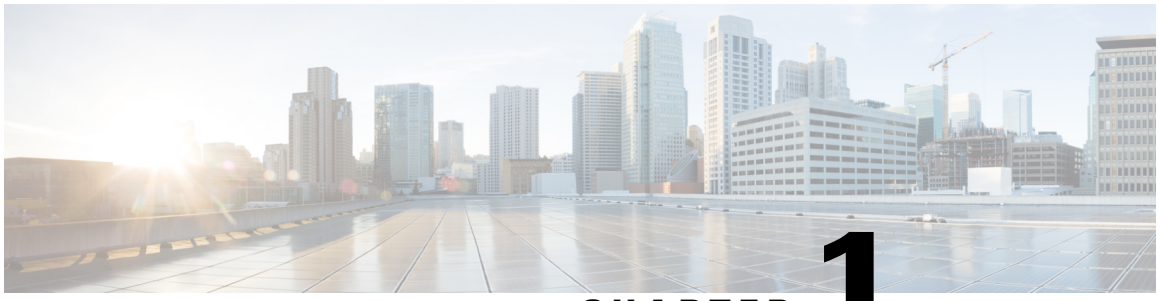
IPsec Data Plane Configuration Guide, Cisco IOS Release 15M&T

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

IPsec Anti-Replay Window Expanding and Disabling

Cisco IP security (IPsec) authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. The decryptor keeps track of which packets it has seen on the basis of these numbers. Currently, the default window size is 64 packets. Generally, this number (window size) is sufficient, but there are times when you may want to expand this window size. The IPsec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep track of more than 64 packets.

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.

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 at the end of this module.

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 IPsec Anti-Replay Window Expanding and Disabling

- Before configuring this feature, you should have already created a crypto profile.

Information About IPsec Anti-Replay Window Expanding and Disabling

IPsec Anti-Replay Window

Cisco IPsec authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. (Security association [SA] anti-replay is a security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks.) The decryptor checks off the sequence numbers that it has seen before. The encryptor assigns sequence numbers in an increasing order. The decryptor remembers the value X of the highest sequence number that it has already seen. N is the window size, and the decryptor also remembers whether it has seen packets having sequence numbers from X-N+1 through X. Any packet with the sequence number X-N is discarded. Currently, N is set at 64, so only 64 packets can be tracked by the decryptor.

At times, however, the 64-packet window size is not sufficient. For example, Cisco quality of service (QoS) gives priority to high-priority packets, which could cause some low-priority packets to be discarded when they arrive outside of the 64 packet replay window at the decryptor. The IPsec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep track of more than 64 packets.

Increasing the anti-replay window size has no impact on throughput and security. The impact on memory is insignificant because only an extra 128 bytes per incoming IPsec SA is needed to store the sequence number on the decryptor. It is recommended that you use the full 1024 window size to eliminate any future anti-replay problems.

How to Configure IPsec Anti-Replay Window Expanding and Disabling

Configuring IPsec Anti-Replay Window Expanding and Disabling Globally

To configure IPsec Anti-Replay Window: Expanding and Disabling globally (so that it affects all SAs that are created), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association replay window-size** [*N*]
4. **crypto ipsec security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association replay window-size** [*N*] **Example:** `Router (config)# crypto ipsec security-association replay window-size 256`	Sets the size of the SA replay window globally. **Note** Configure this command or the **crypto ipsec security-association replay disable** command. The two commands are not used at the same time.
Step 4	**crypto ipsec security-association replay disable** **Example:** `Router (config)# crypto ipsec security-association replay disable`	Disables checking globally. **Note** Configure this command or the **crypto ipsec security-association replay window-size** command. The two commands are not used at the same time.

Configuring IPsec Anti-Replay Window Expanding and Disablingon a Crypto Map

To configure IPsec Anti-Replay Window: Expanding and Disabling on a crypto map so that it affects those SAs that have been created using a specific crypto map or profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**]
4. **set security-association replay window-size [** *N* **]**
5. **set security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] **Example:** `Router (config)# crypto map ETHO 17 ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps.
Step 4	**set security-association replay window-size [** *N* **]** **Example:** `Router (crypto-map)# set security-association replay window-size 128`	Controls the SAs that are created using the policy specified by a particular crypto map, dynamic crypto map, or cyrpto profile. **Note** Configure this command or the **set security-association replay disable** command. The two commands are not used at the same time.
Step 5	**set security-association replay disable** **Example:** `Router (crypto-map)# set security-association replay disable`	Disables replay checking for a particular crypto map, dynamic crypto map, or crypto profile. **Note** Configure this command or the **set security-association replay window-size** command. The two commands are not used at the same time.

Troubleshooting Tips

- If your replay window size has not been set to a number that is high enough for the number of packets received, you will receive a system message such as the following:

```
*Nov 17 19:27:32.279: %CRYPTO-4-PKT_REPLAY_ERR: decrypt: replay check failed connection
id=1
```

The above message is generated when a received packet is judged to be outside the anti-replay window.

Configuration Examples for IPsec Anti-Replay Window Expanding and Disabling

Global Expanding and Disabling of an Anti-Replay Window Example

The following example shows that the anti-replay window size has been set globally to 1024:

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN-Gateway1
!
boot-start-marker
boot-end-marker
!
!
clock timezone EST 0
no aaa new-model
ip subnet-zero
!
!
ip audit po max-events 100
no ftp-server write-enable
!
!
crypto isakmp policy 10
  encryption aes
  authentication pre-share
  group 14
crypto isakmp key cisco123 address 192.165.201.2
crypto ipsec security-association replay window-size 1024
crypto ipsec transform-set basic esp-aes esp-sha-hmac

!
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!

!
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 172.16.2.0 0.0.0.255 access-list 101 remark
 Crypto ACL
!
!
control-plane
!
!
```

```
line con 0
line aux 0
line vty 0 4
!
!
end
```

Expanding and Disabling of an Anti-Replay Window for Crypto Maps or Crypto Profiles Example

The following example shows the expanding and disabling of an anti-replay windoe for a Particular Crypto Map, Dynamic Crypto Map, or Crypto Profile. In this example, anti-replay checking is disabled for IPsec connections to 172.17.150.2 but enabled (and the default window size is 64) for IPsec connections to 172.17.150.3 and 172.17.150.4:

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname networkserver1
!
enable secret 5 $1$KxKv$cbqKsZtQTLJLGPN.tErFZ1 enable password ww !
ip subnet-zero
!
cns event-service server
crypto isakmp policy 1
  encr aes
  authentication pre-share
  group 14
crypto isakmp key cisco170 address 172.17.150.2 crypto isakmp key cisco180
address 172.17.150.3 crypto isakmp key cisco190 address 172.17.150.4
crypto ipsec transform-set 170cisco esp-aes esp-sha-hmac crypto ipsec
transform-set 180cisco esp-aes esp-sha-hmac crypto ipsec transform-set
190cisco esp-aes esp-sha-hmac
crypto map ETH0 17 ipsec-isakmp
  set peer 172.17.150.2
  set security-association replay disable
  set transform-set 170cisco
  match address 170
crypto map ETH0 18 ipsec-isakmp
  set peer 192.168.1.3
  set transform-set 180cisco
  match address 180
crypto map ETH0 19 ipsec-isakmp
  set peer 192.168.1.4
  set transform-set 190cisco
  match address 190 !
interface Ethernet0
 ip address 172.17.150.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no mop enabled
 crypto map ETH0
!
interface Serial0
 ip address 172.16.160.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
ip classless
ip route 172.18.170.0 255.255.255.0 172.17.150.2 ip route 172.19.180.0 255.255.255.0
172.17.150.3 ip route 172.20.190.0 255.255.255.0 172.17.150.4 no ip http server !
access-list 170 permit ip 172.16.160.0 0.0.0.255 172.18.170.0 0.0.0.255 access-list 180
permit ip 172.16.160.0 0.0.0.255 172.19.180.0 0.0.0.255 access-list 190 permit ip 172.16.160.0
```

```
 0.0.0.255 172.20.190.0 0.0.0.255 !
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Security Command Reference
IP security and encryption	Configuring Security for VPNs with IPsec

MIBs

MIBs	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None.	--

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

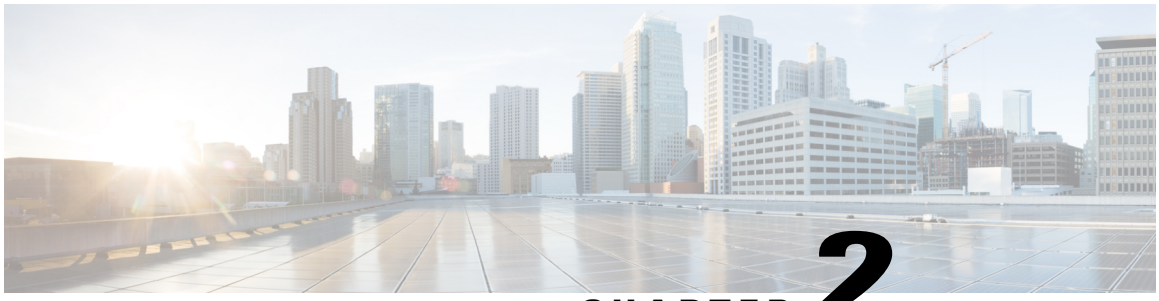
Feature Information for IPsec Anti-Replay Window Expanding and Disabling

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 1: Feature Information for IPsec Anti-Replay Window: Expanding and Disabling

Feature Name	Releases	Feature Information
IPsec Anti-Replay Window: Expanding and Disabling	12.3(14)T 12.2(33)SRA 12.2(33)SRA	Cisco IP security (IPsec) authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. The decryptor keeps track of which packets it has seen on the basis of these numbers. Currently, the default window size is 64 packets. Generally, this number (window size) is sufficient, but there are times when you may want to expand this window size. The IPsec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep track of more than 64 packets. This feature was introduced in Cisco IOS Release 12.3(14)T. This feature was integrated into Cisco IOS Release 12.2(33)SRA. This feature was integrated into Cisco IOS Release 12.2(18)SXF6. The following commands were introduced or modified: **crypto ipsec security-association replay disable, crypto ipsec security-association replay window-size, set security-association replay disable, set security-association replay window-size .**



CHAPTER 2

Pre-Fragmentation for IPsec VPNs

The Pre-Fragmentation for IPsec VPNs feature increases performance between Cisco IOS routers and VPN clients by delivering encryption throughput at maximum encryption hardware accelerator speeds for packets that are near the maximum transmission unit (MTU) size. Packets are fragmented into equally sized units to prevent further downstream fragmentation.

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 at the end of this module.

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 Pre-Fragmentation for IPsec VPNs

Take the following information into consideration before this feature is configured:

- Pre-fragmentation for IPsec VPNs operates in IPsec tunnel mode and IPsec tunnel mode with GRE, but not with IPsec transport mode.
- Pre-fragmentation for IPsec VPNs configured on the decrypting router in a unidirectional traffic scenario does not improve the performance or change the behavior of either of the peers.

- Pre-fragmentation for IPsec VPNs occurs before the transform is applied if compression is turned on for outgoing packets.
- Pre-fragmentation for IPsec VPNs functionality depends on the egress interface **crypto ipsec df-bit** configuration and the incoming packet "do not fragment" (DF) bit state. See the table below.

Table 2: Pre-Fragmentation for IPsec VPNs Dependencies

Pre-Fragmentation for IPsec VPNs Feature State (Enabled/Disabled)	Egress Interface "crypto ipsec df-bit" Configuration	Incoming Packet DF Bit State	Result
Enabled	crypto ipsec df-bit clear	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit clear	1	Fragmentation occurs before encryption.
Disabled	crypto ipsec df-bit clear	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit clear	1	Fragmentation occurs after encryption and packets are reassembled before decryption.
Enabled	crypto ipsec df-bit set	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit set	1	Packets are dropped.
Disabled	crypto ipsec df-bit set	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit set	1	Packets are dropped.
Enabled	crypto ipsec df-bit copy	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit copy	1	Packets are dropped.
Disabled	crypto ipsec df-bit copy	0	Fragmentation occurs after encryption, and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit copy	1	Packets are dropped.

Information About Pre-Fragmentation for IPsec VPNs

Pre-fragmentation for IPsec VPNs

When a packet is nearly the size of the MTU of the outbound link of the encrypting router and it is encapsulated with IPsec headers, it is likely to exceed the MTU of the outbound link. This causes packet fragmentation after encryption. The decrypting router must then reassemble these packets in the process path, which decreases the decrypting router's performance.

The Pre-fragmentation for IPsec VPNs feature increases the decrypting router's performance by enabling it to operate in the high-performance CEF path instead of the process path. An encrypting router can predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPsec security association (SA). If it is predetermined that the packet exceeds the MTU of the output interface, the packet is fragmented before encryption. This function avoids process-level reassembly before decryption and helps improve decryption performance and overall IPsec traffic throughput.

Note The pre-fragmentation feature is turned off by default for tunnel interfaces. To receive pre-fragmentation performance benefits, turn pre-fragmentation on after ensuring that the tunnel interfaces have the same MTU on both ends.

Crypto maps are no longer used to define fragmentation behavior that occurred before and after encryption. Now, IPsec Virtual Tunnel Interface (also referred to as Virtual-Template interface) (VTI) fragmentation behavior is determined by the IP MTU settings that are configured on the VTI.

See the IPsec Virtual Tunnel Interface feature document for more information on VTIs.

Note If fragmentation after-encryption behavior is desired, then set the VTI IP MTU to a value that is greater than the egress router interface IP MTU. Use the **show ip interface tunnel**command to display the IP MTU value.

How to Configure Pre-Fragmentation for IPsec VPNs

Configuring Pre-Fragmentation for IPsec VPNs

Perform this task to configure Pre-Fragmentation for IPsec VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip mtu** *bytes*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config-if)# interface tunnel0`	Specifies the interface on which the VTI is configured and enters interface configuration mode.
Step 4	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 1500` **Example:**	Specifies the VTI MTU size in bytes of IP packets on the egress interface for IPsec VPNs. **Note** If after-encryption fragmentation behavior is desired, then set the VTI IP MTU to a value that is greater than the egress router interface IP MTU. Use the **show ip interface tunnel**command to display the IP MTU value.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference
IPsec	IPsec Virtual Tunnel Interface feature document

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

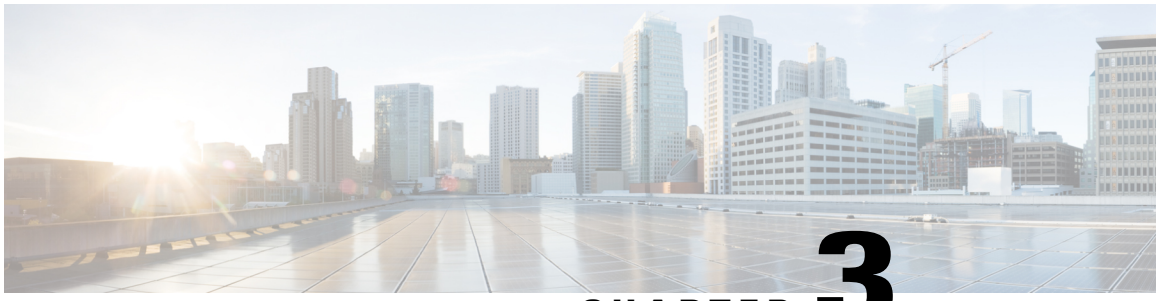
Feature Information for Pre-Fragmentation for IPsec VPNs

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 3: Feature Information for Pre-Fragmentation for IPsec VPNs

Feature Name	Releases	Feature Information
Pre-Fragmentation for IPsec VPNs	12.1(11b)E 12.2(13)T 12.2(14)S	This feature increases performance between Cisco IOS routers and VPN clients by delivering encryption throughput at maximum encryption hardware accelerator speeds for packets that are near the maximum transmission unit (MTU) size. Packets are fragmented into equally sized units to prevent further downstream fragmentation. The following command was introduced or modified: **ip mtu (interface configuration) .**



CHAPTER 3

Invalid Security Parameter Index Recovery

When an invalid security parameter index error (shown as “Invalid SPI”) occurs in IP Security (IPsec) packet processing, the Invalid Security Parameter Index Recovery feature allows for an Internet Key Exchange (IKE) security association (SA) to be established. The “IKE” module sends notification of the “Invalid SPI” error to the originating IPsec peer so that Security Association Databases (SADBs) can be resynchronized and successful packet processing can be resumed.

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.

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 at the end of this module.

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 Invalid Security Parameter Index Recovery

Before configuring the Invalid Security Parameter Index Recovery feature, you must have enabled Internet Key Exchange (IKE) and IPsec on your router.

Restrictions for Invalid Security Parameter Index Recovery

If an IKE SA is being initiated to notify an IPsec peer of an "Invalid SPI" error, there is the risk that a denial-of-service (DoS) attack can occur. The Invalid Security Parameter Index Recovery feature has a built-in mechanism to minimize such a risk, but because there is a risk, the Invalid Security Parameter Index Recovery feature is not enabled by default. You must enable the command using command-line interface (CLI).

Information About Invalid Security Parameter Index Recovery

How the Invalid Security Parameter Index Recovery Feature Works

An IPsec "black hole" occurs when one IPsec peer "dies" (for example, a peer can "die" if a reboot occurs or if an IPsec peer somehow gets reset). Because one of the peers (the receiving peer) is completely reset, it loses its IKE SA with the other peer. Generally, when an IPsec peer receives a packet for which it cannot find an SA, it tries to send an IKE "INVALID SPI NOTIFY" message to the data originator. This notification is sent using the IKE SA. If there is no IKE SA available, the receiving peer drops the packet.

Note A single security association (SA) has only two peers. However, a SADB can have multiple SAs, whereby each SA has an association with a different peer.

When an invalid security parameter index (SPI) is encountered, the Invalid Security Parameter Index feature provides for the setting up of an IKE SA with the originator of the data, and the IKE "INVALID SPI NOTIFY" message is sent. The peer that originated the data "sees" the "INVALID SPI NOTIFY" message and deletes the IPsec SA that has the invalid SPI. If there is further traffic from the originating peer, there will not be any IPsec SAs, and new SAs will be set up. Traffic will flow again. The default behavior (that is, without configuring the Invalid Security Parameter Index Recovery feature) is that the data packet that caused the invalid SPI error is dropped. The originating peer keeps on sending the data using the IPsec SA that has the invalid SPI, and the receiving peer keeps dropping the traffic (thus creating the "black hole").

The IPsec module uses the IKE module to send an IKE "INVALID SPI NOTIFY" message to the other peer. Once the invalid SPI recovery is in place, there should not be any significant dropping of packets although the IPsec SA setup can itself result in the dropping of a few packets.

To configure your router for the Invalid Security Parameter Index Recovery feature, use the **crypto isakmp invalid-spi-recovery** command. The IKE SA will not be initiated unless you have configured this command.

How to Configure Invalid Security Parameter Index Recovery

Configuring Invalid Security Parameter Index Recovery

To configure the Invalid Security Parameter Index Recovery feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp invalid-spi-recovery**

DETAILED STEPS

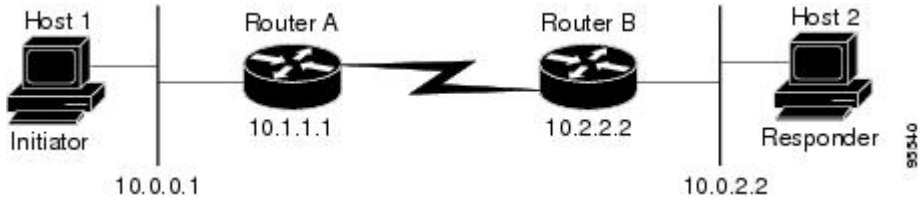
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp invalid-spi-recovery** **Example:** `Router (config)#` **crypto isakmp invalid-spi-recovery**	Initiates the IKE module process whereby the IKE module notifies the receiving peer that an “Invalid SPI” error has occurred.

Verifying the Invalid Security Parameter Index Recovery Configuration

To determine the status of the IPsec SA for traffic between two peers, you can use the **show crypto ipsec sa** command. If the IPsec SA is available on one peer and not on the other, there is a “black hole” situation, in which case you will see the invalid SPI errors being logged for the receiving peer. If you turn console logging on or check the syslog server, you will see that these errors are also being logged.

The figure below shows the topology of a typical preshared configuration setup. Host 1 is the initiating peer (initiator), and Host 2 is the receiving peer (responder).

Figure 1: Preshared Configuration Topology



SUMMARY STEPS

1. Initiate the IKE and IPsec SAs between Host 1 and Host 2
2. Clear the IKE and IPsec SAs on Router B
3. Send traffic from Host 1 to Host 2 and ensure that new IKE and IPsec SAs are correctly established

DETAILED STEPS

Step 1 Initiate the IKE and IPsec SAs between Host 1 and Host 2

Router A

Example:

