IPsec Data Plane Configuration Guide, Cisco IOS XE Release 2

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

IPsec Anti-Replay Window Expandingand 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 map or crypto profile.
- To configure the IPsec Anti-Replay Window: Expanding and Disabling feature, you should understand the following concept: IPsec Anti-Replay Window, page 2

Information About IPsec Anti-Replay Window Expandingand 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 even though they could be one of the last 64 packets received by 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 Expandingand 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-- except for those that are specifically overridden on a per-crypto map basis), 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, page 4

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-ReplayWindow 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 2.1
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
 authentication pre-share
crypto isakmp key cisco123 address 192.165.201.2 !
crypto ipsec security-association replay window-size 1024 !
crypto ipsec transform-set basic esp-des esp-md5-hmac !
crypto map mymap 10 ipsec-isakmp
 set peer 192.165.201.2
 set transform-set basic
 match address 101
!
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial1/0
 ip address 192.165.200.2 255.255.255.252  serial restart-delay 0  crypto map mymap !
ip classless
ip route 0.0.0.0 0.0.0.0 192.165.200.1
no ip http server
no ip http secure-server
!
!
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 that 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
authentication pre-share
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-des esp-md5-hmac crypto ipsec transform-set
180cisco esp-des esp-md5-hmac crypto ipsec transform-set 190cisco esp-des esp-md5-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 FastEtherneto0
 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
logi
end
```

Additional References

The following sections provide references related to IPsec Anti-Replay Window: Expanding and Disabling.

Related Documents

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

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

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

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	Cisco IOS XE Release 2.1	The following commands were introduced or modified: **crypto ipsec security-association replay disable, ipsec security-association replay window-size, security-association replay disable, security-association replay window-size**.

Pre-Fragmentation for IPsec VPNs

The Pre-Fragmentation for IPsec VPNs feature increases performance between Cisco IOS XE 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 feature information and caveats, see 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 document.

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 XE commands	Cisco IOS Master Commands List, All Releases
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	Cisco IOS XE 2.1	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) .**

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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 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 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 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 a Preshared 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 daigram 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
4. Check for an invalid SPI message on Router B

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-des 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: 8 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-des 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: 8 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 Fastethernet1/0
interface: FastEthernet1/0
    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-des 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: 8 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-des 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: 8 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: FastEthernet1/0
    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: FastEthernet1/0
    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-des 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: 8 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-des 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: 8 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
```

Step 4 Check for an invalid SPI message on Router B

Example:

```
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-des 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-des 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-des 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-des 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-des esp-sha-hmac , sa_conn_id= 939529224
ipseca-72a#
```

Configuration Examples for Invalid SecurityParameter 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, page 23 shows the topology used for this example.

Router A

```
Router# show running-config
Building configuration...
Current configuration : 2048 bytes
!
version 2.1
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
 authentication pre-share
 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-des 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 2.1
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
 authentication pre-share
 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-des 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 FastEthernet1/0
```

```
 ip address 10.2.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
 crypto map testtag1
!
interface FastEthernet1/1
 ip address 10.0.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
!
interface FastEthernet1/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/4
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/5
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/6
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/7
 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

The following sections provide references relate to Invalid Security Parameter Index Recovery.

Related Documents

Related Topic	Document Title
Configuring IKE	Configuring Internet Key Exchange for IPsec VPNs
Interface commands	Cisco IOS Master Command List

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

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

Feature Information for Invalid Security ParameterIndex 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 Special Parameter Index (SPI) Recovery	Cisco IOS XE Release 2.1	When an invalid SPI occurs in IPsec packet processing, the Invalid Security Parameter Index Recovery feature allows for an IKE 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. The following command was introduced or modified: **crypto isakmp invalid-spi-recovery.**

IPsec Dead Peer Detection PeriodicMessage Option

The IPsec Dead Peer Detection Periodic Message Option feature allows you to configure your 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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).
- An IKE peer that supports DPD (dead peer detection). Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS XE software in all modes of operation--site-to-site, Easy VPN remote, and Easy VPN server.

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 XE Keepalive Features Work

DPD and Cisco IOS XE keepalives function on the basis of the timer. If the timer is set for 10 seconds, the router will send 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 will not find out 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 will initiate 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 XE Keepalive Featureswith Multiple Peers in the Crypto Map

DPD and Cisco IOS XE 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 will switch 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* [*retries*] [**periodic** | **on-demand**]

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 isakmp keepalive** *seconds* [*retries*] [**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 XE 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 will cause 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 will be 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.
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.

	Command or Action	Purpose
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 will cause a router to cycle through the peer list when it detects that the first peer is dead.

Note Cisco IOS XE 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.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
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 3des
  authentication pre-share
  group 2
!
```

