CHAPTER 24

Configuring IKE Features Using the IPsec VPN SPA

This chapter provides information about configuring Internet Key Exchange (IKE) related features using the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

Note For detailed information on Internet Key Exchange (IKE), refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For additional information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xliv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of IKE

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

Note For more detailed information on IKE, refer to the *Cisco IOS Security Configuration Guide.*

IKE automatically negotiates IPsec security associations (SAs) and enables IPsec secure communications without costly manual preconfiguration. Specifically, IKE provides the following benefits:

- Eliminates the need to manually specify all the IPsec security parameters in the crypto maps at both peers.

Note Beginning in Cisco IOS Release 12.2SXH, manual keying is no longer supported.

- Allows you to specify a lifetime for the IPsec security association (SA).
- Allows encryption keys to change during IPsec sessions.
- Allows IPsec to provide anti-replay services.
- Permits certification authority (CA) support for a manageable, scalable IPsec implementation.
- Allows dynamic authentication of peers.

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

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

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

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

Configuring Advanced Encryption Standard in an IKE Policy Map

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

To configure the AES encryption algorithm within an IKE policy map, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto isakmp policy priority`	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. - *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 2	`Router(config-isakmp)# encryption {aes	aes 192	aes 256}`	Specifies the encryption algorithm within an IKE policy. - **aes**—Specifies 128-bit AES as the encryption algorithm. - **aes 192**—Specifies 192-bit AES as the encryption algorithm. - **aes 256**—Specifies 256-bit AES as the encryption algorithm.

Step 3	`...`
	`Router(config-isakmp)# exit`
		Specifies any other policy values appropriate to your configuration, and then exits ISAKMP policy configuration mode.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Verifying the AES IKE Policy

To verify the configuration of the AES IKE policy, enter the **show crypto isakmp policy** command:

```
Router# show crypto isakmp policy

Protection suite of priority 1
encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).
hash algorithm:        Secure Hash Standard
authentication method: Pre-Shared Key
Diffie-Hellman group:  #1 (768 bit)
lifetime: 3600 seconds, no volume limit

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

For an AES configuration example, see the "Advanced Encryption Standard Configuration Example" section on page 24-22.

Configuring ISAKMP Keyrings

A crypto keyring is a collection of preshared and RSA public keys. You can configure a keyring and then associate it with the Internet Security Association and Key Management Protocol (ISAKMP) profile. The crypto ISAKMP profile may contain zero, one, or more than one keyring.

The ISAKMP keyrings feature (also known as the SafeNet IPsec VPN Client Support feature) allows you to use the **local-address** command to limit the scope of an ISAKMP profile or ISAKMP keyring configuration to a local termination address or interface. The benefit of this feature is that different customers can use the same peer identities and ISAKMP keys by using different local termination addresses.

ISAKMP Keyrings Configuration Guidelines and Restrictions

When configuring ISAKMP keyrings, follow these guidelines and restrictions:

- The local address option works only for the primary address of an interface.
- If an IP address is provided, the administrator must ensure that the connection of the peer terminates to the address that is provided.
- If the IP address does not exist on the device, or if the interface does not have an IP address, the ISAKMP profile or ISAKMP keyring will be effectively disabled.

Limiting an ISAKMP Profile to a Local Termination Address or Interface

To configure an ISAKMP profile and limit it to a local termination address or interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto isakmp profile profile-name`	Defines an ISAKMP profile and enters ISAKMP profile configuration mode. • *profile-name*—Name of the ISAKMP profile.
Step 2	`Router(conf-isa-profile)# keyring keyring-name`	(Optional) Configures a keyring with an ISAKMP profile. • *keyring-name*—Name of the crypto keyring. **Note** A keyring is not needed inside an ISAKMP profile for local termination to work. Local termination works even if Rivest, Shamir, and Adelman (RSA) certificates are used.

	Command	Purpose
Step 3	`Router(conf-isa-profile)# match identity address address`	Matches an identity from a peer in an ISAKMP profile. • *address*—IP address of the remote peer.
Step 4	`Router(conf-isa-profile)# local-address {interface-name | ip-address [vrf-tag]}`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface. • *interface-name*—Name of the local interface. • *ip-address*—Local termination address. • *vrf-tag*—(Optional) Scope of the IP address will be limited to the VRF.

Limiting a Keyring to a Local Termination Address or Interface

To configure an ISAKMP keyring and limit its scope to a local termination address or interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# keyring keyring-name`	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring.
Step 2	`Router(conf-keyring)# local-address {interface-name | ip-address [vrf-tag]}`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface. • *interface-name*—Name of the local interface. • *ip-address*—Local termination address. • *vrf-tag*—(Optional) Scope of the IP address will be limited to the VRF.
Step 3	`Router(conf-keyring)# pre-shared-key address address`	Defines a preshared key to be used for IKE authentication. • *address*—IP address.

For complete configuration information for SafeNet IPsec VPN Client Support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gt_scse.html

For ISAKMP keyrings configuration examples, see the "ISAKMP Keyrings Configuration Examples" section on page 24-22.

Configuring Certificate to ISAKMP Profile Mapping

The Certificate to ISAKMP Profile Mapping feature enables you to assign an Internet Security Association and Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate. In addition, this feature allows you to assign a group name to those peers that are assigned an ISAKMP profile.

Note Certificate to ISAKMP Profile Mapping is only supported as of Cisco IOS Release 12.2(33)SXH.

Certificate to ISAKMP Profile Mapping Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring Certificate to ISAKMP Profile Mapping:

- This feature will not be applicable if you use Rivest, Shamir, and Adelman (RSA)-signature or RSA-encryption authentication without certificate exchange. ISAKMP peers must be configured to do RSA-signature or RSA-encryption authentication using certificates.

Mapping the Certificate to the ISAKMP Profile

To map the certificate to the ISAKMP profile, perform the following task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# **crypto isakmp profile** *profile-name*`	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 2	`Router(config-isa-prof)# **match certificate** *certificate-map*`	Accepts the name of a certificate map. • *certificate-map*—Name of the certificate map.

Verifying the Certificate to ISAKMP Profile Mapping Configuration

To verify that the subject name of the certificate map has been properly configured, enter the **show crypto pki certificates** and the **debug crypto isakmp** commands.

The **show crypto pki certificates** command displays all current IKE security associations (SAs) at a peer. The **debug crypto isakmp** command displays messages about IKE events.

The following examples show that a certificate has been mapped to an ISAKMP profile. The examples include the configurations for the responder and initiator, the **show crypto pki certificates** command output verifying that the subject name of the certificate map has been configured, and the **debug crypto isakmp** command output showing that the certificate has gone through certificate map matching and been matched to the ISAKMP profile.

Responder Configuration