```
Router# show crypto isakmp sa
  f_vrf/i_vrf   dst             src             state       conn-id slot
       / 10.2.2.2              10.1.1.1        QM_IDLE               1       0
```

Router B

Example:

```
Router# show crypto isakmp sa
  f_vrf/i_vrf   dst             src             state       conn-id slot
       /        10.1.1.1        10.2.2.2        QM_IDLE               1       0
```

Router A

Example:

```
Router# show crypto ipsec sa interface fastethernet0/0
interface: FastEthernet0/0
    Crypto map tag: testtag1, local addr. 10.1.1.1
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   current_peer: 10.2.2.2:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
    #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
     local crypto endpt.: 10.1.1.1, remote crypto endpt.: 10.2.2.2
     path mtu 1500, media mtu 1500
     current outbound spi: 7AA69CB7
     inbound esp sas:
      spi: 0x249C5062(614223970)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537831/3595)
        IV size: 16 bytes
        replay detection support: Y
     inbound ah sas:
      spi: 0xB16D1587(2976716167)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537831/3595)
        replay detection support: Y
     inbound pcp sas:
     outbound esp sas:
      spi: 0x7AA69CB7(2057739447)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537835/3595)
        IV size: 16 bytes
        replay detection support: Y
     outbound ah sas:
      spi: 0x1214F0D(18960141)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537835/3594)
        replay detection support: Y
     outbound pcp sas:
```

Router B

Example:

```
Router# show crypto ipsec sa interface ethernet1/0
interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
    #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
     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: 249C5062
     inbound esp sas:
      spi: 0x7AA69CB7(2057739447)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4421281/3593)
```

```
      IV size: 16 bytes
      replay detection support: Y
    inbound ah sas:
     spi: 0x1214F0D(18960141)
      transform: ah-sha-hmac ,
      in use settings ={Tunnel, }
      slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
      crypto engine type: Hardware
      sa timing: remaining key lifetime (k/sec): (4421281/3593)
      replay detection support: Y
    inbound pcp sas:
    outbound esp sas:
     spi: 0x249C5062(614223970)
      transform: esp-aes esp-sha-hmac ,
      in use settings ={Tunnel, }
      slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
      crypto engine type: Hardware
      sa timing: remaining key lifetime (k/sec): (4421285/3593)
      IV size: 16 bytes
      replay detection support: Y
    outbound ah sas:
     spi: 0xB16D1587(2976716167)
      transform: ah-sha-hmac ,
      in use settings ={Tunnel, }
      slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
      crypto engine type: Hardware
      sa timing: remaining key lifetime (k/sec): (4421285/3592)
      replay detection support: Y
    outbound pcp sas:
```

Step 2 Clear the IKE and IPsec SAs on Router B

Example:

```
Router# clear crypto isakmp
Router# clear crypto sa
Router# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
      /         10.2.2.2.       10.1.1.1      MM_NO_STATE           1       0 (deleted)
Router# show crypto ipsec sa
interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #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
     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: 0
     inbound esp sas:
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
     outbound ah sas:
     outbound pcp sas:
```

Step 3 Send traffic from Host 1 to Host 2 and ensure that new IKE and IPsec SAs are correctly established

Example:

```
ping
```

```
Protocol [ip]: ip
Target IP address: 10.0.2.2
Repeat count [5]: 30
Datagram size [100]: 100
Timeout in seconds [2]:
Extended commands [n]: no
Sweep range of sizes [n]: n
Type escape sequence to abort.
Sending 30, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
..!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 93 percent (28/30), round-trip min/avg/max = 1/3/8 ms
RouterB# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.1.1.1       10.2.2.2     QM_IDLE              3       0
       /        10.1.1.1       10.2.2.2      MM_NO_STATE          1       0 (deleted)
RouterB# show crypto ipsec sa

interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 28, #pkts encrypt: 28, #pkts digest: 28
    #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 decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: D763771F
     inbound esp sas:
      spi: 0xE7AB4256(3886760534)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5127, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        IV size: 16 bytes
        replay detection support: Y
     inbound ah sas:
      spi: 0xF9205CED(4179647725)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5125, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        replay detection support: Y
     inbound pcp sas:
     outbound esp sas:
      spi: 0xD763771F(3613619999)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5128, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3596)
        IV size: 16 bytes
        replay detection support: Y
     outbound ah sas:
      spi: 0xEB95406F(3952427119)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5126, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3595)
        replay detection support: Y
     outbound pcp sas:
RouterA# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
```

```
       /        10.2.2.2       10.1.1.1      MM_NO_STATE           1       0 (deleted)
       /        10.2.2.2       10.1.1.1      QM_IDLE               2       0
    Check for an invalid SPI message on Router B
Router# show logging
Syslog logging: enabled (10 messages dropped, 13 messages rate-limited, 0 flushes, 0 overruns, xml
disabled)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled
    Buffer logging: level debugging, 43 messages logged, xml disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 72 message lines logged
Log Buffer (8000 bytes):
*Mar 24 20:55:45.739: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
        destaddr=10.2.2.2, prot=51, spi=0x1214F0D(18960141), srcaddr=10.1.1.1
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #2,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-aes esp-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:47.743: IPSEC(key_engine): got a queue event with 2 kei messages
*Mar 24 20:55:47.743: IPSEC(spi_response): getting spi 4179647725 for SA
        from 10.2.2.2        to 10.1.1.1     for prot 2
*Mar 24 20:55:47.747: IPSEC(spi_response): getting spi 3886760534 for SA
        from 10.2.2.2     to 10.1.1.1        for prot 3
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524099
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524100
*Mar 24 20:55:48.135: IPSEC(key_engine): got a queue event with 4 kei messages
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xF9205CED(4179647725), conn_id= 939529221, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xEB95406F(3952427119), conn_id= 939529222, keysize= 0, flags= 0xA
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-aes esp-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xE7AB4256(3886760534), conn_id= 939529223, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-aes esp-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xD763771F(3613619999), conn_id= 939529224, keysize= 0, flags= 0xA
*Mar 24 20:55:48.139: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:48.139: IPSEC(mtree_add_ident): src 10.2.2.2, dest 10.1.1.1, dest_port 0
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 51,
    sa_spi= 0xF9205CED(4179647725),
```

```
    sa_trans= ah-sha-hmac , sa_conn_id= 939529221
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.2.2, sa_prot= 51,
    sa_spi= 0xEB95406F(3952427119),
    sa_trans= ah-sha-hmac , sa_conn_id= 939529222
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 50,
    sa_spi= 0xE7AB4256(3886760534),
    sa_trans= esp-aes esp-sha-hmac , sa_conn_id= 939529223
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.2.2, sa_prot= 50,
    sa_spi= 0xD763771F(3613619999),
    sa_trans= esp-aes esp-sha-hmac , sa_conn_id= 939529224
ipseca-72a#
```

Configuration Examples for Invalid Security Parameter Index Recovery

Invalid Security Parameter Index Recovery Example

The following example shows that invalid security parameter index recovery has been configured on Router A and Router B. Invalid Security Parameter Index Recovery Example, on page 25 shows the topology used for this example.

Router A

```
Router# show running-config
Building configuration...
Current configuration : 2048 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service tcp-small-servers
!
hostname ipseca-71a
!
logging queue-limit 100
no logging console
enable secret 5 $1$4GZB$L2YOmnenOCNAu0jgFxebT/
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
```

```
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
  lifetime 180
crypto isakmp key 0 1234 address 10.2.2.2
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-aes esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
 set peer 10.2.2.2
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.0.0.0
 no ip route-cache cef
 duplex full
 speed 100
 crypto map testtag1
!
interface FastEthernet0/1
 ip address 10.0.0.1 255.0.0.0
 no ip route-cache cef
 duplex auto
 speed auto
!
interface Serial1/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial1/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial1/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.3.3.3 255.0.0.0 10.2.0.1
```

```
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.0.1 host 10.0.2.2
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 password lab
 login
!
!
end
ipseca-71a#
```

Router B

```
Router# show running-config
Building configuration...
Current configuration : 2849 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname ipseca-72a
!
logging queue-limit 100
no logging console
enable secret 5 $1$kKqL$5Th5Qhw1ubDkkK90KWFxi1
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
crypto isakmp policy 1
  encryption aes
  authentication pre-share
```

```
  group 14
  lifetime 180
crypto isakmp key 0 1234 address 10.1.1.1
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-aes esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/0
 ip address 10.2.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
 crypto map testtag1
!
interface Ethernet1/1
 ip address 10.0.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
!
interface Ethernet1/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/4
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/5
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/6
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/7
 no ip address
 no ip route-cache
```