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 esp-3des-sha esp-3des esp-sha-hmac
crypto map test 1 ipsec-isakmp
  set peer 10.2.80.209
  set transform-set esp-3des-sha
  match address 101
!
!
interface FastEthernet0
  ip address 10.1.32.14 255.255.255.0
  speed auto
  crypto map test
!
```

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 will resend 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 FastEthernet0
  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 XE Keepalives Used in Conjunction with Multiple Peers in a Crypto Map Example

The following example shows that DPD and Cisco IOS XE keepalives are used in conjunction with multiple peers in a crypto map configuration when IKE will be 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

The following sections provide references related to IPsec Dead Peer Detection Periodic Message Option.

Related Documents

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

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS 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 website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Dead Peer DetectionPeriodic 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 Dead Peer Detection

Feature Name	Releases	Feature Information
Dead Peer Detection Periodic Message Option	Cisco IOS XE Release 2.1	This feature allows you to configure your router to query the liveliness of its 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. The following command was introduced or modified: **crypto isakmp keepalive**.

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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Benefit of IPsec NAT Transparency

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.

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

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, page 47
- Incompatibility Between Checksums and NAT Resolved, page 47
- Incompatibility Between Fixed IKE Destination Ports and PAT Resolved, page 47

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)

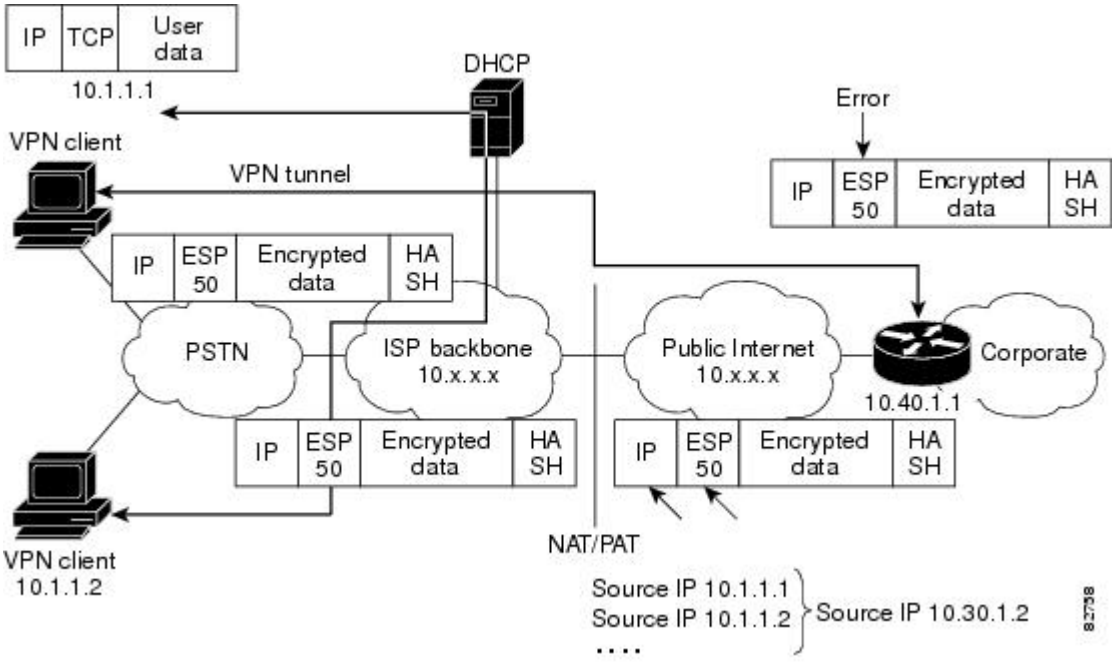
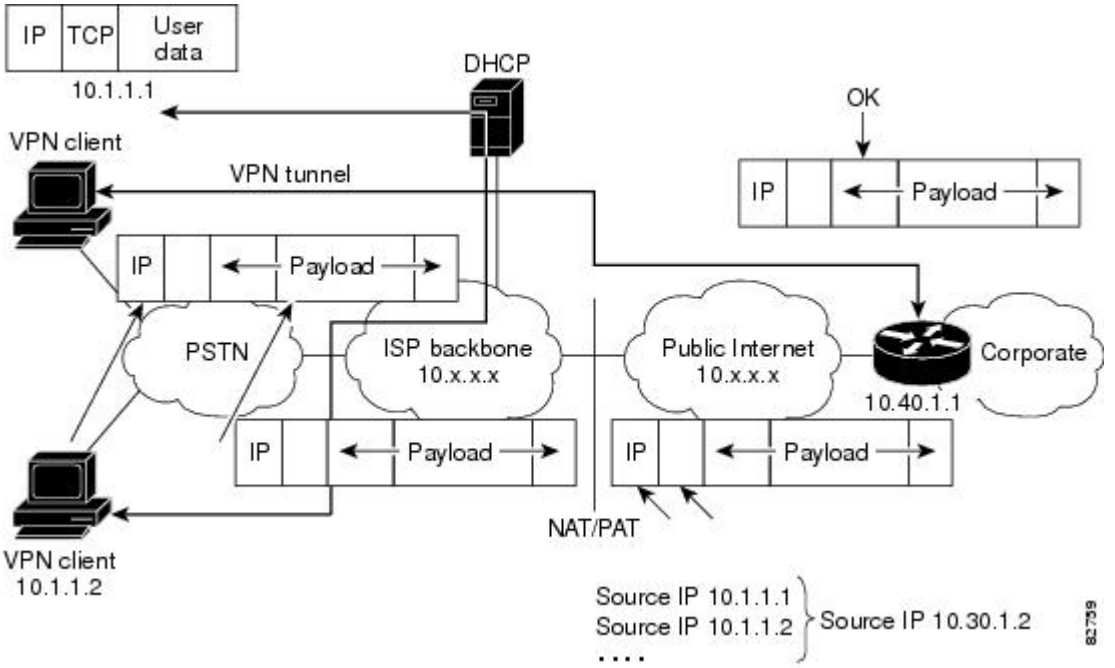


Figure 3 IPsec Packet with UDP Encapsulation



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-IKE marker (which is 8 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. The first figure below shows an IPsec packet before and after transport mode is applied; the second figure below shows an IPsec packet before and after tunnel mode is applied.

Figure 4 Transport Mode--IPsec Packet Before and After ESP Encapsulation

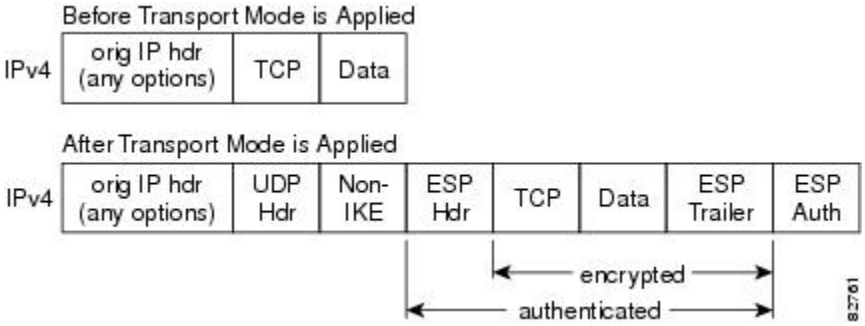
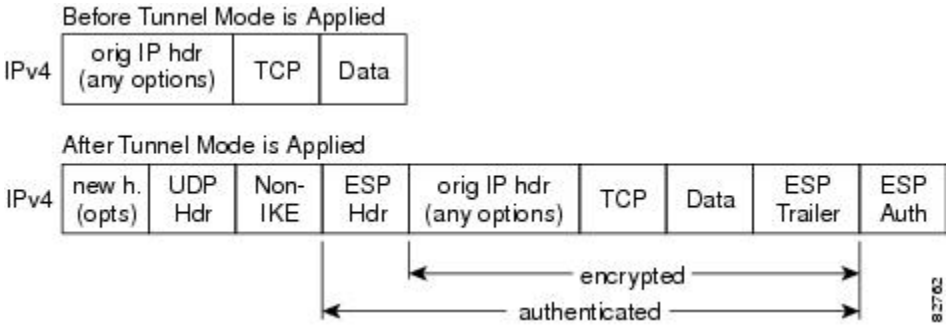


Figure 5 Tunnel Mode--IPsec Packet Before and After ESP Encapsulation



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 (via the **crypto isakmp 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, page 50

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 XE Release 2.1. 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.
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 between 5 to 3,600 seconds.
		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
 authentication pre-share
crypto isakmp key 1234 address 10.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set t2
 match address 101
```