```
crypto pki certificate map cert_map 10
! The above line is the certificate map definition.
```

```
 subject-name co ou = green
! The above line shows that the subject name must have "ou = green."
!
crypto isakmp profile certpro
! The above line shows that this is the ISAKMP profile that will match if the certificate
of the peer matches cert_map (shown on third line below).
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
```

Initiator Configuration

```
crypto ca trustpoint LaBcA
 enrollment url http://10.76.82.20:80/cgi-bin/openscep
 subject-name ou=green,c=IN
! The above line ensures that the subject name "ou = green" is set.
 revocation-check none
```

Command Output for show crypto pki certificates for the Initiator

```
Router# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 21
  Certificate Usage: General Purpose
  Issuer:
    cn=blue-lab CA
    o=CISCO
    c=IN
  Subject:
    Name: Router.cisco.com
    c=IN
    ou=green
! The above line is a double check that "ou = green" has been set as the subject name.
    hostname=Router.cisco.com
  Validity Date:
    start date: 14:34:30 UTC Mar 31 2004
    end   date: 14:34:30 UTC Apr 1 2009
    renew date: 00:00:00 UTC Jan 1 1970
  Associated Trustpoints: LaBcA
```

Command Output for debug crypto isakmp for the Responder

```
Router# debug crypto isakmp

*Nov  6 19:31:25.010: ISAKMP:(0): SA request profile is prof2
*Nov  6 19:31:25.010: ISAKMP: Found a peer struct for 14.0.0.2, peer port 500
*Nov  6 19:31:25.010: ISAKMP: Locking peer struct 0x13884FB8, refcount 349 for
isakmp_initiator
*Nov  6 19:31:25.010: ISAKMP[I]: sa->swdb: Vlan3
*Nov  6 19:31:25.010: ISAKMP: local port 500, remote port 500
*Nov  6 19:31:25.010: ISAKMP: set new node 0 to QM_IDLE
*Nov  6 19:31:25.010: ISAKMP: Find a dup sa in the avl tree during calling isadb_insert sa
= 13C041E8
*Nov  6 19:31:25.010: ISAKMP:(0):Can not start Aggressive mode, trying Main mode.
*Nov  6 19:31:25.010: ISAKMP:(0):Profile has no keyring, aborting key search
*Nov  6 19:31:25.010: ISAKMP:(0): constructed NAT-T vendor-07 ID
*Nov  6 19:31:25.010: ISAKMP:(0): constructed NAT-T vendor-03 ID
*Nov  6 19:31:25.010: ISAKMP:(0): constructed NAT-T vendor-02 ID
*Nov  6 19:31:25.010: ISAKMP:(0):Input = IKE_MESG_FROM_IPSEC, IKE_SA_REQ_MM
*Nov  6 19:31:25.010: ISAKMP:(0):Old State = IKE_READY  New State = IKE_I_MM1
```

```
*Nov  6 19:31:25.010: ISAKMP:(0): beginning Main Mode exchange
*Nov  6 19:31:25.010: ISAKMP:(0): sending packet to 14.0.0.2 my_port 500 peer_port 500 (I) 
MM_NO_STATE
*Nov  6 19:31:25.018: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf 
(N) NEW SA
*Nov  6 19:31:25.018: ISAKMP: Found a peer struct for 14.0.0.2, peer port 500
*Nov  6 19:31:25.018: ISAKMP: Locking peer struct 0x13884FB8, refcount 350 for 
crypto_isakmp_process_block
*Nov  6 19:31:25.018: ISAKMP[R]: sa->swdb: Vlan2
*Nov  6 19:31:25.018: ISAKMP: local port 500, remote port 500
*Nov  6 19:31:25.018: ISAKMP: Find a dup sa in the avl tree during calling isadb_insert sa 
= 148C68D8
*Nov  6 19:31:25.018: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.018: ISAKMP:(0):Old State = IKE_READY  New State = IKE_R_MM1

*Nov  6 19:31:25.018: ISAKMP:(0): processing SA payload. message ID = 0
*Nov  6 19:31:25.018: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.018: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.018: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID seems Unity/DPD but major 157 mismatch
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID is NAT-T v3
*Nov  6 19:31:25.018: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID is NAT-T v2
*Nov  6 19:31:25.038: ISAKMP:(0):Checking ISAKMP transform 1 against priority 1 policy
*Nov  6 19:31:25.038: ISAKMP:      encryption 3DES-CBC
*Nov  6 19:31:25.038: ISAKMP:      hash MD5
*Nov  6 19:31:25.038: ISAKMP:      default group 1
*Nov  6 19:31:25.038: ISAKMP:      auth RSA sig
*Nov  6 19:31:25.038: ISAKMP:      life type in seconds
*Nov  6 19:31:25.038: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
*Nov  6 19:31:25.042: ISAKMP:(0):atts are acceptable. Next payload is 3
*Nov  6 19:31:25.042: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.042: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.042: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID seems Unity/DPD but major 157 mismatch
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID is NAT-T v3
*Nov  6 19:31:25.042: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID is NAT-T v2
*Nov  6 19:31:25.042: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.042: ISAKMP:(0):Old State = IKE_R_MM1  New State = IKE_R_MM1