```
 no ip mroute-cache
 shutdown
 duplex half
!
interface Serial3/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial3/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.0.0.0 255.0.0.0 10.2.0.1
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.2.2 host 10.0.0.1
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
!
end
```

Additional References

Related Documents

Related Topic	Document Title
Configuring IKE	Configuring Internet Key Exchange for IPsec VPNs
Interface commands	Cisco IOS Master Command List
Recommended cryptographic algorithms	Next Generation Encryption

MIBs

MIBs	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

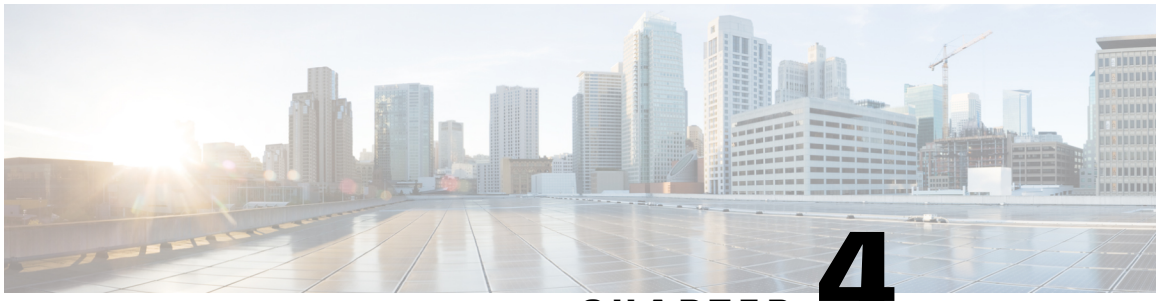
Feature Information for Invalid Security Parameter Index Recovery

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 Invalid Security Parameter Index Recovery

Feature Name	Releases	Feature Information
Invalid Security Parameter Index Recovery	12.3(2)T 12.2(18)SXE	When an invalid security parameter index error (shown as "Invalid SPI") occurs in IP Security (IPsec) packet processing, the Invalid Security Parameter Index Recovery feature allows for an Internet Key Exchange (IKE) security association (SA) to be established. The "IKE" module sends notification of the "Invalid SPI" error to the originating IPsec peer so that Security Association Databases (SADBs) can be resynchronized and successful packet processing can be resumed. This feature was introduced in Cisco IOS Release 12.3(2)T. This feature was integrated into Cisco IOS Release 12.2(18)SXE. The following command was introduced or modified: **crypto isakmp invalid-spi-recovery**



CHAPTER 4

IPsec Dead Peer Detection Periodic Message Option

The IPsec Dead Peer Detection Periodic Message Option feature is used to configure the router to query the liveliness of its Internet Key Exchange (IKE) peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers.

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.

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 at the end of this module.

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 IPsec Dead Peer Detection PeriodicMessage Option

Before configuring the IPsec Dead Peer Detection Periodic Message Option feature, you should have the following:

- Familiarity with configuring IP Security (IPsec).

Restrictions for IPsec Dead Peer Detection PeriodicMessage Option

Using periodic DPD potentially allows the router to detect an unresponsive IKE peer with better response time when compared to on-demand DPD. However, use of periodic DPD incurs extra overhead. When communicating to large numbers of IKE peers, you should consider using on-demand DPD instead.

Information About IPsec Dead Peer DetectionPeriodic Message Option

How DPD and Cisco IOS Keepalive Features Work

DPD and Cisco IOS keepalives function on the basis of the timer. If the timer is set for 10 seconds, the router sends a “hello” message every 10 seconds (unless, of course, the router receives a “hello” message from the peer). The benefit of IOS keepalives and periodic DPD is earlier detection of dead peers. However, IOS keepalives and periodic DPD rely on periodic messages that have to be sent with considerable frequency. The result of sending frequent messages is that the communicating peers must encrypt and decrypt more packets.

DPD also has an on-demand approach. The contrasting on-demand approach is the default. With on-demand DPD, messages are sent on the basis of traffic patterns. For example, if a router has to send outbound traffic and the liveliness of the peer is questionable, the router sends a DPD message to query the status of the peer. If a router has no traffic to send, it never sends a DPD message. If a peer is dead, and the router never has any traffic to send to the peer, the router does not discover this until the IKE or IPsec security association (SA) has to be rekeyed (the liveliness of the peer is unimportant if the router is not trying to communicate with the peer). On the other hand, if the router has traffic to send to the peer, and the peer does not respond, the router initiates a DPD message to determine the state of the peer.

Using the IPsec Dead Peer Detection Periodic Message Option

With the IPsec Dead Peer Detection Periodic Message Option feature, you can configure your router so that DPD messages are “forced” at regular intervals. This forced approach results in earlier detection of dead peers. For example, if a router has no traffic to send, a DPD message is still sent at regular intervals, and if a peer is dead, the router does not have to wait until the IKE SA times out to find out.

If you want to configure the DPD periodic message option, you should use the **crypto isakmp keepalive** command with the **periodic** keyword. If you do not configure the **periodic** keyword, the router defaults to the on-demand approach.

Note When the **crypto isakmp keepalive** command is configured, the Cisco IOS software negotiates the use of Cisco IOS keepalives or DPD, depending on which protocol the peer supports.

Using DPD and Cisco IOS Keepalive Featureswith Multiple Peers in the Crypto Map

DPD and IOS keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the router to detect a dead IKE peer, and when the router detects the dead state, the router deletes the IPsec and IKE SAs to the peer. If you configure multiple peers, the router switches over to the next listed peer for a stateless failover.

Using DPD in an Easy VPN Remote Configuration

DPD can be used in an Easy VPN remote configuration. See the section Configuring DPD for an Easy VPN Remote section.

How to Configure IPsec Dead Peer Detection PeriodicMessage Option

Configuring a Periodic DPD Message

To configure a periodic DPD message, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp keepalive** *seconds* [*retry-seconds*] [**periodic** | **on-demand**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp keepalive** *seconds* [*retry-seconds*] [**periodic** | **on-demand**] **Example:** `Router (config)# crypto isakmp keepalive 10 periodic`	Allows the gateway to send DPD messages to the peer. • *seconds* --When the **periodic** keyword is used, this argument is the number of seconds between DPD messages; the range is from 10 to 3600 seconds. When the **on-demand** keyword is used, this argument is the number of seconds during which traffic is not received from the peer before DPD retry messages are sent if there is data (IPSec) traffic to send; the range is from 10 to 3600 seconds. **Note** If you do not specify a time interval, an error message appears. • *retry-seconds* --(Optional) Number of seconds between DPD retry messages if the DPD retry message is missed by the peer; the range is from 2 to 60 seconds. Once 1 DPD message is missed by the peer, the router moves to a more aggressive state and sends the DPD retry message at the faster retry interval, which is the number of seconds between DPD retries if the DPD message is missed by the peer. The default DPD retry message is sent every 2 seconds. Five aggressive DPD retry messages can be missed before the tunnel is marked as down. **Note** To configure DPD with IPsec High Availability (HA), the recommendation is to use a value other than the default (which is 2 seconds). A keepalive timer of 10 seconds with 5 retries seems to work well with HA because of the time that it takes for the router to get into active mode. • **periodic** --(Optional) DPD messages are sent at regular intervals. • **on-demand** --(Optional) The default behavior. DPD retries are sent on demand. **Note** Because this option is the default, the **on-demand** keyword does not appear in configuration output.

Configuring DPD and Cisco IOS Keepalives with Multiple Peersin the Crypto Map

To configure DPD and IOS keepalives to be used in conjunction with the crypto map to allow for stateless failover, perform the following steps. This configuration causes a router to cycle through the peer list when it detects that the first peer is dead.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **set peer** {*host-name* [**dynamic**] | *ip-address*}
5. **set transform-set** *transform-set-name*
6. **match address** [*access-list-id* | *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router (config)# crypto map green 1 ipsec-isakmp`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • The **ipsec-isakmp** keyword indicates that IKE is used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 4	**set peer** {*host-name* [**dynamic**]	*ip-address*} **Example:** `Router (config-crypto-map)# set peer 10.12.12.12`	Specifies an IPsec peer in a crypto map entry. • You can specify multiple peers by repeating this command.

	Command or Action	Purpose
Step 5	**set transform-set** *transform-set-name* **Example:** `Router (config-crypto-map)# set transform-set txfm`	Specifies which transform sets can be used with the crypto map entry. • You can specify more than one transform set name by repeating this command.
Step 6	**match address** [*access-list-id* | *name*] **Example:** `Router (config-crypto-map)# match address 101`	Specifies an extended access list for a crypto map entry.

Configuring DPD for an Easy VPN Remote

To configure DPD in an Easy VPN remote configuration, perform the following steps. This configuration also causes a router to cycle through the peer list when it detects that the first peer is dead.

Note IOS keepalives are not supported for Easy VPN remote configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **connect** {**auto** | **manual**}
5. **group** *group-name* **key** *group-key*
6. **mode** {**client** | **network-extension**}
7. **peer** {*ipaddress* | *hostname*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ezvpn-config1`	Creates a Cisco Easy VPN remote configuration and enters the Cisco Easy VPN Remote configuration mode.
Step 4	**connect** {**auto** | **manual**} **Example:** `Router (config-crypto-ezvpn)# connect manual`	Manually establishes and terminates an IPsec VPN tunnel on demand. • The **auto** keyword option is the default setting.
Step 5	**group** *group-name* **key** *group-key* **Example:** `Router (config-crypto-ezvpn)# group unity key preshared`	Specifies the group name and key value for the Virtual Private Network (VPN) connection.
Step 6	**mode** {**client** | **network-extension**} **Example:** `Router (config-crypto-ezvpn)# mode client`	Specifies the VPN mode of operation of the router.
Step 7	**peer** {*ipaddress* | *hostname*} **Example:** `Router (config-crypto-ezvpn)# peer 10.10.10.10`	Sets the peer IP address or host name for the VPN connection. • A hostname can be specified only when the router has a DNS server available for host-name resolution. • This command can be repeated multiple times.

Verifying That DPD Is Enabled

DPD allows the router to clear the IKE state when a peer becomes unreachable. If DPD is enabled and the peer is unreachable for some time, you can use the **clear crypto session** command to manually clear IKE and IPsec SAs.

The **debug crypto isakmp** command can be used to verify that DPD is enabled.

SUMMARY STEPS

1. **enable**
2. **clear crypto session** [**local** *ip-address* [**port** *local-port*]] [**remote** *ip-address* [**port** *remote-port*]] | [**fvrf** *vrf-name*] [**ivrf** *vrf-name*]
3. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto session** [**local** *ip-address* [**port** *local-port*]] [**remote** *ip-address* [**port** *remote-port*]]	[**fvrf** *vrf-name*] [**ivrf** *vrf-name*] **Example:** `Router# clear crypto session`	Deletes crypto sessions (IPsec and IKE SAs).
Step 3	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Configuration Examples for IPsec Dead Peer DetectionPeriodic Message Option

Site-to-Site Setup with Periodic DPD Enabled Example

The following configurations are for a site-to-site setup with no periodic DPD enabled. The configurations are for the IKE Phase 1 policy and for the IKE preshared key.

IKE Phase 1 Policy

```
crypto isakmp policy 1
  encryption aes
  authentication pre-share
```

```
  group 14
!
```

IKE Preshared Key

```
crypto isakmp key kd94j1ksldz address 10.2.80.209 255.255.255.0
crypto isakmp keepalive 10 periodic
crypto ipsec transform-set Trans1 esp-aes esp-sha-hmac

!
!
interface
  ip address 10.1.32.14 255.255.255.0
  speed auto

!
```

Easy VPN Remote with DPD Enabled Example

The following configuration tells the router to send a periodic DPD message every 30 seconds. If the peer fails to respond to the DPD R_U_THERE message, the router resends the message every 20 seconds (four transmissions altogether).

```
crypto isakmp keepalive 30 20 periodic
crypto ipsec client ezvpn ezvpn-config
  connect auto
  group unity key preshared
  mode client
  peer 10.2.80.209
!
!
interface Ethernet0
  ip address 10.2.3.4 255.255.255.0
  half-duplex
  crypto ipsec client ezvpn ezvpn-config inside
!
interface FastEthernet0
  ip address 10.1.32.14 255.255.255.0
  speed auto
  crypto ipsec client ezvpn ezvpn-config outside
```

Verifying DPD Configuration Using the debug crypto isakmp Command Example

The following sample output from the **debug crypto isakmp** command verifies that IKE DPD is enabled:

```
*Mar 25 15:17:14.131: ISAKMP:(0:1:HW:2):IKE_DPD is enabled, initializing timers
```

To see that IKE DPD is enabled (and that the peer supports DPD): when periodic DPD is enabled, you should see the following debug messages at the interval specified by the command:

```
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2):purging node 899852982 *Mar 25 15:18:52.111:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:18:52.111: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to sending the DPD R_U_THERE message.

```
*Mar 25 15:18:52.123: ISAKMP (0:268435457): received packet from 10.2.80.209
dport 500 sport 500 Global (I) QM_IDLE
```

```
*Mar 25 15:18:52.123: ISAKMP: set new node -443923643 to QM_IDLE *Mar 25 15:18:52.131:
ISAKMP:(0:1:HW:2): processing HASH payload. message ID =
-443923643
*Mar 25 15:18:52.131: ISAKMP:(0:1:HW:2): processing NOTIFY R_U_THERE_ACK protocol 1
 spi 0, message ID = -443923643, sa = 81BA4DD4
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2): DPD/R_U_THERE_ACK received from peer
10.2.80.209, sequence 0x9
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):deleting node -443923643 error FALSE
reason "informational (in) state 1"
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_INFO_NOTIFY *Mar
25 15:18:52.135: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to receiving the acknowledge (ACK) message from the peer.

```
Router#
*Mar 25 15:47:35.335: ISAKMP: set new node -90798077 to QM_IDLE *Mar 25 15:47:35.343:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:35.343: ISAKMP:(0:1:HW:2):purging node -90798077 *Mar 25 15:47:35.347:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:47:35.347: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:36.611: ISAKMP:(0:1:HW:2):purging node 1515050537 *Mar 25 15:47:37.343:
ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:37.343: ISAKMP: set new node -1592471565 to QM_IDLE *Mar 25 15:47:37.351:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:37.351: ISAKMP:(0:1:HW:2):purging node -1592471565 *Mar 25 15:47:37.355:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:37.355: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:39.355: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:39.355: ISAKMP: set new node 1758739401 to QM_IDLE *Mar 25 15:47:39.363:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:39.363: ISAKMP:(0:1:HW:2):purging node 1758739401 *Mar 25 15:47:39.367:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:39.367: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:41.367: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:41.367: ISAKMP: set new node 320258858 to QM_IDLE *Mar 25 15:47:41.375:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):purging node 320258858 *Mar 25 15:47:41.379:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:43.379: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:43.379: ISAKMP: set new node -744493014 to QM_IDLE *Mar 25 15:47:43.387:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:43.387: ISAKMP:(0:1:HW:2):purging node -744493014 *Mar 25 15:47:43.391:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:43.391: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):peer 10.2.80.209 not responding! *Mar 25 15:47:45.391:
 ISAKMP:(0:1:HW:2):peer does not do paranoid keepalives.
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.395: ISAKMP: Unlocking IPSEC struct 0x81E5C4E8 from
delete_siblings, count 0
```

```
*Mar 25 15:47:45.395: %CRYPTO-5-SESSION_STATUS: Crypto tunnel is DOWN.  Peer
10.2.80.209:500       Id: 10.2.80.209
*Mar 25 15:47:45.399: ISAKMP: set new node -2061951065 to QM_IDLE *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):purging node -2061951065 *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_DEST_SA
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.415: ISAKMP: Unlocking IKE struct 0x81E5C4E8 for
isadb_mark_sa_deleted(), count 0
*Mar 25 15:47:45.415: ISAKMP: Deleting peer node by peer_reap for 10.2.80.209:
81E5C4E8
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -1067612752 error TRUE
reason "peers alive"
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -114443536 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node 2116015069 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node -1981865558 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):Input = IKE_MESG_INTERNAL, IKE_PHASE1_DEL *Mar 25
15:47:45.419: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA
*Mar 25 15:47:45.419: ISAKMP: received ke message (4/1)
*Mar 25 15:47:45.419: ISAKMP: received ke message (3/1)
*Mar 25 15:47:45.423: ISAKMP: ignoring request to send delete notify (no ISAKMP
sa) src 10.1.32.14 dst 10.2.80.209 for SPI 0x3A7B69BF
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting SA reason "" state (I)
MM_NO_STATE (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -1067612752 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -114443536 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node 2116015069 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):deleting node -1981865558 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH *Mar 25
15:47:45.427: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA
```

The above message shows what happens when the remote peer is unreachable. The router sends one DPD R_U_THERE message and four retransmissions before it finally deletes the IPsec and IKE SAs.

DPD and Cisco IOS Keepalives Used in Conjunction with Multiple Peers in a Crypto Map Example

The following example shows that DPD and Cisco IOS keepalives are used in conjunction with multiple peers in a crypto map configuration when IKE is used to establish the security associations (SAs). In this example, an SA could be set up to the IPsec peer at 10.0.0.1, 10.0.0.2, or 10.0.0.3.

```
crypto map green 1 ipsec-isakmp
  set peer 10.0.0.1
  set peer 10.0.0.2
  set peer 10.0.0.3
  set transform-set txfm
  match address 101
```

DPD Used in Conjunction with Multiple Peers for an Easy VPN Remote Example

The following example shows that DPD is used in conjunction with multiple peers in an Easy VPN remote configuration. In this example, an SA could be set up to the IPsec peer at 10.10.10.10, 10.2.2.2, or 10.3.3.3.

