decrypted and decapsulated, they continue to their IP destination as clear text.

Figure 7: Using a Crypto Map to Configure IPv6 over IPv4 GRE Tunnel Encryption
The following figure shows encryption using tunnel protection command on the GRE tunnel interface. The encrypted packets enter the router through the tunnel interface and are decrypted and decapsulated before they continue to their destination as clear text.

**Figure 8: Using Tunnel Protection to Configure IPv6 over IPv4 GRE Tunnel Encryption**

There are two key differences in using the crypto map and tunnel protection methods:

- The IPSec crypto map is tied to the physical interface and is checked as packets are forwarded out through the physical interface. At this point, the GRE tunnel has already encapsulated the packet.
- Tunnel protection ties the encryption functionality to the GRE tunnel and is checked after the packet is GRE encapsulated but before the packet is handed to the physical interface.
How to Configure IPv6 over IPv4 GRE Tunnel Protection

Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map

SUMMARY STEPS

1. enable
2. configure terminal
3. ipv6 multicast-routing
4. ipv6 unicast-routing
5. interface type number
6. ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}
7. tunnel mode {aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ip | gre ipv6 | ipip [decapsulate-any] | ipsec ipv4 | iptalk | ipv6 | ipsec ipv6 | mpls | nos | rbscp}
8. tunnel source {ip-address | ipv6-address | interface-typeinterface-number}
9. tunnel destination {hostname | ip-address | ipv6-address}
10. exit
11. crypto isakmp policy priority
12. authentication {rsa-sig | rsa-encr | pre-share}
13. hash {sha | md5}
14. group {1 | 2 | 5}
15. encryption {des | 3des | aes 192 | aes 256}
16. exit
17. crypto isakmp key enc-type-digit keystring {address peer-address [mask] | ipv6 {ipv6-address/ipv6-prefix | hostname hostname} | no-xauth}
18. crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3] [transform4]
19. access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit} protocol source source wildcard destination destination wildcard [precedence precedence] [tos tos] [time-range time-range-name] [fragments] [log [word] | log-input [word]]
20. crypto map [ipv6] map-name seq-num [ipsec-isakmp [dynamic dynamic-map-name | discover | profile profile-name]]
21. set peer {hostname [dynamic] [default] | ip-address [default]}
22. set transform-set transform-set-name [transform-set-name2...transform-set-name6]
23. match address [access-list-id | name]
24. exit
25. interface type number
26. crypto map map-name [redundancy standby-group-name [stateful]]
27. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 multicast-routing</td>
<td>Enables multicast routing using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD) on all IPv6-enabled interfaces of the router and enables multicast forwarding.</td>
</tr>
<tr>
<td>Example: Router(config)# ipv6 multicast-routing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enable this command only if you are using IPv6 multicast. If you are using IPv6 unicast, you need not enable this command.</td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv6 unicast-routing</td>
<td>Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td>Example: Router(config)# ipv6 unicast-routing</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> interface type number</td>
<td>Specifies a tunnel interface and number, and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# interface tunnel 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> ipv6 address {ipv6-address/prefix-length</td>
<td>prefix-name sub-bits/prefix-length}</td>
</tr>
<tr>
<td>Example: Router(config-if)# ipv6 address 0:0:0:7272::72/64</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> tunnel mode {aurp</td>
<td>cayman</td>
</tr>
<tr>
<td>Example: Router(config-if)# tunnel mode gre ip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> tunnel source {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example: Router(config-if)# tunnel source ethernet0</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 9</strong> tunnel destination `{hostname</td>
<td>ip-address</td>
</tr>
<tr>
<td><strong>Step 10</strong> exit</td>
<td>Exits interface configuration mode and returns to global configuration mode. Example: Router(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 11</strong> crypto isakmp policy <code>priority</code></td>
<td>Defines an Internet Key Exchange (IKE) policy, and enters ISAKMP policy configuration mode. • Policy number 1 indicates the policy with the highest priority. The lower the <code>priority</code> argument value, the higher the priority. Example: Router(config)# crypto isakmp policy 15</td>
</tr>
<tr>
<td><strong>Step 12</strong> authentication `{rsa-sig</td>
<td>rsa-encr</td>
</tr>
<tr>
<td><strong>Step 13</strong> hash `{sha</td>
<td>md5}`</td>
</tr>
<tr>
<td><strong>Step 14</strong> group `{1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Step 15</strong> encryption `{des</td>
<td>3des</td>
</tr>
<tr>
<td><strong>Step 16</strong> exit</td>
<td>Exits ISAKMP policy configuration mode and enters global configuration mode. Example: Router(config-isakmp-policy)# exit</td>
</tr>
<tr>
<td><strong>Step 17</strong> crypto isakmp key `enc-type-digit keystring {address peer-address [mask]</td>
<td>ipv6 {ipv6-address/ipv6-prefix}</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 18** | **crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3] [transform4]**
| | Defines a transform set.  
| **Example:** | `Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des` |
| **Step 19** | **access-list access-list-number [dynamic dynamic-name [timeout minutes]|deny|permit] [source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [time-range time-range-name] [fragments] [log [word] | log-input [word]]**
| | Defines an extended IP access list.  
| **Example:** | `Router(config)# access-list 110 permit gre host 192.168.0.16 host 172.16.0.12` |
| **Step 20** | **crypto map [ipv6] map-name seq-num [ipsec-isakmp [dynamic dynamic-map-name | discover | profile profile-name]]**
| | Creates a new crypto map entry or profile and enters crypto map configuration mode.  
| **Example:** | `Router(config)# crypto map mymap 10 ipsec-isakmp` |
| **Step 21** | **set peer [hostname [dynamic] [default] | ip-address [default]]**
| | Specifies an IP Security (IPsec) peer in a crypto map entry.  
| **Example:** | `Router(config-crypto-map)# set peer 10.0.0.1` |
| **Step 22** | **set transform-set transform-set-name [transform-set-name2...transform-set-name6]**
| | Specifies the transform set that can be used with the crypto map entry.  
| **Example:** | `Router(config-crypto-map)# set transform-set myset0` |
| **Step 23** | **match address [access-list-id | name]**
| | Specifies an extended access list for a crypto map entry.  
| **Example:** | `Router(config-crypto-map)# match address 102` |
| **Step 24** | **exit**
| | Exits crypto map configuration mode and returns to global configuration mode.  
| **Example:** | `Router(config-crypto-map)# exit` |
| **Step 25** | **interface type number**
| | Specifies an interface and number and enters interface configuration mode.  
| **Example:** | `Router(config)# interface ethernet 1` |
### Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection

**SUMMARY STEPS**

1. enable
2. configure terminal
3. ipv6 multicast-routing
4. ipv6 unicast-routing
5. crypto isakmp policy priority
6. authentication {rsa-sig | rsa-encr | pre-share}
7. hash {sha | md5}
8. group {1 | 2 | 5}
9. encryption {des | 3des | aes | aes 192 | aes 256}
10. exit
11. crypto isakmp key enc-type-digit keystring {address peer-address [mask] | ipv6 {ipv6-address/ipv6-prefix} | hostname hostname} [no-xauth]
12. crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3] [transform4]
13. crypto ipsec profile profile-name
14. set transform-set transform-set-name [transform-set-name2...transform-set-name6]
15. exit
16. interface type number
17. ipv6 address {ipv6-address / prefix-length | prefix-name sub-bits/prefix-length}
18. tunnel mode {aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ip | gre ipv6 | ipip[decapsulate-any] | ipsec ipv4 | iptalk | ipv6 | ipsec ipv6 | mpls | nos | rbscp}
19. tunnel source {ip-address | ipv6-address | interface-type interface-number}
20. tunnel destination {hostname | ip-address | ipv6-address}
21. tunnel protection ipsec profile name [shared]
22. end
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** configure    | Enters global configuration mode.                                      |
| terminal                |                                                                         |
| Example:                | Router# configure terminal                                             |