*Nov  6 19:31:25.046: ISAKMP:(0): constructed NAT-T vendor-07 ID
*Nov  6 19:31:25.046: ISAKMP:(0): sending packet to 14.0.0.2 my_port 500 peer_port 500 (R) 
MM_SA_SETUP
*Nov  6 19:31:25.046: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.046: ISAKMP:(0):Old State = IKE_R_MM1  New State = IKE_R_MM2

*Nov  6 19:31:25.046: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf 
(I) MM_NO_STATE
*Nov  6 19:31:25.046: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.046: ISAKMP:(0):Old State = IKE_I_MM1  New State = IKE_I_MM2

*Nov  6 19:31:25.046: ISAKMP:(0): processing SA payload. message ID = 0
*Nov  6 19:31:25.046: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.046: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.046: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.046: ISAKMP : Looking for xauth in profile prof2
*Nov  6 19:31:25.046: ISAKMP:(0):Checking ISAKMP transform 1 against priority 1 policy
*Nov  6 19:31:25.046: ISAKMP:      encryption 3DES-CBC
```

```
*Nov  6 19:31:25.046: ISAKMP:      hash MD5
*Nov  6 19:31:25.046: ISAKMP:      default group 1
*Nov  6 19:31:25.046: ISAKMP:      auth RSA sig
*Nov  6 19:31:25.050: ISAKMP:      life type in seconds
*Nov  6 19:31:25.050: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
*Nov  6 19:31:25.050: ISAKMP:(0):atts are acceptable. Next payload is 0
*Nov  6 19:31:25.050: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.050: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.050: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.050: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.050: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM2

*Nov  6 19:31:25.050: ISAKMP (0): constructing CERT_REQ for issuer
cn=mscavpn1,ou=isbu,o=cisco
*Nov  6 19:31:25.054: ISAKMP:(0): sending packet to 14.0.0.2 my_port 500 peer_port 500 (I)
MM_SA_SETUP
*Nov  6 19:31:25.054: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.054: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM3

*Nov  6 19:31:25.058: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf
(R) MM_SA_SETUP
*Nov  6 19:31:25.062: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.062: ISAKMP:(0):Old State = IKE_R_MM2  New State = IKE_R_MM3

*Nov  6 19:31:25.062: ISAKMP:(0): processing KE payload. message ID = 0
*Nov  6 19:31:25.062: ISAKMP:(0): processing NONCE payload. message ID = 0
*Nov  6 19:31:25.062: ISAKMP:(83727): processing CERT_REQ payload. message ID = 0
*Nov  6 19:31:25.062: ISAKMP:(83727): peer wants a CT_X509_SIGNATURE cert
*Nov  6 19:31:25.066: ISAKMP:(83727): peer want cert issued by cn=mscavpn1,ou=isbu,o=cisco
*Nov  6 19:31:25.066: ISAKMP:(83727): Choosing trustpoint MSCA as issuer
*Nov  6 19:31:25.066: ISAKMP:(83727): processing vendor id payload
*Nov  6 19:31:25.066: ISAKMP:(83727): vendor ID is DPD
*Nov  6 19:31:25.066: ISAKMP:(83727): processing vendor id payload
*Nov  6 19:31:25.066: ISAKMP:(83727): speaking to another IOS box!
*Nov  6 19:31:25.066: ISAKMP:(83727): processing vendor id payload
*Nov  6 19:31:25.066: ISAKMP:(83727): vendor ID seems Unity/DPD but major 230 mismatch
*Nov  6 19:31:25.066: ISAKMP:(83727): vendor ID is XAUTH
*Nov  6 19:31:25.066: ISAKMP (83727): His hash no match - this node outside NAT
*Nov  6 19:31:25.066: ISAKMP (83727): No NAT Found for self or peer
*Nov  6 19:31:25.066: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.066: ISAKMP:(83727):Old State = IKE_R_MM3  New State = IKE_R_MM3

*Nov  6 19:31:25.066: ISAKMP (83727): constructing CERT_REQ for issuer
cn=mscavpn1,ou=isbu,o=cisco
*Nov  6 19:31:25.066: ISAKMP:(83727): sending packet to 14.0.0.2 my_port 500 peer_port 500
(R) MM_KEY_EXCH
*Nov  6 19:31:25.070: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.070: ISAKMP:(83727):Old State = IKE_R_MM3  New State = IKE_R_MM4

*Nov  6 19:31:25.070: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf
(I) MM_SA_SETUP
*Nov  6 19:31:25.070: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.070: ISAKMP:(0):Old State = IKE_I_MM3  New State = IKE_I_MM4

*Nov  6 19:31:25.070: ISAKMP:(0): processing KE payload. message ID = 0
*Nov  6 19:31:25.074: ISAKMP:(0): processing NONCE payload. message ID = 0
*Nov  6 19:31:25.098: ISKAMP: growing send buffer from 1024 to 3072

*Nov  6 19:31:25.118: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500
fvrf (R) MM_KEY_EXCH
*Nov  6 19:31:25.122: ISAKMP:(83727):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.122: ISAKMP:(83727):Old State = IKE_R_MM4  New State = IKE_R_MM5