```
crypto ipsec client ezvpn ezvpn-config
  connect auto
  group unity key preshared
  mode client
  peer 10.10.10.10
  peer 10.2.2.2
  peer 10.3.3.3
```

Additional References

Related Documents

Related Topic	Document Title
Configuring IPsec	Configuring Security for VPNs with IPsec
IPsec commands	Cisco IOS Security Command Reference

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
DPD conforms to the Internet draft “draft-ietf-ipsec-dpd-04.txt,” which is pending publication as an Informational RFC (a number has not yet been assigned).	--

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

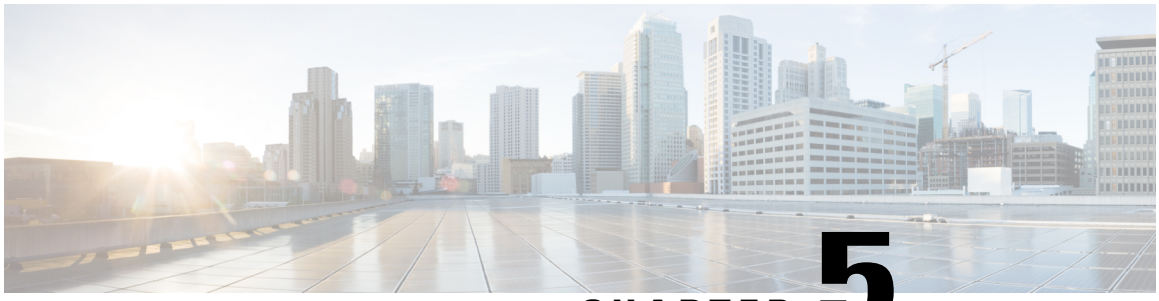
Feature Information for IPsec Dead Peer Detection Periodic Message Option

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 5: Feature Information for IPsec Dead Peer Detection Periodic Message Option

Feature Name	Releases	Feature Information
IPsec Dead Peer Detection Periodic Message Option	12.3(7)T 12.2(33)SRA 12.2(33)SXH	The IPsec Dead Peer Detection Periodic Message Option feature is used to configure the router to query the liveliness of its Internet Key Exchange (IKE) peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers. This feature was introduced in Cisco IOS Release 12.3(7)T. This feature was integrated into Cisco IOS Release 12.2(33)SRA This feature was integrated into Cisco IOS Release 12.2(33)SXH The following command was introduced: **crypto isakmp keepalive**.



CHAPTER 5

IPsec NAT Transparency

The IPsec NAT Transparency feature introduces support for IP Security (IPsec) traffic to travel through Network Address Translation (NAT) or Port Address Translation (PAT) points in the network by addressing many known incompatibilities between NAT and IPsec.

Before the introduction of this feature, a standard IPsec virtual private network (VPN) tunnel would not work if there were one or more NAT or PAT points in the delivery path of the IPsec packet. This feature makes NAT IPsec-aware, thereby, allowing remote access users to build IPsec tunnels to home gateways.

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.

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 at the end of this module.

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 NAT Transparency

Although this feature addresses many incompatibilities between NAT and IPsec, the following problems still exist:

Internet Key Exchange (IKE) IP Address and NAT

This incompatibility applies only when IP addresses are used as a search key to find a preshared key. Modification of the IP source or destination addresses by NAT or reverse NAT results in a mismatch between the IP address and the preshared key.

Embedded IP Addresses and NAT

Because the payload is integrity protected, any IP address enclosed within IPsec packets cannot be translated by NAT. Protocols that use embedded IP addresses include FTP, Internet Relay Chat (IRC), Simple Network Management Protocol (SNMP), Lightweight Directory Access Protocol (LDAP), H.323, and Session Initiation Protocol (SIP).

Information About IPsec NAT Transparency

Feature Design of IPsec NAT Traversal

The IPsec NAT Transparency feature introduces support for IPsec traffic to travel through NAT or PAT points in the network by encapsulating IPsec packets in a User Datagram Protocol (UDP) wrapper, which allows the packets to travel across NAT devices. The following sections define the details of NAT traversal:

- IKE Phase 1 Negotiation NAT Detection, on page 48
- IKE Phase 2 Negotiation NAT Traversal Decision, on page 49
- UDP Encapsulation of IPsec Packets for NAT Traversal, on page 49
- UDP Encapsulated Process for Software Engines Transport Mode and Tunnel Mode ESP Encapsulation, on page 51

IKE Phase 1 Negotiation NAT Detection

During Internet Key Exchange (IKE) phase 1 negotiation, two types of NAT detection occur before IKE Quick Mode begins--NAT support and NAT existence along the network path.

To detect NAT support, you should exchange the vendor identification (ID) string with the remote peer. During Main Mode (MM) 1 and MM 2 of IKE phase 1, the remote peer sends a vendor ID string payload to its peer to indicate that this version supports NAT traversal. Thereafter, NAT existence along the network path can be determined.

Detecting whether NAT exists along the network path allows you to find any NAT device between two peers and the exact location of NAT. A NAT device can translate the private IP address and port to public value (or from public to private). This translation changes the IP address and port if the packet goes through the device. To detect whether a NAT device exists along the network path, the peers should send a payload with

hashes of the IP address and port of both the source and destination address from each end. If both ends calculate the hashes and the hashes match, each peer knows that a NAT device does not exist on the network path between them. If the hashes do not match (that is, someone translated the address or port), then each peer needs to perform NAT traversal to get the IPsec packet through the network.

The hashes are sent as a series of NAT discovery (NAT-D) payloads. Each payload contains one hash; if multiple hashes exist, multiple NAT-D payloads are sent. In most environments, there are only two NAT-D payloads--one for the source address and port and one for the destination address and port. The destination NAT-D payload is sent first, followed by the source NAT-D payload, which implies that the receiver should expect to process the local NAT-D payload first and the remote NAT-D payload second. The NAT-D payloads are included in the third and fourth messages in Main Mode and in the second and third messages in Aggressive Mode (AM).

IKE Phase 2 Negotiation NAT Traversal Decision

While IKE phase 1 detects NAT support and NAT existence along the network path, IKE phase 2 decides whether or not the peers at both ends will use NAT traversal. Quick Mode (QM) security association (SA) payload in QM1 and QM2 is used to for NAT traversal negotiation.

Because the NAT device changes the IP address and port number, incompatibilities between NAT and IPsec can be created. Thus, exchanging the original source address bypasses any incompatibilities.

UDP Encapsulation of IPsec Packets for NAT Traversal

In addition to allowing IPsec packets to traverse across NAT devices, UDP encapsulation also addresses many incompatibility issues between IPsec and NAT and PAT. The resolved issues are as follows:

Incompatibility Between IPsec ESP and PAT--Resolved

If PAT found a legislative IP address and port, it would drop the Encapsulating Security Payload (ESP) packet. To prevent this scenario, UDP encapsulation is used to hide the ESP packet behind the UDP header. Thus, PAT treats the ESP packet as a UDP packet, processing the ESP packet as a normal UDP packet.

Incompatibility Between Checksums and NAT--Resolved

In the new UDP header, the checksum value is always assigned to zero. This value prevents an intermediate device from validating the checksum against the packet checksum, thereby, resolving the TCP UDP checksum issue because NAT changes the IP source and destination addresses.

Incompatibility Between Fixed IKE Destination Ports and PAT--Resolved

PAT changes the port address in the new UDP header for translation and leaves the original payload unchanged.

To see how UDP encapsulation helps to send IPSec packets see the figures below.

Figure 2: Standard IPsec Tunnel Through a NAT/PAT Point (No UDP Encapsulation)

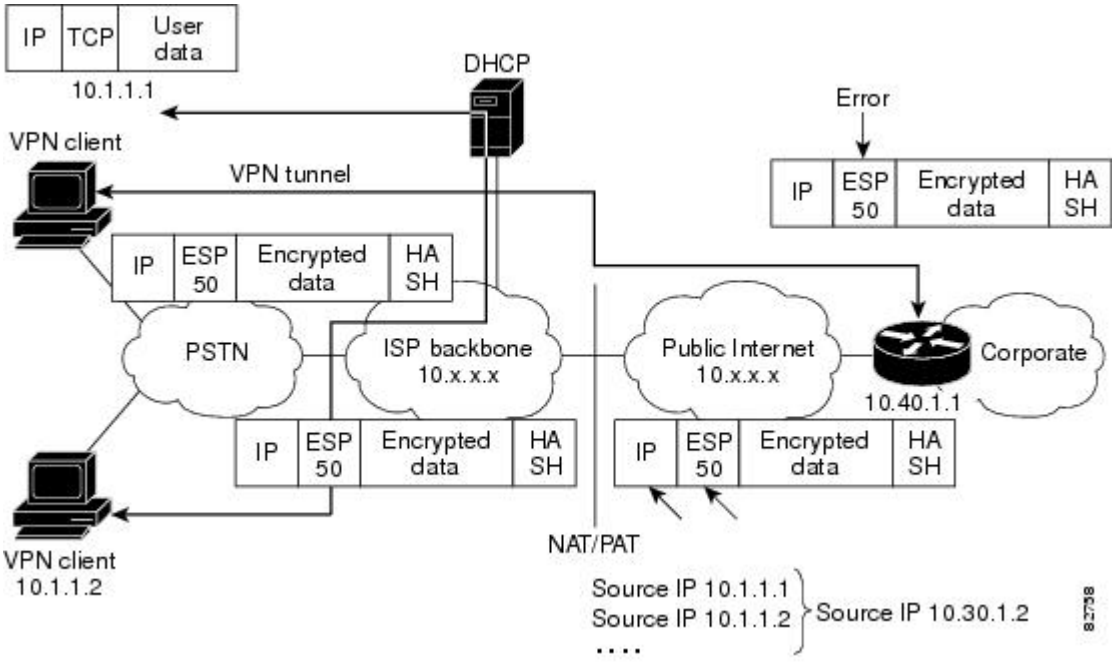
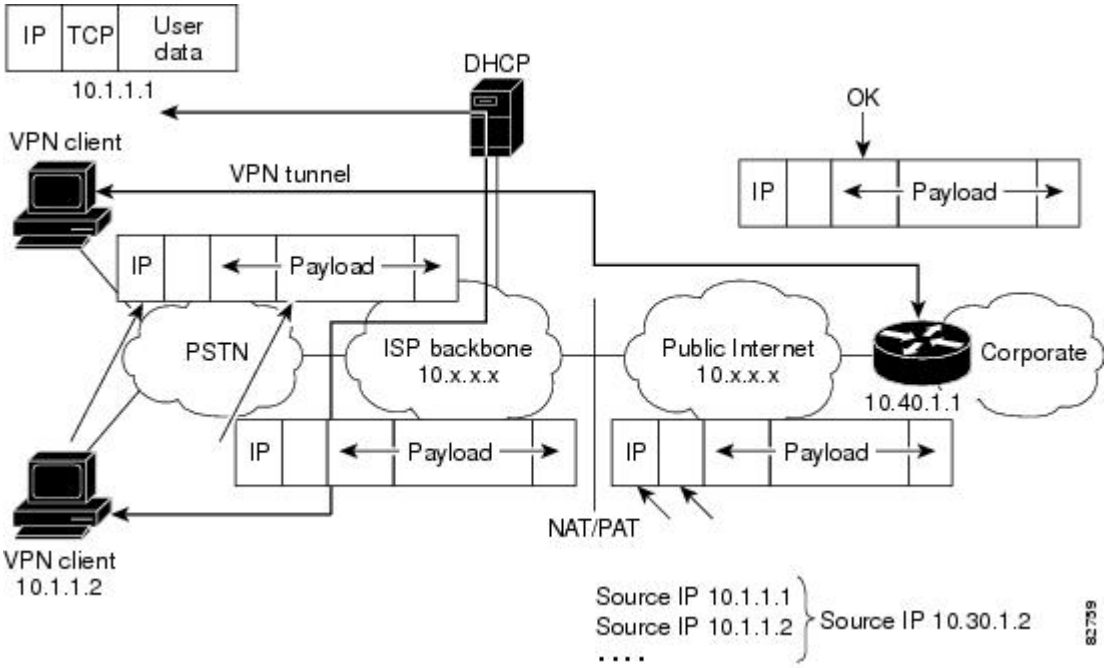


Figure 3: IPsec Packet with UDP Encapsulation



UDP Encapsulated Process for Software Engines Transport Mode and Tunnel Mode ESP Encapsulation

After the IPsec packet is encrypted by a hardware accelerator or a software crypto engine, a UDP header and a non-ESP marker (which is 4 bytes in length) are inserted between the original IP header and ESP header. The total length, protocol, and checksum fields are changed to match this modification.

NAT Keepalives

NAT keepalives are enabled to keep the dynamic NAT mapping alive during a connection between two peers. NAT keepalives are UDP packets with an unencrypted payload of 1 byte. Although the current dead peer detection (DPD) implementation is similar to NAT keepalives, there is a slight difference: DPD is used to detect peer status, while NAT keepalives are sent if the IPsec entity did not send or receive the packet at a specified period of time--valid range is between 5 to 3600 seconds.

If NAT keepalives are enabled (through the **crypto isamkp nat keepalive** command), users should ensure that the idle value is shorter than the NAT mapping expiration time, which is 20 seconds.

How to Configure NAT and IPsec

Configuring NAT Traversal

NAT Traversal is a feature that is auto detected by VPN devices. There are no configuration steps for a router running Cisco IOS Release 12.2(13)T. If both VPN devices are NAT-T capable, NAT Traversal is auto detected and auto negotiated.

Disabling NAT Traversal

You may wish to disable NAT traversal if you already know that your network uses IPsec-awareness NAT (spi-matching scheme). To disable NAT traversal, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto ipsec nat-transparency udp-encapsulation**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**no crypto ipsec nat-transparency udp-encapsulation**
	
	Example:
	`Router(config)#`
	`no crypto ipsec nat-transparency udp-encapsulation`
		Disables NAT traversal.

Configuring NAT Keepalives

To configure your router to send NAT keepalives, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp nat keepalive** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp nat keepalive** *seconds* **Example:** `Router(config)# crypto isakmp nat keepalive 20`	Allows an IPsec node to send NAT keepalive packets. • *seconds* --The number of seconds between keepalive packets; range is from 5 to 3,600. **Note** When the timer is modified, it is modified for every Internet Security Association Key Management Protocol (ISAKMP) security association (SA) when the keepalive for that SA is sent based on the existing timer. **Note** A five-percent jitter mechanism value is applied to the timer to avoid security association rekey collisions. If there are many peer routers, and the timer is configured too low, then the router can experience high CPU usage.

Verifying IPsec Configuration

To verify your configuration, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**show crypto ipsec sa** [**map** *map-name*	**address**	**identity**] [**detail** **Example:** `Router# show crypto ipsec sa`	Displays the settings used by current SAs.

Configuration Examples for IPsec and NAT

NAT Keepalives Configuration Example

The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
crypto isakmp key 1234 address 56.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-aes esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 56.0.0.1
 set transform-set t2
 match address 101
```

Additional References

Related Documents

Related Topic	Document Title
Additional NAT configuration tasks.	• Configuring NAT for IP Address Conservation • Using Application Level Gateways with NAT • Configuring NAT for High Availability • Configuring Hosted NAT Traversal for Session Border Controller • Integrating NAT with MPLS VPNs • Scalability for Stateful NAT • NAT - Optimized SIP Media Path with SDP
Additional NAT commands	Cisco IOS IP Addressing Services Command Reference
Additional IPsec configuration tasks	Configuring Security for VPNs with IPsec
Additional IPsec commands	Cisco IOS Security Command Reference

Related Topic	Document Title
Information on IKE	Configuring Internet Key Exchange for IPsec VPNs
Additional information on IKE dead peer detection.	Easy VPN Server
Recommended cryptographic algorithms	Next Generation Encryption

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs[1]	Title
RFC 2402	IP Authentication Header
RFC 2406	IP Encapsulating Security Payload (ESP)
RFC 3947	Negotiation of NAT-Traversal in the IKE

1 Not all supported RFCs are listed.

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec NAT Transparency

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 IPsec NAT Transparency

Feature Name	Releases	Feature Information
IPsec NAT Transparency	12.2(13)T	The IPsec NAT Transparency feature introduces support for IP Security (IPsec) traffic to travel through Network Address Translation (NAT) or Port Address Translation (PAT) points in the network by addressing many known incompatibilities between NAT and IPsec. In 12.2(13)T, this feature was introduced on the Cisco IOS software. The following commands were introduced or modified: **crypto isamkp nat keepalive**, **access-list (IP extended) , show crypto ipsec sa**.

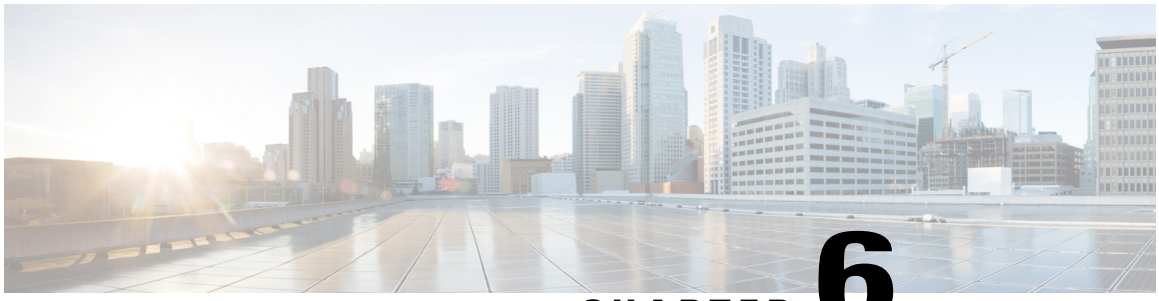
Glossary

IKE --Internet Key Exchange. Hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. Although IKE can be used with other protocols, its initial implementation is with IPsec. IKE provides authentication of the IPsec peers, negotiates IPsec keys, and negotiates IPsec security associations (SAs).

IPsec --IP Security. Framework of open standards developed by the Internet Engineering Task Force (IETF). IPsec provides security for 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.

NAT --Network Address Translation. Translates a private IP address used inside the corporation to a public, routable address for use on the outside of the corporation, such as the Internet. NAT is considered a one-to-one mapping of addresses from private to public.

PAT --Port Address Translation. Like NAT, PAT also translated private IP address to public, routable addresses. Unlike NAT, PAT provides a many-to-one mapping of private addresses to a public address; each instance of the public address is associated with a particular port number to provide uniqueness. PAT can be used in environments where the cost of obtaining a range of public addresses is too expensive for an organization.



CHAPTER 6

DF Bit Override Functionality with IPsec Tunnels

The DF Bit Override Functionality with IPsec Tunnels feature allows customers to configure the setting of the DF bit when encapsulating tunnel mode IPsec traffic on a global or per-interface level. Thus, if the DF bit is set to clear, routers can fragment packets regardless of the original DF bit setting.

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.

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 at the end of this module.

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 DF Bit Override Functionality with IPsec Tunnels

IPsec must be enabled on your router.

Restrictions for DF Bit Override Functionality with IPsec Tunnels

Performance Impact

Because each packet is reassembled at the process level, a significant performance impact occurs at a high data rate. Two major caveats are as follows:

- The reassemble queue can fill up and force fragments to be dropped.
- The traffic is slower because of the process switching.