| **Step 3** ipv6         | Enables multicast routing using Protocol Independent Multicast (PIM)   |
| multicast-routing       | and Multicast Listener Discovery (MLD) on all IPv6-enabled interfaces   |
| Example:                | of the router and enables multicast forwarding.                        |
| Router(config)# ipv6    | • Enable this command only if you are using IPv6 multicast. If you are  |
| multicast-routing       | using IPv6 unicast, you do not need to enable this command.            |

| **Step 4** ipv6         | Enables the forwarding of IPv6 unicast datagrams.                      |
| unicast-routing         |                                                                         |
| Example:                | Router(config)# ipv6 unicast-routing                                   |

| **Step 5** crypto       | Defines an IKE policy, and enters ISAKMP policy configuration mode.     |
| isakmp policy priority  | Policy number 1 indicates the policy with the highest priority. The    |
| Example:                | lower the priority argument value, the higher the priority.           |
| Router(config)# crypto  |                                                                         |
| isakmp policy priority  | 15                                                                      |

| **Step 6** authentication| Specifies the authentication method within an Internet Key Exchange     |
| {rsa-sig | rsa-encr | pre-share}                | (IKE) policy.                                                          |
| Example:                  | • The rsa-sig and rsa-encr keywords are not supported in IPv6.          |
| Router(config-isakmp-policy)# authentication pre-share |                                                               |

| **Step 7** hash sha md5 | Specifies the hash algorithm within an IKE policy.                     |
| Example:                | Router(config-isakmp-policy)# hash md5                                  |

<p>| <strong>Step 8</strong> group 1 2 5 | Specifies the Diffie-Hellman group identifier within an IKE policy.    |
| Example:                | Router(config-isakmp-policy)# group 2                                   |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong></td>
<td>encryption {des</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-isakmp-policy)# encryption 3des</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-isakmp-policy)# exit</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>crypto isakmp key enc-type-digit keystring {address peer-address [mask]</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3] [transform4]</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>crypto ipsec profile profile-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto ipsec profile ipsecprof</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>set transform-set transform-set-name [transform-set-name2...transform-set-name6]</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(ipsec-profile)# set transform-set myset0</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(ipsec-profile)# exit</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>interface type number</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# interface tunnel 1</td>
</tr>
</tbody>
</table>
### IPv6 over IPv4 GRE Tunnel Protection

#### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 17</strong></td>
<td>ipv6 address {ipv6-address / prefix-length</td>
</tr>
<tr>
<td></td>
<td>Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127</td>
</tr>
<tr>
<td><strong>Step 18</strong></td>
<td>tunnel mode {aurp</td>
</tr>
<tr>
<td></td>
<td>Specifies a GRE IPv6 tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-if)# tunnel mode gre ip</td>
</tr>
<tr>
<td><strong>Step 19</strong></td>
<td>tunnel source {ip-address</td>
</tr>
<tr>
<td></td>
<td>Specifies the source address or the source interface type and number for the tunnel interface.</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-if)# tunnel source 10.0.0.1</td>
</tr>
<tr>
<td><strong>Step 20</strong></td>
<td>tunnel destination {hostname</td>
</tr>
<tr>
<td></td>
<td>Specifies the destination address or hostname for the tunnel interface.</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-if)# tunnel destination 172.16.0.12</td>
</tr>
<tr>
<td><strong>Step 21</strong></td>
<td>tunnel protection ipsec profile name [shared]</td>
</tr>
<tr>
<td></td>
<td>Associates a tunnel interface with an IPsec profile.</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-if)# tunnel protection ipsec profile ipsecprof</td>
</tr>
<tr>
<td><strong>Step 22</strong></td>
<td>end</td>
</tr>
<tr>
<td></td>
<td>Exits interface configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-if)# end</td>
</tr>
</tbody>
</table>