Additional References

The following sections provide references related to the IPsec NAT Transparency feature.

Related Documents

Related Topic	Document Title
Additional NAT configuration tasks	• " Configuring NAT for IP Address Conservation " module in the *Cisco IOS XE IP Addressing Services Configuration Guide* • " Using Application Level Gateways with NAT " module in the *Cisco IOS XE IP Addressing Services Configuration Guide* • " Configuring NAT for High Availability " module in the *Cisco IOS XE IP Addressing Services Configuration Guide* • " Integrating NAT with MPLS VPNs " module in the *Cisco IOS XE IP Addressing Services Configuration Guide*
Additional NAT commands	Cisco IOS IP Addressing Services Command Reference
Additional IPsec configuration tasks	" Configuring Security for VPNs with IPsec " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
Additional IPsec commands	Cisco IOS Security Command Reference
Information on IKE	" Configuring Internet Key Exchange for IPsec VPNs " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
Additional information on IKE dead peer detection	" Easy VPN Server " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*

Standards

Standards	Title
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFCs[1]	Title
RFC 2402	IP Authentication Header
RFC 2406	IP Encapsulating Security Payload (ESP)

Technical Assistance

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

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.

1 Not all supported RFCs are listed.

Table 6 Feature Information for IPsec NAT Transparency

Feature Name	Releases	Feature Information
IPsec NAT Transparency	Cisco IOS XE Release 2.1	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. 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.

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

DF Bit Override Functionality with IPsec Tunnels

The DF Bit Override Functionality with IPsec Tunnels feature allows you 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

Feature Overview

The DF Bit Override Functionality with IPsec Tunnels feature allows you to specify whether your 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 user 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

If your configurations have hosts that prevent you from learning about the available MTU size, you can configure your 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

To set the DF bit for the encapsulating header in tunnel mode, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec df-bit** [**clear** | **set** | **copy**]

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. To set the DF bit for a specified interface, use the **crypto ipsec df-bit** command in interface configuration mode. **Note** DF bit interface configuration settings override all DF bit global configuration settings.

Verifying DF Bit Setting

To verify the current DF Bit settings on your router, use the **show running-config** command in EXEC mode.

Configuration Examples for DB 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 FastEthernet. Thus, all interfaces except FastEthernet will allow the router to send packets larger than the available MTU size; FastEthernet will allow the router to fragment the packet.

```
crypto isakmp policy 1
   hash md5
   authentication pre-share
crypto isakmp key Delaware address 192.168.10.66
crypto isakmp key Key-What-Key address 192.168.11.19
!
!
crypto ipsec transform-set exampleset ah-md5-hmac esp-des
crypto ipsec df-bit clear
!
!
crypto map armadillo 1 ipsec-isakmp
set peer 192.168.10.66
set transform-set exampleset
match address 101
!
crypto map basilisk 1 ipsec-isakmp
set peer 192.168.11.19
set transform-set exampleset
match address 102
!
!
interface FastEthernet
   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 FastEthernet1
   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

The following sections provide references related to the DF Bit Override Functionality with IPsec Tunnels feature.

Related Documents

Related Topic	Document Title
Internet Key Exchange and IPsec networks	Configuring Internet Key Exchange for IPsec VPNs
IPsec network commands	Cisco IOS Security Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

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

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	Cisco IOS XE Release 2.1	This feature allows users 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. The following commands were introduced or modified: **crypto ipsec df-bit**.

IPsec Security Association Idle Timers

When a router running the Cisco IOS XE 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. The IPsec Security Association Idle Timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. Benefits of this feature include:

- Increased availability of resources
- Improved scalability of Cisco IOS XE IPsec deployments. Because this feature prevents the wasting of resources by idle peers, more resources will be available to create new SAs as required.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 the " Configuring Internet Key Exchange Security Protocol " chapter of the *Cisco IOS XE Security Configuration Guide* .

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.
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.

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.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
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
```

Additional References

The following sections provide references related to the IPsec Security Association Idle Timers feature.

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

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	---

Technical Assistance

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

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 8 Feature Information for IPsec Security Association Idle Timers

Feature Name	Releases	Feature Information
IPsec Security Association Idle Timers	Cisco IOS XE Release 2.1	When a router running the Cisco IOS XE 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. The IPsec Security Association Idle Timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. The following command was introduced or modified: **crypto ipsec security-association idle-time**.
	Cisco IOS XE Release 2.1	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 command was introduced or modified: **set security-association idle-time**.