DF Bit Setting Requirement

If several interfaces share the same crypto map using the local address feature, these interfaces must share the same DF bit setting.

Feature Availability

This feature is available only for IPsec tunnel mode. (IPsec transport mode is not affected because it does not provide an encapsulating IP header.)

Information About DF Bit Override Functionality with IPsec Tunnels

The DF Bit Override Functionality with IPsec Tunnels feature allows customers to specify whether their router can clear, set, or copy the Don't Fragment (DF) bit from the encapsulated header. A DF bit is a bit within the IP header that determines whether a router is allowed to fragment a packet.

Some customer configurations have hosts that perform the following functions:

- Set the DF bit in packets they send
- Use firewalls that block Internet Control Message Protocol (ICMP) errors from outside the firewall, preventing hosts from learning about the maximum transmission unit (MTU) size outside the firewall
- Use IP Security (IPsec) to encapsulate packets, reducing the available MTU size

Customers whose configurations have hosts that prevent them from learning about their available MTU size can configure their router to clear the DF bit and fragment the packet.

Note In compliance with RFC 2401, this feature can be configured globally or per interface. If both levels are configured, the interface configuration will override the global configuration.

How to Configure DF Bit Override Functionality with IPsec Tunnels

Configuring the DF Bit for the Encapsulating Header in Tunnel Mode

The following task sets the DF bit for the encapsulating header in tunnel mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec df-bit** [**clear** | **set** | **copy**]
4. **interface** *type number*
5. **crypto ipsec df-bit** [**clear** | **set** | **copy**]
6. **exit**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec df-bit** [**clear**	**set**	**copy**] **Example:** `Router(config)# crypto ipsec df-bit set`	Sets the DF bit for the encapsulating header in tunnel mode for all interfaces. • The **clear** keyword clears the DF bit in the outer IP header, and the router may fragment the packet to add the IP Security (IPSec) encapsulation.

	Command or Action	Purpose
		• The **set** keyword sets the DF bit in the outer IP header, however, the router may fragment the packet if the original packet had the DF bit cleared. • The **copy** keyword has the router look in the original packet for the outer DF bit setting. The copy keyword is the default setting.
Step 4	**interface** *type number* **Example:** `Router(config-if)# interface Ethernet0/0`	Specifies the interface on which the DF bit is configured and enters interface configuration mode.
Step 5	**crypto ipsec df-bit** [**clear** | **set** | **copy**] **Example:** `Router(config-if)# crypto ipsec df-bit clear`	(Optional) Sets the DF bit for a specified interface, **Note** DF bit interface configuration settings override all DF bit global configuration settings.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits interface configuration mode and enters EXEC mode.
Step 7	**show running-config** **Example:** `Router# show running-config`	Verifies the current DF Bit settings on your router.

Configuration Example for DF Bit Override Functionality with IPsec Tunnels

DF Bit Setting Configuration Example

In following example, the router is configured to globally clear the setting for the DF bit and copy the DF bit on the interface named Ethernet0. Thus, all interfaces except Ethernet0 will allow the router to send packets larger than the available MTU size; Ethernet0 will allow the router to fragment the packet.

```
crypto isakmp policy 1
  encryption aes
  hash sha
  authentication pre-share
  group 14
crypto isakmp key Delaware address 192.168.10.66
crypto isakmp key Key-What-Key address 192.168.11.19
```

```
!
!
crypto ipsec transform-set BearMama ah-sha-hmac esp-aes
crypto ipsec df-bit clear
!
!
crypto map armadillo 1 ipsec-isakmp
set peer 192.168.10.66
set transform-set BearMama
match address 101
!
crypto map basilisk 1 ipsec-isakmp
set peer 192.168.11.19
set transform-set BearMama
match address 102
!
!
interface Ethernet0
   ip address 192.168.10.38 255.255.255.0
   ip broadcast-address 0.0.0.0
   media-type 10BaseT
   crypto map armadillo
   crypto ipsec df-bit copy
!
interface Ethernet1
   ip address 192.168.11.75 255.255.255.0
   ip broadcast-address 0.0.0.0
   media-type 10BaseT
   crypto map basilisk
!
interface Serial0
   no ip address
   ip broadcast-address 0.0.0.0
   no ip route-cache
   no ip mroute-cache
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	Cisco IOS Security Command Reference
Recommended cryptographic algorithms	Next Generation Encryption

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2401	Security Architecture for the Internet Protocol

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

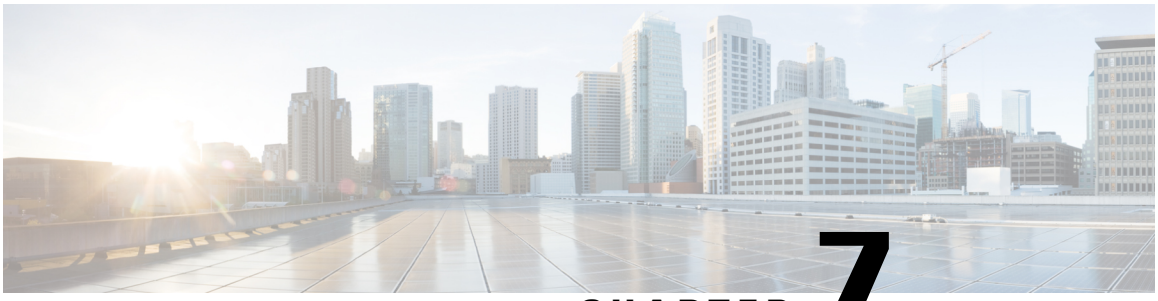
Feature Information for DF Bit Override Functionality with IPsec Tunnels

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 7: Feature Information for DF Bit Override Functionality with IPsec Tunnels

Feature Name	Releases	Feature Information
DF Bit Override Functionality with IPSec Tunnels	12.2(11)T	The DF Bit Override Functionality with IPsec Tunnels feature allows customers to configure the setting of the DF bit when encapsulating tunnel mode IPsec traffic on a global or per-interface level. Thus, if the DF bit is set to clear, routers can fragment packets regardless of the original DF bit setting. This feature was introduced in Cisco IOS Release 12.2(11)T. The following commands were introduced or modified: **crypto ipsec df-bit (global configuration), crypto ipsec df-bit (interface configuration) .**



CHAPTER 7

Crypto Access Check on Clear-Text Packets

The Crypto Access Check on Clear-Text Packets feature removes the checking of clear-text packets that go through the IP Security (IPSec) tunnel just prior to encryption or just after decryption. The clear-text packets were checked against the outside physical interface access control lists (ACLs). This checking was often referred to as a double ACL check. This feature enables easier configuration of ACLs and eliminates the security risks that are associated with a double check when using dynamic crypto maps.

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.

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 at the end of this module.

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 Access Check on Clear-Text Packets

- You should be familiar with configuring IPSec.
- You should be familiar with ACLs.

Restrictions for Crypto Access Check on Clear-Text Packets

- This feature does not apply to IPSec configurations on the Virtual Private Network (VPN) service module (card) on Cisco Catalyst 6500 series switches and Cisco 7600 series router platforms.
- This feature supports only extended ACLs.

Information About Crypto Access Check on Clear-Text Packets

Crypto Access Check on Clear-Text Packets Overview

The Crypto Access Check on Clear-Text Packets feature provides four changes for the interaction between IPSec and interface access lists. The changes are as follows:

- Removes the checking of inbound, just-decrypted clear-text packets against the outside interface inbound ACL.
- Removes the checking of outbound clear-text packets just prior to encryption against the outside interface outbound ACL.
- Adds the checking of outbound encrypted packets against the outside interface outbound ACL.
- Adds the capability to configure ACLs under the crypto map to check inbound clear-text packets after decryption or outbound clear-text packets prior to encryption.

This feature enables the easier and more consistent configuration of ACLs that control packet movement in and out of the outside interface as well as in and out of the IPSec encryption tunnel. This feature also eliminates security risks that are associated with the current double check when using dynamic crypto maps.

Configuration Changes That Are Required for This Feature

This feature requires the following configuration changes to be performed. Some are required and some are optional.

Prior to Upgrading

Prior to upgrading to this feature, you should do the following. This change is required.

Check all outside interfaces for outbound ACLs. If any outbound ACLs exist, check to ensure that they include access-list entries (ACEs) that permit outbound Encapsulating Security Payload (ESP) IP protocol 50 packets

or Authentication Header (AH) IP protocol 51 packets. The ACL entries will be needed after the upgrade because the outbound encrypted packets will be checked against the outside interface outbound ACL. If the ESP or AH packets are not allowed by the outside interface outbound ACL, the IPSec VPN tunnels will not forward traffic.

After Upgrading

After upgrading to this feature, you should do the following. The first two procedures are required if you are using dynamic crypto maps. However, these procedures are recommended even if you are not using dynamic crypto maps. The third and fourth procedures are optional.

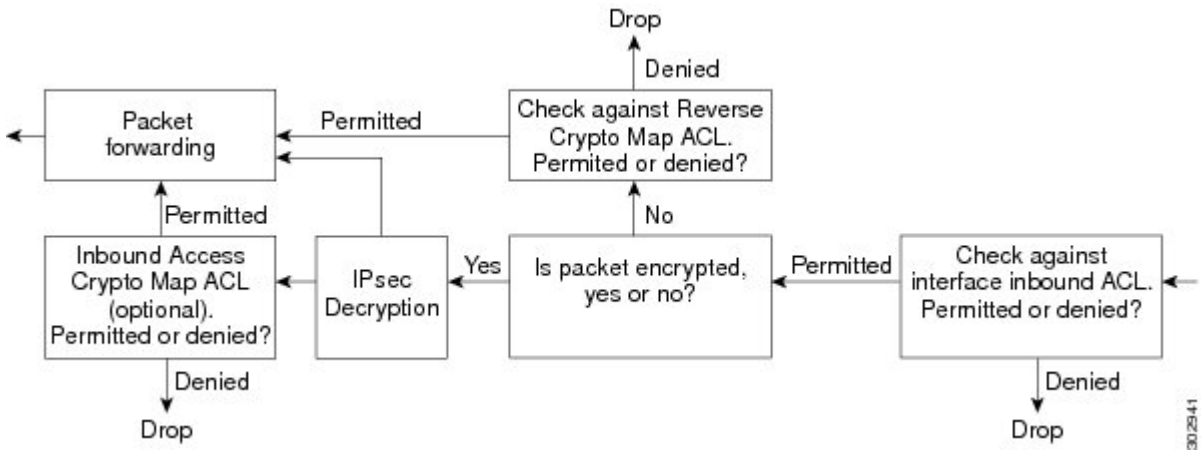
- Check all outside interfaces for inbound ACLs that contain ACEs that permit inbound, just-decrypted clear-text packets. These ACEs need to be removed if dynamic crypto maps are being used because when the IPSec tunnel is not "up," the ACEs will allow the clear-text packets into the network. If dynamic crypto maps are not being used, the ACEs can still be removed to simplify the outside interface ACLs.
- Check all outside interfaces for outbound ACLs that contain ACEs that permit outbound clear-text packets that would be encrypted. These ACEs need to be removed if dynamic crypto maps are being used because when the IPSec tunnel is not up, these ACEs will allow the clear-text packets out of the network. If dynamic crypto maps are not being used, these ACEs can still be removed to simplify the outside interface ACLs.
- Add an outbound crypto map access ACL under the crypto map to deny to-be-encrypted, outbound clear-text packets that should be dropped. Be sure that you also permit all other packets in this ACL.
- Add an inbound crypto map access ACL under the crypto map to deny just-decrypted, inbound clear-text packets that should be dropped. Be sure to also permit all other packets in this ACL.

The last two configuration changes are needed only in the rare cases in which the crypto map ACL (that selects packets to be encrypted) is more general than the packet flows that you want to encrypt. Adding outbound or inbound crypto map ACLs is usually done to keep the crypto map ACL small and simple, which saves CPU utilization and memory. The **set ip access-group** command, which is used to cause the checking of clear-text packets after decryption and before encryption, can be used under the crypto map to accomplish this task independent of the outside interface ACLs.

ACL Checking Behavior After Upgrading to This Feature

The diagam below illustrates the ACL checking behavior on the inbound path using the Crypto Access Check on Clear-Text Packets feature.

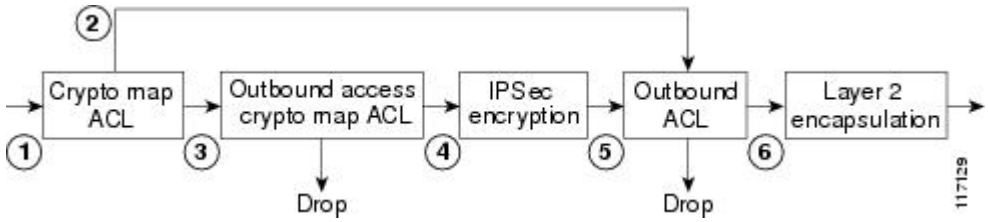
Figure 4: New Inbound Encrypted Packet Flow



1. Arriving IP packet is checked against the interface inbound ACL. If it is denied, it is dropped.
2. If IP packet is a permitted not-encrypted packet, it is forwarded and checked against the reverse crypto map ACL. If the result of the reverse crypto map ACL is permit, then the packet is dropped.
3. If IP packet is encrypted, it is then decrypted.
4. Just-decrypted IP packet is forwarded.
5. Just-decrypted IP packet is checked against the inbound access crypto map ACL (optional). If the packet is denied, it is dropped.

The diagram below illustrates the ACL checking behavior on the outbound path using the Crypto Access Check on Clear-Text Packets feature.

Figure 5: New Outbound Encrypted Packet Flow



1. All departing IP packets are checked against the crypto map ACL. If the packets are permitted, they are marked for encryption.
2. IP packets not marked for encryption are checked against the outbound interface ACL. If the packets are denied, they are dropped.

3 IP packets marked for encryption are checked against the outbound access crypto map ACL (optional). If the packets are denied, they are dropped.

4 Permitted IP packets are encrypted.

5 Encrypted IP packets are checked against the outbound interface ACL. If the packets are denied, they are dropped.

6 Permitted IP packets are Layer 2 encapsulated.

Backward Compatibility

If the Cisco IOS software is subsequently downgraded to a release that does not have the Crypto Access Check on Clear-Text Packets feature, the just-decrypted and to-be-encrypted clear-text packets will again be blocked by the outside interface ACLs. Therefore, if you have removed lines from the interface ACLs, you should undo the changes that were made to the ACLs if you are downgrading to an earlier version.

How to Configure Crypto Map Access ACLs

Adding or Removing ACLs

To add or remove crypto map access ACLs, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* seq-*number*
4. **set ip access-group** {*access-list-number* |*access-list-name*} {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto map** *map-name* seq-*number* **Example:** `Router(config)# crypto map vpn1 10`	Selects the crypto map and the sequence map entry under the crypto map to which you want to add the crypto map access ACL; also enters crypto map configuration mode.
Step 4	**set ip access-group** {*access-list-number* |*access-list-name*} {**in** | **out**} **Example:** `Router(config-crypto-map)# set ip access-group 151 in`	Allows you to check the postdecrypted or preencrypted packet against an ACL without having to use the outside physical interface ACL.

Verifying the Configured ACLs

The **show ip access-list** command can be used to verify the crypto input or output access-check ACLs that have been configured. Also, the packets that have been dropped in the context of the crypto input access-check ACL in the inbound path will be logged as receive (recv) errors, and packets dropped on the outbound path will be logged as send errors.

The **show crypto map** command can be used to verify crypto map configuration information.

SUMMARY STEPS

1. **enable**
2. show ip access-list [*access-list-number* | *access-list-name* | **dynamic**]
3. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show ip access-list [*access-list-number* | *access-list-name* | **dynamic**] **Example:** `Router# show ip access-list Internetfilter`	Displays a configured ACL.

	Command or Action	Purpose
Step 3	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays the crypto maps that have been configured.

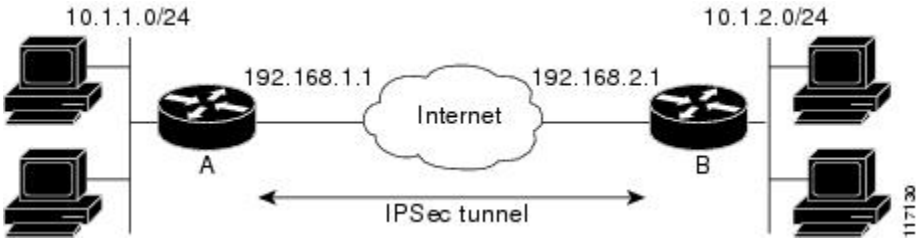
Configuration Examples for Crypto Access Checkon Clear-Text Packets

This section contains the output for the following stages of crypto access configuration:

- Previous IPSec ACL Configuration Example, on page 73
- New IPSec ACL Configuration Without Crypto Access ACLs Example, on page 74
- New IPSec ACL Configuration with Crypto Access ACLs Example, on page 74
- Authentication Proxy IPSec and CBAC Configuration Example, on page 75

The network diagram used for the following examples is shown below.

Figure 6: Network Diagram for Crypto Access Check Configuration Examples



The configuration examples assume these policy rules:

- Allow only encrypted host traffic between hosts on 10.1.1.0/24 and 10.1.2.0/24.
- No clear-text traffic from the Internet to any host.

Previous IPSec ACL Configuration Example

The following is a sample configuration using an earlier version of Cisco IOS software (before Release 12.3(8)T). The configuration shows outside interface ACLs with a double check on the inbound packets.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
```

```
 match address 101
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
 ip access-group 160 out
 crypto map vpnmap
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
access-list 150 permit ip 10.1.2.0 0.0.0.255 10.1.1.0 0.0.0.255
access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
```