*Nov  6 19:31:25.122: ISAKMP:(83727): processing ID payload. message ID = 0
```

```
*Nov  6 19:31:25.122: ISAKMP (83727): ID payload
        next-payload : 6
        type         : 3
        USER FQDN    : a@vrf2.com
        protocol     : 17
        port         : 500
        length       : 18
*Nov  6 19:31:25.134: ISAKMP:(83727):: peer matches prof2 profile
*Nov  6 19:31:25.134: ISAKMP:(83727): processing CERT payload. message ID = 0
*Nov  6 19:31:25.134: ISAKMP:(83727): processing a CT_X509_SIGNATURE cert
*Nov  6 19:31:25.142: ISAKMP:(83727): peer's pubkey isn't cached
*Nov  6 19:31:25.158: %CRYPTO-6-IKMP_NO_ID_CERT_USER_FQDN_MATCH: ID of a@vrf2.com (type 3)
and certificate user fqdn with empty
*Nov  6 19:31:25.158: ISAKMP (83727): adding peer's pubkey to cache
*Nov  6 19:31:25.158: ISAKMP:(83727): processing SIG payload. message ID = 0
*Nov  6 19:31:25.162: ISAKMP:(83727):SA authentication status:
        authenticated
*Nov  6 19:31:25.162: ISAKMP:(83727):SA has been authenticated with 14.0.0.2
*Nov  6 19:31:25.162: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.162: ISAKMP:(83727):Old State = IKE_R_MM5  New State = IKE_R_MM5

*Nov  6 19:31:25.170: ISAKMP:(83727):SA is doing RSA signature authentication using id
type ID_USER_FQDN
*Nov  6 19:31:25.170: ISAKMP (83727): ID payload
        next-payload : 6
        type         : 3
        USER FQDN    : a@vrf2.com
        protocol     : 17
        port         : 500
        length       : 18
*Nov  6 19:31:25.170: ISAKMP:(83727):Total payload length: 18
*Nov  6 19:31:25.182: ISAKMP (83727): constructing CERT payload for
cn=HUB,ou=isbu,o=cisco,hostname=HUB.cisco.com,serialNumber=1234D
*Nov  6 19:31:25.182: ISKAMP: growing send buffer from 1024 to 3072
*Nov  6 19:31:25.186: ISAKMP:(83727): using the MSCA trustpoint's keypair to sign
*Nov  6 19:31:25.194: ISAKMP:(83727): sending packet to 14.0.0.2 my_port 500 peer_port 500
(R) MM_KEY_EXCH
*Nov  6 19:31:25.198: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.198: ISAKMP:(83727):Old State = IKE_R_MM5  New State = IKE_P1_COMPLETE

*Nov  6 19:31:25.198: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
*Nov  6 19:31:25.198: ISAKMP:(83727):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Nov  6 19:31:25.238: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500
fvrf (R) QM_IDLE
*Nov  6 19:31:25.238: ISAKMP: set new node -134314170 to QM_IDLE
*Nov  6 19:31:25.242: ISAKMP:(83727): processing HASH payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing SA payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727):Checking IPSec proposal 1
*Nov  6 19:31:25.242: ISAKMP: transform 1, ESP_3DES
*Nov  6 19:31:25.242: ISAKMP:   attributes in transform:
*Nov  6 19:31:25.242: ISAKMP:      encaps is 1 (Tunnel)
*Nov  6 19:31:25.242: ISAKMP:      SA life type in seconds
*Nov  6 19:31:25.242: ISAKMP:      SA life duration (basic) of 3600
*Nov  6 19:31:25.242: ISAKMP:      SA life type in kilobytes
*Nov  6 19:31:25.242: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
*Nov  6 19:31:25.242: ISAKMP:      authenticator is HMAC-SHA
*Nov  6 19:31:25.242: ISAKMP:(83727):atts are acceptable.
*Nov  6 19:31:25.242: ISAKMP:(83727): processing NONCE payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing ID payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing ID payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727):QM Responder gets spi
```

```
*Nov  6 19:31:25.242: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_FROM_PEER, 
IKE_QM_EXCH
*Nov  6 19:31:25.242: ISAKMP:(83727):Old State = IKE_QM_READY  New State = 
IKE_QM_SPI_STARVE
*Nov  6 19:31:25.242: ISAKMP:(83727): Creating IPSec SAs
*Nov  6 19:31:25.246:         inbound SA from 14.0.0.2 to 15.0.0.2 (f/i)  1/714
        (proxy 12.0.0.2 to 13.0.0.2)
*Nov  6 19:31:25.246:         has spi 0x917AD879 and conn_id 0
*Nov  6 19:31:25.246:         lifetime of 3600 seconds
*Nov  6 19:31:25.246:         lifetime of 4608000 kilobytes
*Nov  6 19:31:25.246:         outbound SA from 15.0.0.2 to 14.0.0.2 (f/i) 1/714
        (proxy 13.0.0.2 to 12.0.0.2)
*Nov  6 19:31:25.246:         has spi  0xC54A5A05 and conn_id 0
*Nov  6 19:31:25.246:         lifetime of 3600 seconds
*Nov  6 19:31:25.246:         lifetime of 4608000 kilobytes
*Nov  6 19:31:25.246:  ISAKMP: Failed to find peer index node to update peer_info_list
*Nov  6 19:31:25.250: ISAKMP:(83727): sending packet to 14.0.0.2 my_port 500 peer_port 500 
(R) QM_IDLE
*Nov  6 19:31:25.250: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_INTERNAL, 
IKE_GOT_SPI
*Nov  6 19:31:25.250: ISAKMP:(83727):Old State = IKE_QM_SPI_STARVE  New State = 
IKE_QM_R_QM2
*Nov  6 19:31:25.270: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500 
fvrf (R) QM_IDLE
*Nov  6 19:31:25.274: ISAKMP:(83727):deleting node -134314170 error FALSE reason "QM done 
(await)"
*Nov  6 19:31:25.274: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_FROM_PEER, 
IKE_QM_EXCH
*Nov  6 19:31:25.274: ISAKMP:(83727):Old State = IKE_QM_R_QM2  New State = 
IKE_QM_PHASE2_COMPLETE
*Nov  6 19:32:15.282: ISAKMP:(83727):purging node -134314170
```

Command Output for show crypto isakmp sa [detail] for the Responder

```
Router# show crypto isakmp sa vrf vrf2
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
15.0.0.2        14.0.0.2        QM_IDLE          83727 ACTIVE prof2

IPv6 Crypto ISAKMP SA

Router# show crypto isakmp sa detail vrf vrf2
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local           Remote          I-VRF    Status Encr Hash Auth DH Lifetime Cap.