---

## Configuration Examples for IPv6 over IPv4 GRE Tunnel Protection

### Example: Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map

```
Router> enable
Router# configure terminal
Router(config)# ipv6 multicast-routing
Router(config)# ipv6 unicast-routing
Router(config)# interface tunnel 10
Router(config-if)# ipv6 address my-prefix 0:0:7272::72/64
```
Example: Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection

The following example configures IPsec tunnel protection on an IPv4 GRE tunnel. IPv6 multicast routing is enabled using the `ipv6 multicast-routing` command.

```
Router(config)# ipv6 multicast-routing
Router(config)# ipv6 unicast-routing
Router(config-isakmp-policy)# authentication pre-share
Router(config-isakmp-policy)# hash md5
Router(config-isakmp-policy)# group 2
Router(config-isakmp-policy)# encryption 3des
Router(config-isakmp-policy)# exit
Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0
Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0
```

Additional References

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 Multicast Routing</td>
<td>IPv6 Implementation Guide</td>
</tr>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
</tbody>
</table>
**Feature Information for IPv6 over IPv4 GRE Tunnel Protection**

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
The IPv6 over IPv4 GRE tunnel protection feature allows both IPv6 unicast and multicast traffic to pass through a protected GRE tunnel. The following sections provide information about this feature:

- Information About IPv6 over IPv4 GRE Tunnel Protection
- How to Configure IPv6 over IPv4 GRE Tunnel Protection
RFC 430x IPsec Support

The RFC 430x IPsec Support includes features—RFC 430x IPsec Support Phase 1 and RFC 430x IPsec Support Phase 2—that implement Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

- Finding Feature Information, page 111
- Information About RFC 430x IPsec Support, page 111
- How to Configure RFC 430x IPsec Support, page 112
- Configuration Examples for RFC 430x IPsec Support, page 116
- Additional References for RFC 430x IPsec Support, page 117
- Feature Information for RFC 430x IPsec Support, page 118

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About RFC 430x IPsec Support

RFC 430x IPsec Support Phase 1

The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

RFC 4301 specifies the base architecture for IPsec-compliant systems. RFC 4301 describes how to provide a set of security services for traffic at the IP layer, in both the IPv4 and IPv6 environments. The RFC 430x
IPsec Support Phase 1 feature provides support for the following RFC 4301 implementations on Cisco IOS software.

- **Security association (SA) lifetime**—The lifetime of a security association between IPsec and Internet Key Exchange (IKE) or Internet Key Exchange Version 2 (IKEv2) must not exceed the lifetime of the authentication certificate.

- **OPAQUE selectors**—OPAQUE indicates that the corresponding selector field is not available for verification. When IKEv2 encounters an OPAQUE selector, IKEv2 skips, does not process the OPAQUE selector, and moves to next selector for policy verification.

- **Explicit Congestion Notification (ECN) support**—ECN is propagated when decrypting an IPsec packet thereby ensuring the packet source and destination are aware of congestion that occurs within the network.

- **Fragment processing**—Peers must not send Initial and noninitial fragments in the same tunnel. There must be a separate tunnel mode SA for carrying initial and noninitial fragments and separate tunnel mode SA for noninitial fragments. IPsec peers must support discarding of packets and stateful fragment checking to accommodate bypass traffic.

- **Do not fragment-(DF) bit processing**—DF-bit processing must be set on a per SA basis.

- **Dummy packet generation support**—It should be possible to send dummy packets via IPsec SA to encapsulate the packets when traffic is flowing via IPsec SA tunnel.

---

### RFC 430x IPsec Support Phase 2

The RFC 430x IPsec Support Phase 2 feature provides support for the RFC 4301 implementation of encryption and decryption of Internet Control Message Protocol (ICMP) packets on Cisco IOS software.

ICMP error messages are sent when an ICMP error occurs. For example, when a host is not reachable, the intermediate device sends a message to the originator of the ICMP request that the host is not reachable. When an ICMP error message reaches an IPsec encryption policy, it may not be classified to match an existing SA. So, the packets are classified based on the data inside the ICMP error message. This data contains the source and destination address of the original ICMP message. If an SA is found based on the address in the ICMP error message, the SA is used. If there is no SA, an SA is created if the policy permits. For decryption, the post decrypt check is performed on the data inside the ICMP error message if a valid SA is not found.