New IPSec ACL Configuration Without Crypto Access ACLs Example

The following is a sample configuration using the current version of Cisco IOS software (Release 12.3(8)T). Before the crypto map access ACL is added, clear-text packets through the IPSec tunnel are not checked against an ACL (other packets are checked against the outside interface ACLs). Note the permitting of ESP packets in the outside interface outbound ACL.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
 match address 101
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
 ip access-group 160 out
 crypto map vpnmap
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit esp host 192.168.1.1 host 192.168.2.1
```

New IPSec ACL Configuration with Crypto Access ACLs Example

The following is a sample configuration using the current version of Cisco IOS software (Release 12.3(8)T). Before a crypto map access ACL is added, clear-text packets through the IPSec tunnel are checked against the crypto map access ACLs (other packets are checked against the outside interface ACLs).

Note In the following example, all IP packets between the subnets 10.1.1.0/24 and 10.1.2.0/24 are to be encrypted, but the crypto map access ACLs allow only Telnet traffic through the IPSec tunnel.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
 set ip access-group 151 in
 set ip access-group 161 out
 match address 101
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
```

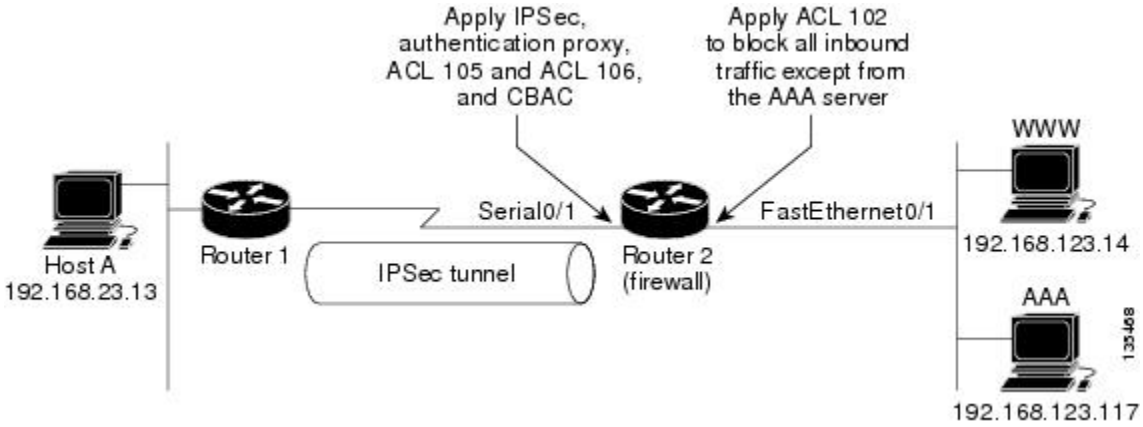
```
 ip access-group 160 out
 crypto map vpnmap
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
access-list 151 permit tcp 10.1.2.0 0.0.0.255 eq telnet 10.1.1.0 0.0.0.255
access-list 151 permit tcp 10.1.2.0 0.0.0.255 10.1.1.0 0.0.0.255 eq telnet
access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit esp host 192.168.1.1 host 192.168.2.1
access-list 161 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255 eq telnet
access-list 161 permit ip 10.1.1.0 0.0.0.255 eq telnet 10.1.2.0 0.0.0.255
```

Authentication Proxy IPSec and CBAC Configuration Example

The following example shows a router configuration using the authentication proxy, IPSec, and CBAC features. The figure below illustrates the configuration.

Note This configuration is effective for Cisco IOS Release 12.3(8)T software and later.

Figure 7: Router Configuration Using Authentication Proxy, IPSec, and CBAC Features



In this example, Host A initiates a HTTP connection with the web server (WWW). The HTTP traffic between Router 1 and Router 2 is encrypted using IPSec. The authentication proxy, IPSec, and CBAC are configured at interface Serial0/1 on Router 2, which is acting as the firewall. ACL 105 allows only IPSec traffic at interface Serial0/1. ACL 106 is crypto access check, which blocks all traffic. ACL 102 is applied at interface FastEthernet0/1 on Router 2 to block all traffic on that interface except traffic from the AAA server.

When Host A initiates a HTTP connection with the web server, the authentication proxy prompts the user at Host A for a username and password. These credentials are verified with the AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness:

- Authentication Proxy IPSec and CBAC Configuration Example, on page 75
- Authentication Proxy IPSec and CBAC Configuration Example, on page 75

Router 1 Configuration Example

```
version 12.3
service timestamps debug uptime
service timestamps log uptime
!
hostname Router1
!
boot-start-marker
boot-end-marker
!
ip subnet-zero
ip cef
!
!
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
!
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
crypto isakmp key cisco1234 address 10.0.0.2
!
!
crypto ipsec transform-set rule_1 esp-gcm
!
crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.2
 set transform-set rule_1
 match address 155
!
!
interface FastEthernet0/0
 ip address 192.168.23.2 255.255.255.0
 speed auto
!
interface Serial1/1
 ip address 10.0.0.1 255.0.0.0
 encapsulation ppp
 clockrate 2000000
 crypto map testtag
!
ip classless
ip route 192.168.123.0 255.255.255.0 10.0.0.2
!
no ip http server
no ip http secure-server
!
access-list 155 permit ip 192.168.23.0 0.0.0.255 192.168.123.0 0.0.0.255
!
control-plane
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
end
```

Router 2 Configuration Example

```
version 12.3
service timestamps debug uptime
```

```
service timestamps log uptime
!
hostname Router2
!
boot-start-marker
boot-end-marker
!
!
resource policy
!
aaa new-model
!
!
aaa authentication login default group tacacs+
aaa authentication login console none
aaa authorization auth-proxy default group tacacs+
!
aaa session-id common
clock timezone MST -8
clock summer-time MDT recurring
no network-clock-participate slot 1
no network-clock-participate wic 0
ip subnet-zero
!
!
no ip dhcp use vrf connected
!
!
ip cef
ip inspect name rule22 tcp
ip inspect name rule22 ftp
ip inspect name rule22 smtp
ip auth-proxy name pxy http inactivity-time 60
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
!
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
crypto isakmp key cisco1234 address 10.0.0.1
!
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-gcm
!
crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.1
 ! Define crypto access check to filter traffic after IPSec decryption
 ! Authentication-proxy downloaded ACEs will be added to this ACL,
 ! not interface ACL.
 set ip access-group 106 in
  set transform-set rule_1
  match address 155
!
!
interface FastEthernet0/1
 ip address 192.168.123.2 255.255.255.0
 ip access-group 102 in
 duplex auto
 speed auto
!
interface Serial0/1
 ip address 10.0.0.2 255.0.0.0
 ip access-group 105 in
 ip inspect rule22 in
 ip auth-proxy pxy
 encapsulation ppp
 crypto map testtag
!
no ip classless
ip route 192.168.23.0 255.255.255.0 10.0.0.1
```

```
!
!
ip http server
ip http access-class 15
ip http authentication aaa
no ip http secure-server
!
access-list 15 deny any
access-list 102 permit tcp host 192.168.123.20 117 eq tacacs host 192.168.123.2
! ACL 155 is interface ACL which allows only IPSec traffic
access-list 105 permit ahp any any
access-list 105 permit esp any any
access-list 105 permit udp any any eq isakmp
! ACL 106 is crypto access check ACL
access-list 106 deny ip any any
access-list 155 permit ip 192.168.123.0 0.0.0.255 192.168.23.0 0.0.0.255
!
!
tacacs-server host 192.168.123.117
tacacs-server directed-request
tacacs-server key cisco
!
control-plane
!
!
line con 0
  exec-timeout 0 0
  login authentication console
line aux 0
 transport input all
 speed 38400
 flowcontrol hardware
line vty 0 4
login authentication console
!
End
```

TACAC+ User Profile Example

```
user = http_1 {
     default service = permit
     login = cleartext mypassword
     service = auth-proxy
          {
              priv-lvl=15
              proxyacl#1="permit tcp any any eq 23"
              proxyacl#2="permit tcp any any eq 21"
              proxyacl#3="permit tcp any any eq 25"
              proxyacl#4="permit tcp any any eq 80"
              proxyacl#5="permit udp any any eq 53"
```

ACL 106, Before Auth-Proxy Authentication

```
Router2# show access-list 106
Extended IP access list 106
    10 deny ip any any (4 matches)
```

ACL 106, After Auth-Proxy Authentication

```
Router2#

show access-list 106
Extended IP access list 106
     permit tcp host 192.168.23.116 any eq telnet
     permit tcp host 192.168.23.116 any eq ftp
     permit tcp host 192.168.23.116 any eq smtp
     permit tcp host 192.168.23.116 any eq www (6 matches)
```

```
 permit udp host 192.168.23.116 any eq domain
10 deny ip any any (4 matches)
```

Additional References

Related Documents

Related Topic	Document Title
Configuring IPSec	“Configuring Internet Key Exchange for IPsec VPNs ” section of the *Cisco IOS Security Configuration Guide: Secure Connectivity*
Configuring ACLs	“ Creating an IP Access List and Applying It to an Interface ” *section of the Cisco IOS Security Configuration Guide: Securing the Data Plane Configuration Guide*
IPSec Commands	Cisco IOS Security Command Reference
Recommended cryptographic algorithms	Next Generation Encryption

MIBs

MIBs	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

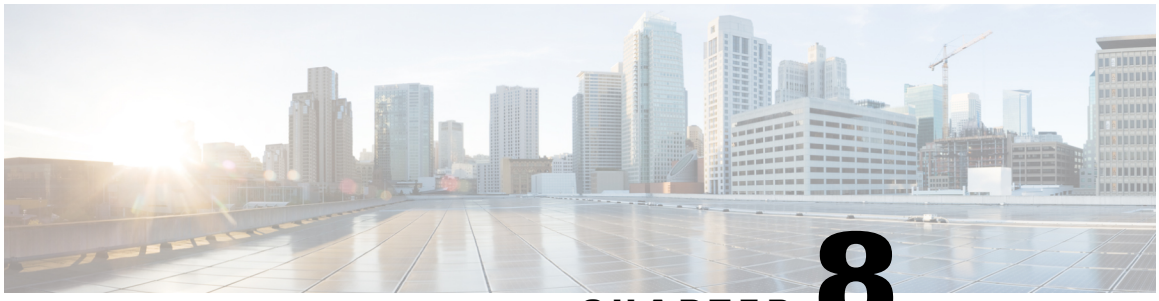
Feature Information for Crypto Access Check on Clear-Text Packets

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 Crypto Access Check on Clear-Text Packets

Feature Name	Releases	Feature Information
Crypto Access Check on Clear-Text Packets	12.3(8)T	The Crypto Access Check on Clear-Text Packets feature removes the checking of clear-text packets that go through the IP Security (IPSec) tunnel just prior to encryption or just after decryption. The clear-text packets were checked against the outside physical interface access control lists (ACLs). This checking was often referred to as a double ACL check. This feature enables easier configuration of ACLs and eliminates the security risks that are associated with a double check when using dynamic crypto maps. This feature was introduced in Cisco IOS Release 12.3(8)T. The following commands were introduced or modified: **set ip access-group , show crypto map (IPsec) .**



CHAPTER 8

IPsec Security Association Idle Timers

When a router running the Cisco IOS software creates an IPsec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers.

With the introduction of the IPsec Security Association Idle Timers feature, there is now an idle timer that can be configured to monitor SAs for activity, allowing SAs for idle peers to be deleted and new SAs to be created as required to increase the availability of resources. This feature also improves the scalability of Cisco IOS IPsec deployments.

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 at the end of this module.

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 IPsec Security Association Idle Timers

You must configure Internet Key Exchange (IKE) as described in Internet Key Exchange for IPsec VPNs

Information About IPsec Security Association Idle Timers

Lifetimes for IPsec Security Associations

The Cisco IOS software currently allows the configuration of lifetimes for IPsec SAs. Lifetimes can be configured globally or per crypto map. There are two lifetimes: a “timed” lifetime and a “traffic-volume” lifetime. A security association expires after the first of these lifetimes is reached.

IPsec Security Association Idle Timers

The IPsec SA idle timers are different from the global lifetimes for IPsec SAs. The expiration of the global lifetime is independent of peer activity. The IPsec SA idle timer allows SAs associated with inactive peers to be deleted before the global lifetime has expired.

If the IPsec SA idle timers are not configured, only the global lifetimes for IPsec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

Note If the last IPsec SA to a given peer is deleted due to idle timer expiration, the Internet Key Exchange (IKE) SA to that peer will also be deleted.

How to Configure IPsec Security Association Idle Timers

Configuring the IPsec SA Idle Timer Globally

This task configures the IPsec SA idle timer globally. The idle timer configuration will be applied to all SAs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association idle-time** *seconds* **Example:** `Router(config)# crypto ipsec security-association idle-time 600`	Configures the IPsec SA idle timer. • The *seconds* argument specifies the time, in seconds, that the idle timer will allow an inactive peer to maintain an SA. Valid values for the *seconds* argument range from 60 to 86400.

Configuring the IPsec SA Idle Timer per Crypto Map

This task configures the IPsec SA idle timer for a specified crypto map. The idle timer configuration will be applied to all SAs under the specified crypto map.

Note This configuration task was available effective with Cisco IOS Release 12.3(14)T.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-number* **ipsec-isakmp**
4. **set security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto map** *map-name seq-number* **ipsec-isakmp** **Example:** `Router(config)# crypto map test 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**set security-association idle-time** *seconds* **Example:** `Router(config-crypto-map)# set security-association idle-time 600`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used. • The *seconds* argument is the number of seconds for which the current peer can be idle before the default peer is used. Valid values are 60 to 86400.

Configuration Examples for IPsec Security Association Idle Timers

Configuring the IPsec SA Idle Timer Globally Example

The following example globally configures the IPsec SA idle timer to drop SAs for inactive peers after 600 seconds:

```
crypto ipsec security-association idle-time 600
```

Configuring the IPsec SA Idle Timer per Crypto Map Example

The following example configures the IPsec SA idle timer for the crypto map named test to drop SAs for inactive peers after 600 seconds:

```
crypto map test 1 ipsec-isakmp
 set security-association idle-time 600
```

Note The above configuration was not available until Cisco IOS Release 12.3(14)T.