83727 15.0.0.2        14.0.0.2        vrf2     ACTIVE 3des md5  rsig 1  23:59:15
       Engine-id:Conn-id =  :15727

IPv6 Crypto ISAKMP SA
```

Assigning the Group Name to the Peer

To associate a group name with an ISAKMP profile that will be assigned to a peer, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto isakmp profile`** *`profile-name`*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 2	`Router (conf-isa-prof)#` **`client configuration group`** *`group-name`*	Accepts the name of a group that will be assigned to a peer when the peer is assigned this crypto ISAKMP profile. • *group-name*—Name of the group to be associated with the peer.

Verifying the Group Name to Peer Assignation Configuration

To verify that a group has been assigned to a peer, enter the **debug crypto isakmp** command.

The **debug crypto isakmp** command displays messages about IKE events.

The following **debug crypto isakmp** output shows that the peer has been matched to the ISAKMP profile named "certpro" and that it has been assigned a group named "new_group."

Initiator Configuration

```
crypto isakmp profile certpro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   client configuration group new_group
! The statement on the above line will assign the group "new_group" to any peer that
matches the ISAKMP profile "certpro."
   initiate mode aggressive
```

Command Output for debug crypto isakmp for the Responder

```
Router# debug crypto isakmp
6d23h: ISAKMP (0:268435461): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:5:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:5:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5
6d23h: ISAKMP:(0:5:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435461): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:5:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:5:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:5:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:5:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:5:HW:2): OU = green
6d23h: ISAKMP:(0:5:HW:2): certificate map matches certpro profile
6d23h: ISAKMP:(0:5:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:5:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:5:HW:2): CERT validity confirmed.
6d23h: ISAKMP:(0:5:HW:2):Profile has no keyring, aborting key search
6d23h: ISAKMP:(0:5:HW:2): Profile certpro assigned peer the group named new_group
```

For complete configuration information for certificate to ISAKMP profile mapping, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t8/feature/guide/gt_isakp.html

For certificate to ISAKMP profile mapping configuration examples, see the "Certificate to ISAKMP Profile Mapping Configuration Examples" section on page 24-23.

Configuring an Encrypted Preshared Key

The Encrypted Preshared Key feature allows you to securely store plain text passwords in type 6 (encrypted) format in NVRAM.

Encrypted Preshared Key Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring an encrypted preshared key:

- Old ROM monitors (ROMMONs) and boot images cannot recognize the new type 6 passwords. If you boot from an old ROMMON, you can expect errors.
- If the password (master key) is changed, or reencrypted, using the **key config-key password-encryption** command, the list registry passes the old key and the new key to the application modules that are using type 6 encryption.
- If the master key that was configured using the **key config-key password-encryption** command is deleted from the system, a warning is printed (and a confirm prompt is issued) that states that all type 6 passwords will become useless. As a security measure, after the passwords have been encrypted, they will never be decrypted in the Cisco IOS software. However, passwords can be reencrypted.

Caution If the password configured using the **key config-key password-encryption** command is lost, it cannot be recovered. The password should be stored in a safe location.

- If you later unconfigure password encryption using the **no password encryption aes** command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption** command exists, the type 6 passwords will be decrypted as and when required by the application.
- Because no one can "read" the password (configured using the **key config-key password-encryption** command), there is no way that the password can be retrieved from the switch. Existing management stations cannot "know" what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone,

meaning that they cannot be loaded onto a switch. Before or after the configurations are loaded onto a switch, the password must be manually added (using the **key config-key password-encryption** command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.

- If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but the following alert message is printed:

```
ciphertext>[for username bar>] is incompatible with the configured master key
```

- If a new master key is configured, all the plain keys are encrypted and made type 6 keys. The existing type 6 keys are not encrypted. The existing type 6 keys are left as is.
- If the old master key is lost or unknown, you have the option of deleting the master key using the **no key config-key password-encryption** command. Deleting the master key using the **no key config-key password-encryption** command causes the existing encrypted passwords to remain encrypted in the switch configuration. The passwords will not be decrypted.

Configuring an Encrypted Preshared Key

To configure an encrypted preshared key, perform the following task beginning global configuration mode:

	Command	Purpose

Step 1	`Router(config)#` **`key config-key password-encryption`**	Stores a type 6 encryption key in private NVRAM.
		Note the following:
		• If you are entering the key interactively (using the **Enter** key) and an encrypted key already exists, you will be prompted for the following:
		`Old key, New key, and Confirm key`
		• If you are entering the key interactively but an encryption key is not present, you will be prompted for the following:
		`New key and Confirm key`
		• If you are removing a password that is already encrypted, you will see the following prompt:
		`WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:`

Step 2	`Router(config)#` **`password-encryption aes`**	Enables the encrypted preshared key.

Verifying the Encrypted Preshared Key Configuration

To verify that a new master key has been configured and that the keys have been encrypted with the new master key, enter the **password logging** command. The following is an example of its output:

```
Router(config)# password logging
Router(config)# key config-key password-encrypt

New key:
Confirm key:
Router(config)#
01:40:57: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master keypas