The encryption and decryption of ICMP error messages can be verified through the encrypt and decrypt counters displayed in the output of the `show crypto ipsec sa` command.

Use the conditional debug commands `debug platform condition feature ipsec dataplane submode feature level info`, `debug platform condition both`, and `debug platform condition start` to view ICMP error message classification.

---

### How to Configure RFC 430x IPsec Support

#### Configuring RFC 430x IPsec Support Globally

Perform this task to configure the RFC 4301 implementations globally.
### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto ipsec security-association dummy {pps rate | seconds seconds}`
4. `crypto ipsec security-association ecn {discard | propogate}`
5. `exit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| **Example:** | Device> enable |
| **Step 2** `configure terminal` | Enters global configuration mode. |
| **Example:** | Device# configure terminal |
| **Step 3** `crypto ipsec security-association dummy {pps rate | seconds seconds}` | Enables the generation and transmission of dummy packets in an IPsec traffic flow. |
| **Example:** | Device(config)# crypto ipsec security-association dummy seconds 5 |
| **Step 4** `crypto ipsec security-association ecn {discard | propogate}` | Enables the Explicit Congestion Notification (ECN) settings in an IPsec traffic flow. |
| **Example:** | Device(config)# crypto ipsec security-association ecn discard |
| **Step 5** `exit` | Exits global configuration mode and returns to privileged EXEC mode. |
| **Example:** | Device(config-crypto-map)# exit |

---

**Configuring RFC 430x IPsec Support Per Crypto Map**

Perform this task to configure the RFC 4301 implementations per crypto map.
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto map map-name seq-num ipsec-isakmp
4. set ipsec security-association dfbit {clear | copy | set}
5. set ipsec security-association dummy {pps rate | seconds seconds}
6. set ipsec security-association ecn {discard | propagate}
7. end
8. show crypto map ipsec sa

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | enable | Enables privileged EXEC mode.  
<p>| Example: | Device&gt; enable | • Enter your password if prompted. |
| Step 2 | configure terminal | Enters global configuration mode. |
| Example: | Device# configure terminal | |
| Step 3 | crypto map map-name seq-num ipsec-isakmp | Specifies the crypto map entry to be created or modified and enters crypto map configuration mode. |
| Example: | Device(config)# crypto map cmap 1 ipsec-isakmp | |
| Step 4 | set ipsec security-association dfbit {clear | copy | set} | Enables do not fragment (DF)-bit processing per security association (SA) for an IPsec traffic flow in a crypto map. |
| Example: | Device(config-crypto-map)# set ipsec security-association dfbit set | |
| Step 5 | set ipsec security-association dummy {pps rate | seconds seconds} | Enables the generation and transmission of dummy packets for an IPsec traffic flow in a crypto map. |
| Example: | Device(config-crypto-map)# set ipsec security-association dummy seconds 5 | |
| Step 6 | set ipsec security-association ecn {discard | propagate} | Enables the Explicit Congestion Notification (ECN) settings per SA for an IPsec traffic flow in a crypto map. |
| Example: | Device(config-crypto-map)# set ipsec security-association ecn propagate | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>end</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-crypto-map)# end</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>show crypto map ipsec sa</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# show crypto map ipsec sa</td>
</tr>
</tbody>
</table>