Additional References

Related Documents

Related Topic	Document Title
Additional information about configuring IKE	Internet Key Exchange for IPsec VPNs
Additional information about configuring global lifetimes for IPsec SAs	• Configuring Security for VPNs with IPsec • IPsec Preferred Peer
Additional Security commands	Cisco IOS Security Command Reference

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

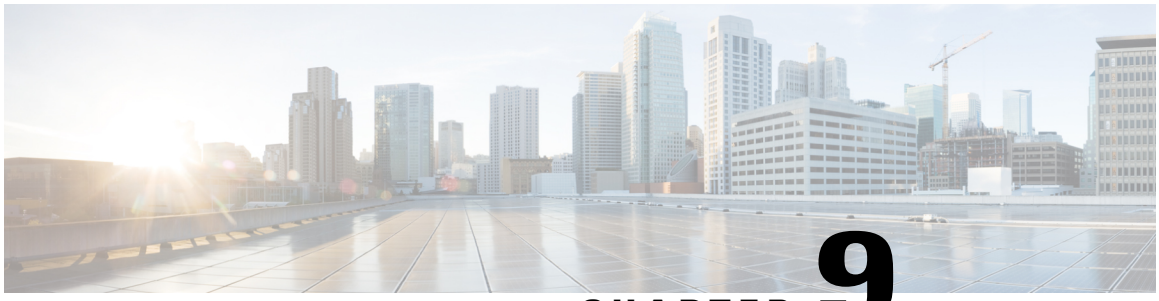
Feature Information for IPsec Security Association Idle Timers

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 9: Feature Information for IPsec Security Association Idle Timers

Feature Name	Releases	Feature Information
IPsec Security Association Idle Timers	12.2(15)T 12.3(14)T	With the introduction of the IPsec Security Association Idle Timers feature, there is now an idle timer that can be configured to monitor SAs for activity, allowing SAs for idle peers to be deleted and new SAs to be created as required to increase the availability of resources. This feature also improves the scalability of Cisco IOS IPsec deployments. This feature was introduced in Cisco IOS Release 12.2(15)T. In Cisco IOS Release 12.3(14)T, the **set security-association idle-time** command was added, allowing for the configuration of an IPsec idle timer for a specified crypto map. The following commands were introduced or modified: **crypto ipsec security-association idle-time, set security-association idle-time .**



CHAPTER 9

Low Latency Queueing for IPsec Encryption Engines

The Low Latency Queueing (LLQ) for IPsec Encryption Engines feature helps reduce overall network latency and congestion by queueing priority designated traffic before it is processed by the crypto processing engine. This queueing guarantees a certain level of crypto engine processing time.

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 at the end of this module.

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 LLQ for IPsec Encryption Engines

To use this feature, you should be familiar with the following:

- Access control lists
- Bandwidth management
- CBWFQ

Restrictions for LLQ for IPsec Encryption Engines

- No per-tunnel QoS policy. An interface QoS policy represents all tunnels.
- Assume the same IP precedence/DSCP marking for inbound and outbound voice packets.
- Assume the IP precedence/DSCP marking for voice packets are done at the source.
- Limited match criteria for voice traffic in the interface QoS policy.
- Assume call admission control is enforced within the enterprise.
- No strict error checking when aggregate policy's bandwidth exceeds crypto engine bandwidth. Only a warning is displayed but configuration is allowed.
- Assume voice packets are either all encrypted or unencrypted.

Information About LLQ for IPsec Encryption Engines

LLQ for IPsec Encryption Engines

Administrators can now use the Low Latency Queueing (LLQ) for IPsec Encryption Engines feature to prioritize voice and data traffic, which was previously only given equal status.

- Voice packets arriving on a router interface can be identified as priority and be directed into a priority packet inbound queue for crypto engine processing. This queue is called the priority queue. The crypto engine undertakes packet processing in a favorable ratio for voice packets. Voice packets are guaranteed a minimum processing bandwidth on the crypto engine. This feature impacts the end user experience by assuring voice quality if voice traffic is directed onto a congested network.
- Data packets arriving at a router interface are directed into a data packet inbound queue for crypto engine processing. This queue is called the best effort queue.

How to Configure LLQ for IPsec Encryption Engines

Perform the tasks described in this section to configure LLQ for IPsec Encryption Engines.

Note See the Quality of Service Solutions Command Reference to learn more about configuring server policies on interfaces.

- Defining Class Maps, on page 89 (required)

- Configuring Class Policy in the Policy Map, on page 90 (required)
- Attaching the Service Policy, on page 94 (required)
- Viewing the LLQ for IPsec Encryption Engines Configuration, on page 95 (optional)

Defining Class Maps

The following steps are used to create a class map containing match criteria against which a packet is checked to determine if it belongs to a class:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** class-map-name
4. **match access-group** {access-group | name access-group-name}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** class-map-name **Example:** `Router(config)# class-map voice`	Specifies the name of the class map to be created.
Step 4	**match access-group** {access-group | name access-group-name} **Example:** `-or-` **Example:**	• The **match access-group** command specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. • The **match input-interface** command specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class. • The **match protocol** command specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.

	Command or Action	Purpose
	match input-interface `interface-name` **Example:** `-or-` **Example:** **match protocol** `protocol` **Example:** `Router(config-cmap)#` **match access-group 102**	

Configuring Class Policy in the Policy Map

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of the other classes defined in the policy map.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes in a policy map must not exceed the minimum committed information rate (CIR) configured for the virtual circuit (VC) minus any bandwidth reserved by the **frame-relay voice bandwidth** and **frame-relay ip rtp priority** commands. If the minimum CIR is not configured, the bandwidth defaults to one half of the CIR. If all of the bandwidth is not allocated, the remaining bandwidth is allocated proportionally among the classes on the basis of their configured bandwidth.

To configure class policies in a policy map, perform the tasks described in the following sections.

- Configuring Class Policy for a Priority Queue, on page 90 (required)
- Configuring Class Policy Using a Specified Bandwidth, on page 91 (optional)
- Configuring the Class-Default Class Policy, on page 92 (optional)

Configuring Class Policy for a Priority Queue

The following steps are used to configure a policy map and give priority to a class within the policy map:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class** *class-name*
5. **priority** *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map to be created or modified.
Step 4	**class** *class-name* **Example:** `Router(config-pmap)#class voice`	Specifies the name of a class to be created and included in the service policy.
Step 5	**priority** *bandwidth-kbps* **Example:** `Router(config-pmap-c)# priority 50`	Creates a strict priority class and specifies the amount of bandwidth, in kbps, to be assigned to the class.

Configuring Class Policy Using a Specified Bandwidth

The following steps are used to configure a policy map and create class policies that make up the service policy. To configure more than one class in the same policy map, repeat Configuring Class Policy Using a Specified Bandwidth and Configuring Class Policy Using a Specified Bandwidth.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class** *class-name*
5. **bandwidth** *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map to be created or modified.
Step 4	**class** *class-name* **Example:** `Router(config-pmap)# class voice`	Specifies the name of a class to be created and included in the service policy.
Step 5	**bandwidth** *bandwidth-kbps* **Example:** `Router(config-pmap-c)# bandwidth 20`	Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.)

Configuring the Class-Default Class Policy

The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given best-effort

treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

The following steps are used to configure a policy map and the class-default class:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** policy-map
4. **class class-default** default-class-name
5. **bandwidth** bandwidth-kbps

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** policy-map **Example:** `Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 4	**class class-default** default-class-name **Example:** `Router(config-pmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.
Step 5	**bandwidth** bandwidth-kbps **Example:** `-or-` **Example:**	Either the **bandwidth** or **fair-queue** command can be used for this step. • The **bandwidth** command specifies the amount of bandwidth, in kbps, to be assigned to the class. • The **fair-queue** command specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.

	Command or Action	Purpose
	fair-queue `[number-of-dynamic-queues]` **Example:** `Router(config-pmap-c)# fair-queue`	

Attaching the Service Policy

The following steps are used to attach a service policy to the output interface and enable LLQ for IPsec encryption engines.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy output** *policy-map*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet0/0`	Specifies the interface using the LLQ for IPsec encryption engines.

	Command or Action	Purpose
Step 4	**service-policy output** *policy-map* **Example:** `Router(config-if)# service-policy output policy1`	Attaches the specified service policy map to the output interface and enables LLQ for IPsec encryption engines.

Viewing the LLQ for IPsec Encryption Engines Configuration

Viewing the LLQ for IPsec Encryption Engines Configuration

The following steps are used to view the contents of a specific policy map or all policy maps configured on an interface, and the LLQ for IPsec encryption engines:

SUMMARY STEPS

1. **enable**
2. **show frame-relay pvc dlci**
3. **show policy-map interface** interface-name
4. **show policy-map interface** interface-name **dlci** *dlci-number*
5. **show crypto eng qos**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show frame-relay pvc dlci** **Example:** `Router# show frame-relay pvc dlci`	Displays statistics about the PVC and the configuration of classes for the policy map on the specified data-link connection identifier (DLCI).

	Command or Action	Purpose
Step 3	**show policy-map interface** interface-name **Example:** `Router# show policy-map interface fastethernet0/0`	When LLQ is configured, displays the configuration of classes for all policy maps.
Step 4	**show policy-map interface** interface-name **dlci** *dlci-number* **Example:** `Router# show policy-map interface fastethernet0/0 dlci 100`	When LLQ is configured, displays the configuration of classes for the policy map on the specified DLCI.
Step 5	**show crypto eng qos** **Example:** `Router#` **show crypto eng qos**	Displays quality of service queueing statistics for LLQ for IPsec encryption engines.

Configuration Examples for LLQ for IPsec Encryption Engines

LLQ for IPsec Encryption Engines Example

In the following example, a strict priority queue with a guaranteed allowed bandwidth of 50 kbps is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

```
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000
```

Next, the class map voice is defined, and the policy map called policy1 is created; a strict priority queue for the class voice is reserved, a bandwidth of 20 kbps is configured for the class bar, and the default class is configured for WFQ. The service-policy command then attaches the policy map to the fas0/0.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
```

```
Router(config)# interface fas0/0
Router(config-if)# service-policy output policy1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	Cisco IOS Security Command Reference
QoS Commands	Cisco IOS Quality of Service Solutions Command Reference
Weighted Fair Queueing	Configuring Weighted Fair Queueing feature module.

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LLQ for IPsec Encryption Engines

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.

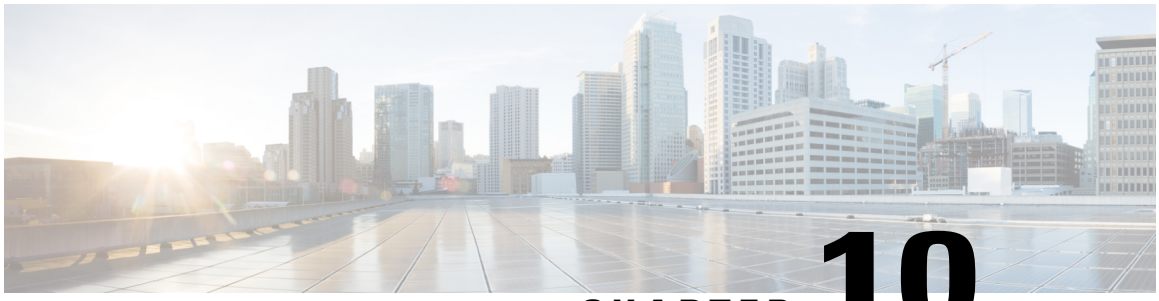
Table 10: Feature Information for Low Latency Queueing (LLQ) for IPsec Encryption Engines

Feature Name	Releases	Feature Information
Feature Information for Low Latency Queueing (LLQ) for IPsec Encryption Engines	12.2(13)T 12.2(14)S	The Low Latency Queueing (LLQ) for IPsec Encryption Engines feature helps reduce overall network latency and congestion by queueing priority designated traffic before it is processed by the crypto processing engine. This queueing guarantees a certain level of crypto engine processing time. This feature was introduced in Cisco IOS Release 12.2(13)T. This feature was integrated into Cisco IOS Release 12.2(14)S. The following commands were introduced or modified: **show crypto eng qos .**

Glossary

IKE --Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPsec). 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.



CHAPTER 10

IPsec IPv6 Phase 2 Support

Cisco IOS IPv6 security features for your Cisco networking devices can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

Cisco IOS IPsec functionality provides network data encryption at the IP packet level, offering a robust, standards-based security solution. IPsec provides data authentication and anti-replay services in addition to data confidentiality services.

IPsec is a mandatory component of IPv6 specification. OSPF for IPv6 provides IPsec authentication support and protection, and IPv6 IPsec tunnel mode and encapsulation is used to protect IPv6 unicast and multicast traffic. This document provides information about implementing IPsec in IPv6 security.

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.

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 at the end of this module.

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 IPsec IPv6 Phase 2 Support

IPsec for IPv6

IP Security, or IPsec, is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provide security for 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. IPsec provides the following optional network security services. In general, local security policy will dictate the use of one or more of these services:

- Data confidentiality--The IPsec sender can encrypt packets before sending 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 IPsec packets sent. This service depends upon the data integrity service.
- Antireplay--The IPsec receiver can detect and reject replayed packets.

With IPsec, data can be sent across a public network without observation, modification, or spoofing. IPsec functionality is similar in both IPv6 and IPv4; however, site-to-site tunnel mode only is supported in IPv6.

In IPv6, IPsec is implemented using the AH authentication header and the ESP extension header. The authentication header provides integrity and authentication of the source. It also provides optional protection against replayed packets. The authentication header protects the integrity of most of the IP header fields and authenticates the source through a signature-based algorithm. The ESP header provides confidentiality, authentication of the source, connectionless integrity of the inner packet, antireplay, and limited traffic flow confidentiality.

The Internet Key Exchange (IKE) protocol is a key management protocol standard that is used in conjunction with IPsec. IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

IKE is a hybrid protocol that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE) (see the figure below). This functionality is similar to the security gateway model using IPv4 IPsec protection.

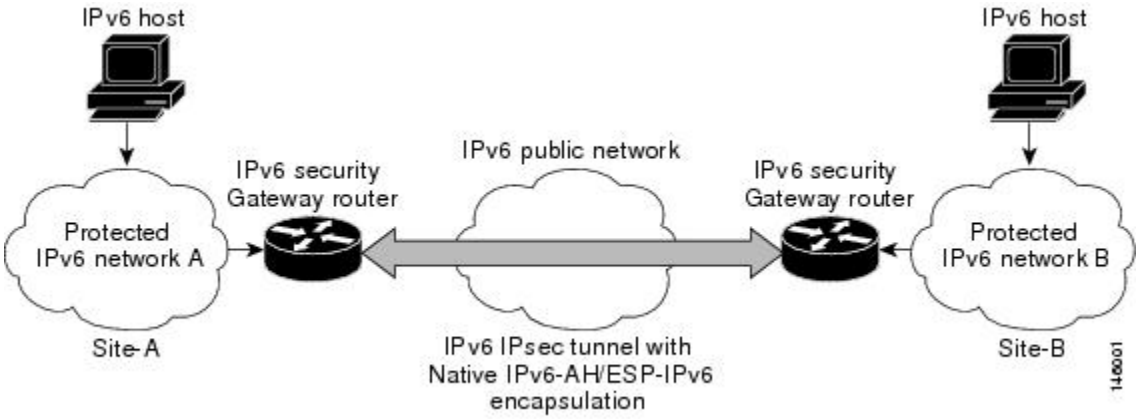
IPv6 IPsec Site-to-Site Protection Using Virtual Tunnel Interface

The IPsec virtual tunnel interface (VTI) provides site-to-site IPv6 crypto protection of IPv6 traffic. Native IPv6 IPsec encapsulation is used to protect all types of IPv6 unicast and multicast traffic.

The IPsec VTI allows IPv6 routers to work as security gateways, establish IPsec tunnels between other security gateway routers, and provide crypto IPsec protection for traffic from internal networks when it is sent across

the public IPv6 Internet (see the figure below). This functionality is similar to the security gateway model using IPv4 IPsec protection.

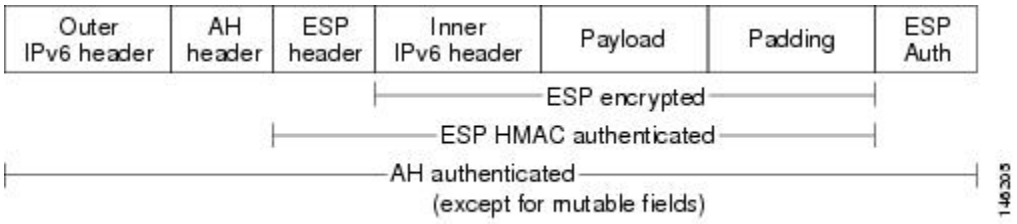
Figure 8: IPsec Tunnel Interface for IPv6



When the IPsec tunnel is configured, IKE and IPsec security associations (SAs) are negotiated and set up before the line protocol for the tunnel interface is changed to the UP state. The remote IKE peer is the same as the tunnel destination address; the local IKE peer will be the address picked from tunnel source interface which has the same IPv6 address scope as tunnel destination address.

The following figures shows the IPsec packet format.

Figure 9: IPv6 IPsec Packet Format



How to Configure IPsec IPv6 Phase 2 Support

Configuring a VTI for Site-to-Site IPv6 IPsec Protection

Creating an IKE Policy and a Preshared Key in IPv6

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

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

You can configure multiple, prioritized policies on each peer--each with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

Note If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the value supported by the other device. Aside from this limitation, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of security risks for your network and your tolerance for these risks.

When the IKE negotiation begins, IKE searches for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the policies received from the other peer. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

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

If a match is found, IKE will complete negotiation, and IPsec security associations will be created. If no acceptable match is found, IKE refuses negotiation and IPsec will not be established.