Router(config)# key config-key password-encrypt
Old key:
New key:
Confirm key:
Router (config)#
01:42:11: TYPE6_PASS: Master key change heralded, re-encrypting the keys
with the new master key
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
```

For complete configuration information for the Encrypted Preshared Key feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t2/feature/guide/gt_epsk.html

For an encrypted preshared key configuration example, see the "Encrypted Preshared Key Configuration Example" section on page 24-23.

Configuring Call Admission Control for IKE

Call Admission Control (CAC) for IKE allows you to limit the number of simultaneous IKE security associations (SAs) that a switch can establish.

Note Call Admission Control is supported in Cisco IOS Release 12.2(33)SXH and later releases.

There are two ways to limit the number of IKE SAs that a switch can establish to or from another switch:

- Configure an absolute IKE SA limit by entering the **crypto call admission limit** command.

  When an IKE SA limit is defined, the switch no longer accepts or initiates new IKE SA requests when this value has been reached as follows: When there is a new SA request from a peer switch, IKE determines if the number of active IKE SAs plus the number of SAs being negotiated meets or exceeds the configured SA limit. If the number is greater than or equal to the limit, the new SA request is rejected and a syslog is generated. This log contains the source destination IP address of the SA request.

- Configure a system resource limit by entering the **call admission limit** command.

  When a system resource limit is defined, the switch no longer accepts or initiates new IKE SA requests when the specified level of system resources is being used as follows: Call Admission Control (CAC) polls a global resource monitor so that IKE knows when the switch is running short of CPU cycles or memory buffers. You can configure a resource limit, from 1 to 100000, that represents a level of system resources. When that level of the system resources is being used, IKE no longer accepts or initiates new IKE SA requests.

CAC is applied to new SAs (that is, when an SA does not already exist between the peers) and rekeying SAs. Every effort is made to preserve existing SAs. Only new SA requests will ever be denied due to a lack of system resources or because the configured IKE SA limit has been reached.

Configuring the IKE Security Association Limit

To configure an IKE Security Association limit, perform the following steps beginning in global configuration mode. When an IKE SA limit is defined, the switch no longer accepts or initiates new IKE SA requests when the limit has been reached:

	Command	Purpose
Step 1	`Router(config)# crypto call admission limit {ike {sa number | in-negotiation-sa number}}`	Specifies the maximum number of IKE SAs that the switch can establish before IKE no longer accepts or initiates new SA requests. • **sa** *number*—Number of active IKE SAs allowed on the switch. The range is 0 to 99999. • **in-negotiation-sa** *number*—Number of in-negotiation IKE SAs allowed on the switch. The range is 10 to 99999. **Note** An ISAKMP connection needs to be built in two directions. If you have 500 spokes in your network, you should set this value at a minimum of 1000 (500 x 2).
Step 2	`Router(config)# exit`	Returns to privileged EXEC mode.

Configuring a System Resource Limit

To configure a system resource limit, perform the following steps beginning in global configuration mode. When an IKE SA limit is defined, the switch no longer accepts or initiates new IKE SA requests when the specified level of system resources is being used.

	Command	Purpose
Step 1	`Router(config)# call admission limit charge`	Instructs IKE to stop initiating or accepting new SA requests (that is, calls for CAC) when the specified level of system resources is being used. • *charge*—Level of the system resources that, when used, causes IKE to stop accepting new SA requests. Valid values are 1 to 100000.
Step 2	`Router(config)# exit`	Returns to privileged EXEC mode.

Clearing Call Admission Statistics

To clear the Call Admission Control counters that track the number of accepted and rejected Internet Key Exchange (IKE) requests, use the **clear crypto call admission statistics** command in global configuration mode:

```
Router(config)# clear crypto call admission statistics
```

Verifying the Call Admission Control for IKE Configuration

To verify that Call Admission Control has been configured, enter the **show call admission statistics** and the **show crypto call admission statistics** commands.

The **show call admission statistics** command monitors the global CAC configuration parameters and the behavior of CAC.

```
Router# show call admission statistics
Total Call admission charges: 0, limit 25
Total calls rejected 12, accepted 51
Load metric: charge 0, unscaled 0
```

The **show crypto call admission statistics** command monitors crypto CAC statistics.