The following is sample output from the `show crypto map ipsec sa` command:

```
Device# show crypto map ipsec sa

interface: Tunnel0
Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
protected vrf: (none)
current peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
#pkts decaps: 28, #pkts decryp: 28, #pkts verify: 28
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
#send dummy packets 852600, #recv dummy packets 424905

local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
current outbound spi: 0xE963D1EC(3915633132)

PFS (Y/N): N, DH group: none
Dummy packet: Initializing

inbound esp sas:
spi: 0x4E01B9A4108327834
 transform: esp-3des esp-md5-hmac,
in use settings ={Tunnel, }
conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

sa timing: remaining key lifetime (k/sec): (4608000/2343)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0x963D1EC915633132
 transform: esp-3des esp-md5-hmac,
in use settings ={Tunnel, }
conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

sa timing: remaining key lifetime (k/sec): (4608000/2343)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)
Configuration Examples for RFC 430x IPsec Support

Example: Configuring RFC 430x IPsec Support Globally

The following examples shows how to configure RFC 430x IPsec Support globally:

Device> enable
Device# configure terminal
Device(config)# crypto ipsec security-association dummy seconds 15
Device(config)# crypto ipsec security-association ecn propogate
Device(config-crypto-map)# exit

Example: Configuring RFC 430x IPsec Support Per Crypto Map

The following examples shows how to configure RFC 430x IPsec Support per crypto map:

Device> enable
Device# configure terminal
Device(config)# crypto map cmap 1 ipsec-isakmp
Device(config-crypto-map)# set security-association copy
Device(config-crypto-map)# set security-association dummy seconds 15
Device(config-crypto-map)# set security-association ecn propogate
Device(config-crypto-map)# end
Device# show crypto map ipsec sa

interface: Tunnel0
Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
protected vrf: (none)
current_peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
#pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decaps: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
#send dummy packets 852600, #recv dummy packets 424905
local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
current outbound spi: 0xE963D1EC(3915633132)
PFS (Y/N): N, DH group: none
Dummy packet: Initializing
inbound esp sas:
spi: 0xF4E01B9A(4108327834)
transform: esp-3des esp-md5-hmac,
in use settings ={Tunnel, }
conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFFFF80000049, crypto map: Tunnel0-head-0
sa timing: remaining key lifetime (k/sec): (4608000/2343)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)
inbound ah sas:
inbound pcp sas:

outbound esp sas:
spi: 0x863D1DE(391563132)
transform: esp-3des esp-md5-hmac,
in use settings = {Tunnel, }
conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFF80000049, crypto map: Tunnel0-head-0

sa timing: remaining key lifetime (k/sec): (4608000/2343)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE

outbound ah sas:

outbound pcp sas:

## Additional References for RFC 430x IPsec Support

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS Commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>Security commands</td>
<td>• Cisco IOS Security Command Reference Commands A to C</td>
</tr>
<tr>
<td></td>
<td>• Cisco IOS Security Command Reference Commands D to L</td>
</tr>
<tr>
<td></td>
<td>• Cisco IOS Security Command Reference Commands M to R</td>
</tr>
<tr>
<td></td>
<td>• Cisco IOS Security Command Reference Commands S to Z</td>
</tr>
<tr>
<td>IKEv2 configuration</td>
<td>Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

### Standards and RFCs

<table>
<thead>
<tr>
<th>Standard/RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 4301</td>
<td>Security Architecture for the Internet Protocol</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

Feature Information for RFC 430x IPsec Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for RFC 430x IPsec Support

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC430x IPsec Support Phase 1</td>
<td>Cisco IOS XE Release 3.12S</td>
<td>The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301. The following commands were introduced or modified: <code>crypto ipsec security-association dummy</code>, <code>crypto ipsec security-association ecn</code>, <code>set ipsec security-association dfbit</code>, <code>set ipsec security-association dummy</code>, <code>set ipsec security-association ecn</code>, <code>show crypto map ipsec sa</code>.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| RFC430x IPsec Support Phase 2 | Cisco IOS XE Release 3.14S | The RFC 430x IPsec Support Phase 2 feature provides support for the RFC 4301 implementation of encryption and decryption of Internet Control Message Protocol (ICMP) packets on Cisco IOS software.  
In Cisco IOS XE Release 3.14S, this feature was introduced on Cisco 4431 Integrated Services Router and Cisco 4451-X Integrated Services Router.  
In Cisco IOS XE Release 3.15S, this feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers and Cisco CSR 1000V Series Cloud Services Router.  
No commands were modified or updated for this feature. |