Note Depending on which authentication method is specified in a policy, additional configuration might be required. If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

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

When two peers use IKE to establish IPsec SAs, each peer sends its identity to the remote peer. Each peer sends either its hostname or its IPv6 address, depending on how you have set the ISAKMP identity of the router.

By default, a peer's ISAKMP identity is the IPv6 address of the peer. If appropriate, you could change the identity to be the peer's hostname instead. As a general rule, set the identities of all peers the same way--either all peers should use their IPv6 addresses or all peers should use their hostnames. If some peers use their hostnames and some peers use their IPv6 addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a DNS lookup is unable to resolve the identity.

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 isakmp policy** *priority*
4. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
5. **hash** {**md5** | **sha** | **sha256** | **sha384** | **sha512**}
6. **group** {**1** | **14** | **15** | **16** | **19** | **2** | **20** | **24** | **5**}
7. **encryption** {**3des** | **aes** | **aes 192** | **aes 256** | **des**}
8. **lifetime** *seconds*
9. **exit**
10. **crypto isakmp key** *enc-type-digit keystring* { **address** *peer-address [mask]* | **ipv6** {*ipv6-address*/*ipv6-prefix*} | **hostname** *hostname} [* **no-xauth**]
11. **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]
12. **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname* | **ipv6** {*ipv6-address* | *ipv6-prefix*}} **key** *key*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 15`	Defines an IKE policy, and enters ISAKMP policy configuration mode. Policy number 1 indicates the policy with the highest priority. The smaller the *priority* argument value, the higher the priority.
Step 4	**authentication** {**rsa-sig**	**rsa-encr**	**pre-share**} **Example:** `Router(config-isakmp-policy)# authentication pre-share`	Specifies the authentication method within an IKE policy. The **rsa-sig** and **rsa-encr** keywords are not supported in IPv6.

	Command or Action	Purpose
Step 5	**hash** {**md5** | **sha** | **sha256** | **sha384** | **sha512**} **Example:** `Router(config-isakmp-policy)# hash sha`	Specifies the hash algorithm within an IKE policy. The algorithm **md5** is no longer recommended. SHA-1, SHA-256, SHA-384 and SHA-512 are the recommended hash algorithms.
Step 6	**group** {**1** | **14** | **15** | **16** | **19** | **2** | **20** | **24** | **5**} **Example:** `Router(config-isakmp-policy)# group 14`	Specifies the Diffie-Hellman group identifier within an IKE policy. • **1—**768-bit DH (No longer recommended.) • **14—**Specifies the 2048-bit DH group. • **15—**Specifies the 3072-bit DH group. • **16—**Specifies the 4096-bit DH group. • **19—**Specifies the 256-bit elliptic curve DH (ECDH) group. • **2—**1024-bit DH (No longer recommended.) • **20—**Specifies the 384-bit ECDH group. • **24—**Specifies the 2048-bit DH/DSA group. • **5—**1536-bit DH (No longer recommended.)
Step 7	**encryption** {**3des** | **aes** | **aes 192** | **aes 256** | **des**} **Example:** `Router(config-isakmp-policy)# encryption aes`	Specifies the encryption algorithm within an IKE policy. • **3des—**168-bit DES (No longer recommended. AES is the recommended encryption algorithm.) • **aes—**128-bit AES • **aes 192—**192-bit AES • **aes 256—**256-bit AES • **des—**56-bit DES-CBC (No longer recommended. AES is the recommended encryption algorithm.)
Step 8	**lifetime** *seconds* **Example:** `Router(config-isakmp-policy)# lifetime 43200`	Specifies the lifetime of an IKE SA. Setting the IKE lifetime value is optional.
Step 9	**exit** **Example:** `Router(config-isakmp-policy)# exit`	Enter this command to exit ISAKMP policy configuration mode and enter global configuration mode.
Step 10	**crypto isakmp key** *enc-type-digit keystring* { **address** *peer-address [mask]* | **ipv6** {*ipv6-address/ipv6-prefix*} | **hostname** *hostname}* [**no-xauth**]	Configures a preshared authentication key.

	Command or Action	Purpose
	Example: `Router(config)# crypto isakmp key 0 my-preshare-key-0 address ipv6 3ffe:1001::2/128`	
Step 11	**crypto keyring** *keyring-name* [**vrf** *fvrf-name*] **Example:** `Router(config)# crypto keyring keyring1`	Defines a crypto keyring to be used during IKE authentication.
Step 12	**pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname* | **ipv6** {*ipv6-address* | *ipv6-prefix*}} **key** *key* **Example:** `Router (config-keyring)# pre-shared-key ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	Defines a preshared key to be used for IKE authentication.
Step 13	**end** **Example:** `Router (config-keyring)# end`	Exits crypto keyring configuration mode and returns to privileged EXEC mode.

Configuring ISAKMP Aggressive Mode

You likely do not need to configure aggressive mode in a site-to-site scenario. The default mode is typically used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer** {**address** {*ipv4-address* | **ipv6** *ipv6-address ipv6-prefix-length*} | **hostname** *fqdn-hostname*}
4. **set aggressive-mode client-endpoint** {*client-endpoint* | **ipv6** *ipv6-address*}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp peer** {**address** {*ipv4-address* | **ipv6** *ipv6-address ipv6-prefix-length*} | **hostname** *fqdn-hostname*} **Example:** `Router(config)# crypto isakmp peer address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	Enables an IPsec peer for IKE querying for tunnel attributes.
Step 4	**set aggressive-mode client-endpoint** {*client-endpoint* | **ipv6** *ipv6-address*} **Example:** `Router(config-isakmp-peer)# set aggressive mode client-endpoint ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	Defines the remote peer's IPv6 address, which will be used by aggressive mode negotiation. The remote peer's address is usually the client side's end-point address.
Step 5	**end** **Example:** `Router(config-isakmp-peer)# end`	Exits crypto ISAKMP peer configuration mode and returns to privileged EXEC mode.

Configuring an IPsec Transform Set and IPsec Profile

A transform set is a combination of security protocols and algorithms that is acceptable to the IPsec routers.

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 ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*]
4. **crypto ipsec profile** *name*
5. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*] **Example:** `Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-aes`	Defines a transform set, and places the router in crypto transform configuration mode.
Step 4	**crypto ipsec profile** *name* **Example:** `Router(config)# crypto ipsec profile profile0`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers.
Step 5	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router (config-crypto-transform)# set-transform-set myset0`	Specifies which transform sets can be used with the crypto map entry.
Step 6	**end** **Example:** `Router (config-crypto-transform)# end`	Exits crypto transform configuration mode and returns to privileged EXEC mode.

Defining an ISAKMP Profile in IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name* [**accounting** *aaalist*
4. **self-identity** {**address** | **address ipv6**] | **fqdn** | **user-fqdn** *user-fqdn*}
5. **match identity** {**group** *group-name* | **address** {*address* [*mask*] [*fvrf*] | **ipv6** *ipv6-address*} | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* [**accounting** *aaalist* **Example:** `Router(config)# crypto isakmp profile profile1`	Defines an ISAKMP profile and audits IPsec user sessions.
Step 4	**self-identity** {**address**	**address ipv6**]	**fqdn**	**user-fqdn** *user-fqdn*} **Example:** `Router(config-isakmp-profile)# self-identity address ipv6`	Defines the identity that the local IKE uses to identify itself to the remote peer.
Step 5	**match identity** {**group** *group-name*	**address** {*address* [*mask*] [*fvrf*]	**ipv6** *ipv6-address*}	**host** *host-name*	**host domain** *domain-name*	**user** *user-fqdn*	**user domain** *domain-name*}	Matches an identity from a remote peer in an ISAKMP profile.

	Command or Action	Purpose
	Example: `Router(config-isakmp-profile)# match identity address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	
Step 6	**end** **Example:** `Router(config-isakmp-profile)# end`	Exits ISAKMP profile configuration mode and returns to privileged EXEC mode.

Configuring IPv6 IPsec VTI

Before You Begin

Use the **ipv6 unicast-routing** command to enable IPv6 unicast routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface tunnel** *tunnel-number*
5. **ipv6 address** *ipv6-address/prefix*
6. **ipv6 enable**
7. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
8. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
9. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**}
10. **tunnel protection ipsec profile** *name* [**shared**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables IPv6 unicast routing. You only need to enable IPv6 unicast routing once, not matter how many interface tunnels you want to configure.
Step 4	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 5	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 3FFE:C000:0:7::/64 eui-64`	Provides an IPv6 address to this tunnel interface, so that IPv6 traffic can be routed to this tunnel.
Step 6	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Enables IPv6 on this tunnel interface.
Step 7	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source ethernet0`	Sets the source address for a tunnel interface.
Step 8	**tunnel destination** {*host-name* | *ip-address* | *ipv6-address*} **Example:** `Router(config-if)# tunnel destination 2001:DB8:1111:2222::1`	Specifies the destination for a tunnel interface.
Step 9	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**} **Example:** `Router(config-if)# tunnel mode ipsec ipv6`	Sets the encapsulation mode for the tunnel interface. For IPsec, only the **ipsec ipv6** keywords are supported.

	Command or Action	Purpose
Step 10	**tunnel protection ipsec profile** *name* [**shared**] **Example:** `Router(config-if)# tunnel protection ipsec profile profile1`	Associates a tunnel interface with an IPsec profile. IPv6 does not support the **shared** keyword.
Step 11	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying IPsec Tunnel Mode Configuration

SUMMARY STEPS

1. **show adjacency** [**summary** [*interface-type interface-number*]] | [**prefix**] [**interface** *interface-number*] [**connectionid** *id*] [**link** {**ipv4**| **ipv6** | **mpls**}] [**detail**]
2. **show crypto engine** {**accelerator** | **brief** | **configuration** | **connections** [**active** | **dh** | **dropped-packet** | **show**] | **qos**}
3. **show crypto ipsec sa** [**ipv6**] [*interface-type interface-number*] [**detailed**]
4. **show crypto isakmp peer** [**config** | **detail**]
5. **show crypto isakmp policy**
6. **show crypto isakmp profile** [**tag** *profilename* | **vrf** *vrfname*]
7. **show crypto map** [**interface** *interface* | **tag** *map-name*]
8. **show crypto session** [**detail**] | [**local** *ip-address* [**port** *local-port*] | [**remote** *ip-address* [**port** *remote-port*]] | **detail**] | **fvfr** *vrf-name* | **ivrf** *vrf-name*]
9. **show crypto socket**
10. **show ipv6 access-list** [*access-list-name*]
11. **show ipv6 cef** [*ipv6-prefix* / *prefix-length*] | [*interface-type interface-number*] [**longer-prefixes** | **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]]
12. **show interface** *type number* **stats**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show adjacency** [**summary** [*interface-type interface-number*]] | [**prefix**] [**interface** *interface-number*] [**connectionid** *id*] [**link** {**ipv4**| **ipv6** | **mpls**}] [**detail**] **Example:** `Router# show adjacency detail`	Displays information about the Cisco Express Forwarding adjacency table or the hardware Layer 3-switching adjacency table.
Step 2	**show crypto engine** {**accelerator** | **brief** | **configuration** | **connections** [**active** | **dh** | **dropped-packet** | **show**] | **qos**} **Example:** `Router# show crypto engine connection active`	Displays a summary of the configuration information for the crypto engines.
Step 3	**show crypto ipsec sa** [**ipv6**] [*interface-type interface-number*] [**detailed**] **Example:** `Router# show crypto ipsec sa ipv6`	Displays the settings used by current SAs in IPv6.
Step 4	**show crypto isakmp peer** [**config** | **detail**] **Example:** `Router# show crypto isakmp peer detail`	Displays peer descriptions.
Step 5	**show crypto isakmp policy** **Example:** `Router# show crypto isakmp policy`	Displays the parameters for each IKE policy.
Step 6	**show crypto isakmp profile** [**tag** *profilename* | **vrf** *vrfname*] **Example:** `Router# show crypto isakmp profile`	Lists all the ISAKMP profiles that are defined on a router.
Step 7	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays the crypto map configuration. The crypto maps shown in this command output are dynamically generated. The user does not have to configure crypto maps.
Step 8	**show crypto session** [**detail**] | [**local** *ip-address* [**port** *local-port*] | [**remote** *ip-address* [**port** *remote-port*]] | **detail**] | **fvfr** *vrf-name* | **ivrf** *vrf-name*]	Displays status information for active crypto sessions.

	Command or Action	Purpose
	Example: `Router# show crypto session`	IPv6 does not support the **fvfr** or **ivrf** keywords or the *vrf-name* argument.
Step 9	**show crypto socket** **Example:** `Router# show crypto socket`	Lists crypto sockets.
Step 10	**show ipv6 access-list** [*access-list-name*] **Example:** `Router# show ipv6 access-list`	Displays the contents of all current IPv6 access lists.
Step 11	**show ipv6 cef** [*ipv6-prefix* / *prefix-length*] | [*interface-type interface-number*] [**longer-prefixes** | **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]] **Example:** `Router# show ipv6 cef`	Displays entries in the IPv6 Forwarding Information Base (FIB).
Step 12	**show interface** *type number* **stats** **Example:** `Router# show interface fddi 3/0/0 stats`	Displays numbers of packets that were process switched, fast switched, and distributed switched.

Troubleshooting IPsec for IPv6 Configuration and Operation

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec**
3. **debug crypto engine packet** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router# enable`	• Enter your password if prompted.
Step 2	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IPsec network events.
Step 3	**debug crypto engine packet** [**detail**] **Example:** `Router# debug crypto engine packet`	Displays the contents of IPv6 packets. **Caution** Using this command could flood the system and increase CPU usage if several packets are being encrypted.

Configuration Examples for IPsec IPv6 Phase 2 Support

Example: Configuring ISAKMP Aggressive Mode

```
Router# show crypto isakmp peer detail

Peer: 2001:DB8:0:1::1 Port: 500 Local: 2001:DB8:0:2::1
Phase1 id: 2001:DB8:0:1::1
flags:
NAS Port: 0 (Normal)
IKE SAs: 1 IPsec SA bundles: 1
last_locker: 0x141A188, last_last_locker: 0x0
last_unlocker: 0x0, last_last_unlocker: 0x0
```

Example: Configuring an ISAKMP Profile in IPv6

The following sample output shows the ISAKMP profiles that are defined on an IPv6 router.

```
Router# show crypto isakmp profile

ISAKMP PROFILE tom
Identities matched are:
ipv6-address 2001:DB8:0:1::1/32
Certificate maps matched are:
Identity presented is: ipv6-address fqdn
keyring(s): <none>
trustpoint(s): <all>
```

Example: Configuring a VTI for Site-to-Site IPv6 IPsec Protection

```
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
!
crypto isakmp key myPreshareKey0 address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128
crypto isakmp keepalive 30 30
!
crypto ipsec transform-set Trans1 ah-sha-hmac esp-aes
!
crypto ipsec profile profile0
  set transform-set Trans1
!
ipv6 cef
!
interface Tunnel0
  ipv6 address 3FFE:1001::/64 eui-64
  ipv6 enable
  ipv6 cef
  tunnel source Ethernet2/0
  tunnel destination 3FFE:2002::A8BB:CCFF:FE01:2C02
  tunnel mode ipsec ipv6
  tunnel protection ipsec profile profile0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Security 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

Related Topic	Document Title
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

Technical Assistance

Description	Link
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.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec IPv6 Phase 2 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 11: Feature Information for IPsec IPv6 Phase 2 Support

Feature Name	Releases	Feature Information
IPsec IPv6 Phase 2 Support	12.4(4)T	Features in this phase support tunnel mode for site-to-site IPsec protection of IPv6 traffic. This feature allows the use of IPv6 IPsec encapsulation to protect IPv6 unicast and multicast traffic. The following commands were introduced or modified: **authentication (IKE policy)**, **crypto ipsec profile**, **crypto isakmp key**, **crypto isakmp peer**, **crypto isakmp policy**, **crypto isakmp profile**, **crypto keyring**, **debug crypto ipv6 ipsec**, **encryption (IKE policy)**, **group (IKE policy)**, **hash (IKE policy)**, **lifetime (IKE policy)**, **match identity**, **pre-shared-key**, **self-identity**, **set aggressive-mode client-endpoint**, **set transform-set**, **show adjacency**, **show crypto engine**, **show crypto ipsec sa**, **show crypto isakmp peers**, **show crypto isakmp policy**, **show crypto isakmp profile**, **show crypto map**, **show crypto session**, **show crypto socket**, **show ipv6 access-list**, **show ipv6 cef**, **tunnel destination**, **tunnel mode**, **tunnel source**.

INDEX

I