```
Router# show crypto call admission statistics
-----------------------------------------------------------
               Crypto Call Admission Control Statistics
-----------------------------------------------------------
System Resource Limit: 0    Max IKE SAs 0
Total IKE SA Count:    0    active:     0   negotiating: 0
Incoming IKE Requests: 0    accepted:   0   rejected:    0
Outgoing IKE Requests: 0    accepted:   0   rejected:    0
Rejected IKE Requests: 0    rsrc low:   0   SA limit:    0
```

For more complete configuration information for Call Admission Control for IKE, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t8/feature/guide/gtcallik.html

For Call Admission Control for IKE configuration examples, see the "Call Admission Control for IKE Configuration Examples" section on page 24-24.

Configuring Dead Peer Detection

Dead Peer Detection (DPD), defined in RFC 3706, is a mechanism used to detect dead IPsec peers. IPsec is a peer-to-peer type of technology. It is possible that IP connectivity may be lost between peers due to routing problems, peer reloading, or some other situation. This lost connectivity can result in black holes where traffic is lost. DPD, based on a traffic-detection method, is one possible mechanism to remedy this situation.

Note The **periodic** option of the **crypto isakmp keepalive** command is only supported as of Cisco IOS Release 12.2(33)SXH; the **on-demand** option is supported in all releases.

DPD supports two options: on-demand or periodic. The on-demand approach is the default. With on-demand DPD, messages are sent on the basis of traffic patterns. For example, if a switch must send outbound traffic and the liveliness of the peer is questionable, the switch sends a DPD message to query the status of the peer. If a switch has no traffic to send, it never sends a DPD message. If a peer is dead, and the switch never has any traffic to send to the peer, the switch 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 switch is not trying to communicate with the peer). On the other hand, if the switch has traffic to send to the peer, and the peer does not respond, the switch will initiate a DPD message to determine the state of the peer.

With the periodic option, you can configure your switch so that DPD messages are forced at regular intervals. This forced approach results in earlier detection of dead peers. For example, if a switch has no traffic to send, a DPD message is still sent at regular intervals, and if a peer is dead, the switch does not have to wait until the IKE SA times out to find out.

DPD is configured using the **crypto isakmp keepalive** command. DPD and Cisco IOS keepalives function on the basis of a timer. If the timer is set for 10 seconds, the switch will send a hello message every 10 seconds (unless, of course, the switch receives a hello message from the peer). The benefit of Cisco IOS keepalives and periodic DPD is earlier detection of dead peers. However, Cisco 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 and Cisco IOS keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the switch to detect a dead IKE peer, and when the switch detects the dead state, the switch deletes the IPsec and IKE SAs to the peer. If you configure multiple peers, the switch will switch over to the next listed peer for a stateless failover.

DPD Configuration Guidelines and Restrictions

When configuring DPD, follow these guidelines and restrictions:

- 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.
- If you do not configure the **periodic** option using the **crypto isakmp keepalive** command, the switch defaults to the **on-demand** approach.
- Before configuring periodic DPD, you should ensure that your IKE peer supports DPD. Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS software in all modes of operation—site-to-site, Easy VPN remote, and Easy VPN server.
- Using periodic DPD potentially allows the switch 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.
- When you configure DPD using the **crypto isakmp keepalive** *seconds* command, the *seconds* argument specifies the interval between DPD messages. In the case of on-demand DPD, the actual interval may be up to twice the configured value.

Configuring a Dead Peer Detection Message

To allow the switch to send DPD messages to the peer, perform the following task:

Command	Purpose
`Router# crypto isakmp keepalive seconds [retries] [periodic | on-demand]`	Converts Switch 1 to standalone mode. • *seconds*—Specifies the number of seconds between DPD messages; the range is from 10 to 3600 seconds. • *retries*—(Optional) Specifies the number of seconds between DPD retries if the DPD message fails; the range is from 2 to 60 seconds. If unspecified, the default is 2 seconds. • **periodic**—(Optional) Specifies that the DPD messages are sent at regular intervals. • **on-demand**—(Optional) Specifies that DPD retries are sent on demand. This is the default behavior.

Note Because the **on-demand** option is the default, the **on-demand** keyword does not appear in configuration output.

Verifying the DPD Configuration

To verify that DPD is enabled, use the **show crypto isakmp sa detail** command in global mode:

```
Router# show crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption

C-id  Local           Remote          I-VRF    Encr Hash Auth DH Lifetime Cap.
273   11.0.0.2        11.0.0.1        ivrf21   3des sha  psk  2  01:59:35 D
       Connection-id:Engine-id =  273:2(hardware)
```

For more complete configuration information for Cisco IOS Dead Peer Detection (DPD) support, refer to the *Cisco IOS Security Command Reference, Release 12.3*.

For DPD configuration examples, see the "Dead Peer Detection Configuration Examples" section on page 24-24.

Understanding IPsec NAT Transparency

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

Before the introduction of this feature, a standard IPsec virtual private network (VPN) tunnel would not work if there were one or more NAT or PAT points in the delivery path of the IPsec packet. This feature allows IPsec to operate through a NAT/PAT device.

For detailed information on NAT Transparency, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftipsnat.html

IPsec NAT Transparency Configuration Guidelines and Restrictions

When configuring IPsec NAT transparency, follow these guidelines and restrictions:

- For non-GRE over IPsec configurations, NAT transparency is supported in both tunnel and transport modes.
- For point-to-point GRE over IPsec configurations, NAT transparency is supported only in tunnel mode.
- For DMVPN configurations, NAT transparency is supported only in transport mode.

Configuring NAT Transparency

NAT transparency is a feature that is auto-detected by the IPsec VPN SPA. There are no configuration steps. If both VPN devices are NAT transparency-capable, NAT transparency is auto-detected and auto-negotiated.

Disabling NAT Transparency

You might want to disable NAT transparency if you already know that your network uses IPsec-awareness NAT (SPI-matching scheme). To disable NAT transparency, use the following command in global configuration mode:

```
Router(config)# no crypto ipsec nat-transparency udp-encapsulation
```

Configuring NAT Keepalives

By default, the NAT keepalive feature is disabled. To configure your switch to send NAT keepalive packets, enter the **crypto isakmp nat keepalive** command in global configuration mode:

```
Router(config)# crypto isakmp nat keepalive seconds
```

In this command, *seconds* specifies the number of seconds between keepalive packets; range is between 5 to 3,600 seconds.

For a NAT keepalive configuration example, see the "ISAKMP NAT Keepalive Configuration Example" section on page 24-24.

Verifying the NAT Configuration

To verify the NAT configuration, enter the **show crypto ipsec sa** command:

Note When you first enter the **show crypto ipsec sa** command, the packet counters may not show the correct values. Repeat the command to show the updated values.

```
Router# show crypto ipsec sa

interface:GigabitEthernet5/0/1
    Crypto map tag:testtag, local addr. 10.2.80.161

    local ident (addr/mask/prot/port):(10.2.80.161/255.255.255.255/0/0)
    remote ident (addr/mask/prot/port):(100.0.0.1/255.255.255.255/0/0)
    current_peer:100.0.0.1:4500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps:109, #pkts encrypt:109, #pkts digest 109
    #pkts decaps:109, #pkts decrypt:109, #pkts verify 109
    #pkts compressed:0, #pkts decompressed:0
    #pkts not compressed:0, #pkts compr. failed:0, #pkts decompress failed:0
    #send errors 90, #recv errors 0

    local crypto endpt.:10.2.80.161, remote crypto endpt.:100.0.0.1:4500
    path mtu 1500, media mtu 1500
    current outbound spi:23945537

    inbound esp sas:
    spi:0xF423E273(4095992435)
    transform:esp-des esp-sha-hmac ,
    in use settings ={Tunnel UDP-Encaps, }
    slot:0, conn id:200, flow_id:1, crypto map:testtag
    sa timing:remaining key lifetime (k/sec):(4607996/2546)
    IV size:8 bytes
    replay detection support:Y

    inbound ah sas:

    inbound pcp sas:

    outbound esp sas:
    spi:0x23945537(596923703)
    transform:esp-des esp-sha-hmac ,
    in use settings ={Tunnel UDP-Encaps, }
    slot:0, conn id:201, flow_id:2, crypto map:testtag
    sa timing:remaining key lifetime (k/sec):(4607998/2519)
    IV size:8 bytes
    replay detection support:Y

    outbound ah sas:

    outbound pcp sas:
```

For complete configuration information for Cisco IOS IPsec NAT transparency support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftipsnat.html

Configuration Examples

This section provides examples of the following configurations:

- Advanced Encryption Standard Configuration Example, page 24-22
- ISAKMP Keyrings Configuration Examples, page 24-22
- Certificate to ISAKMP Profile Mapping Configuration Examples, page 24-23
- Encrypted Preshared Key Configuration Example, page 24-23
- Call Admission Control for IKE Configuration Examples, page 24-24
- Dead Peer Detection Configuration Examples, page 24-24
- ISAKMP NAT Keepalive Configuration Example, page 24-24

Advanced Encryption Standard Configuration Example

The following example configures the Advanced Encryption Standard (AES) 256-bit key:

```
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
```

ISAKMP Keyrings Configuration Examples

The following examples show how to limit the scope of an Internet Security Association and Key Management Protocol (ISAKMP) profile or ISAKMP keyring configuration to a local termination address or interface:

- ISAKMP Profile Bound to a Local Interface Configuration Example, page 24-22
- ISAKMP Keyring Bound to a Local Interface Configuration Example, page 24-22
- ISAKMP Keyring Bound to a Local IP Address Configuration Example, page 24-23

ISAKMP Profile Bound to a Local Interface Configuration Example

The following example configures an ISAKMP profile bound to a local interface:

```
crypto isakmp profile prof1
   keyring key0
   match identity address 11.0.0.2 255.255.255.255
   local-address serial2/0
```

ISAKMP Keyring Bound to a Local Interface Configuration Example

The following example configures an ISAKMP keyring bound only to interface serial2/0:

```
crypto keyring key0
  local-address serial2/0
  pre-shared-key address 11.0.0.2 key 12345
```

ISAKMP Keyring Bound to a Local IP Address Configuration Example

The following example configures an ISAKMP keyring bound only to IP address 11.0.0.1:

```
crypto keyring key0
  local-address 11.0.0.1
  pre-shared-key address 11.0.0.2 key 12345
```

Certificate to ISAKMP Profile Mapping Configuration Examples

The following examples show how to configure Certificate to ISAKMP Profile Mapping:

- Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Configuration Example, page 24-23
- Group Name Assigned to a Peer Associated with an ISAKMP Profile Configuration Example, page 24-23

Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Configuration Example

The following example shows that whenever a certificate contains "ou = green," the ISAKMP profile "cert_pro" will be assigned to the peer:

```
crypto pki certificate map cert_map 10
 subject-name co ou = green
!
crypto isakmp identity dn
crypto isakmp profile cert_pro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
```

Group Name Assigned to a Peer Associated with an ISAKMP Profile Configuration Example

The following example shows that the group "some_group" is to be associated with a peer that has been assigned an ISAKMP profile:

```
crypto isakmp profile id_profile
   ca trust-point 2315
   match identity host domain cisco.com

client configuration group some_group
```

Encrypted Preshared Key Configuration Example

The following example shows a configuration for which a type 6 preshared key has been encrypted:

```
Router(config)# password encryption aes
Router(config)# key config-key password-encrypt
New key:
Confirm key:
Router(config)#
0:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Router(config)# exit
```

Call Admission Control for IKE Configuration Examples

The following examples show how to configure Call Admission Control (CAC) for IKE:

- IKE Security Association Limit Configuration Example, page 24-24
- System Resource Limit Configuration Example, page 24-24

IKE Security Association Limit Configuration Example

The following example shows how to specify that there can be a maximum of 25 SAs before IKE starts rejecting new SA requests:

```
Router(config)# crypto call admission limit ike sa 25
```

System Resource Limit Configuration Example

The following example shows how to specify that IKE should drop SA requests when a given level of system resources are being used:

```
Router(config)# call admission limit 50000
```

Dead Peer Detection Configuration Examples

The following examples show how to configure Dead Peer Detection (DPD):

- On-Demand DPD Configuration Example, page 24-24
- Periodic DPD Configuration Example, page 24-24

On-Demand DPD Configuration Example

The following example shows how to configure on-demand DPD messages. In this example, DPD messages will be sent every 60 seconds and every 5 seconds between retries if the peer does not respond:

```
Router(config)# crypto isakmp keepalive 60 5
```

Periodic DPD Configuration Example

The following example shows how to configure periodic DPD messages. In this example, DPD messages are to be sent at intervals of 10 seconds:

```
Router(config)# crypto isakmp keepalive 10 periodic
```

ISAKMP NAT Keepalive 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 56.0.0.1
crypto isakmp nat keepalive 20
```

```
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 56.0.0.1
 set transform-set t2
 match address 101
```