CHAPTER 8

Configuring Enhanced IPsec Features

This chapter provides information about configuring enhanced IPsec features using the VSPA on the Catalyst 6500 Series switch. It includes the following sections:

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, see the *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

For more information about the commands used in this chapter, see the *Cisco IOS Security Command Reference* at this URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html

Also refer to the related Cisco IOS Release 12.2 software configuration guide, command reference, and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xvi.

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

Overview of Enhanced IPsec Features

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF). It 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.

This chapter describes the advanced IPsec features that can be used to improve scalability and performance of your IPsec VPN.

Configuring Advanced Encryption Standard in a Transform Set

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 a transform set, perform this task beginning in global configuration mode:

Command	Purpose
`Router(config)# crypto ipsec transform-set` *transform-set-name* *transform1*`[`*transform2*`[`*transform3*`]]` `...`	Specifies a transform set and IPsec security profiles and algorithms.

transform-set-name specifies the name of the transform set.

transform1[*transform2*[*transform3*]] defines IPsec security protocols and algorithms. To configure AES, you must choose from the following AES Encapsulating Security Payload (ESP) encryption transforms:

- **esp-aes** specifies ESP with the 128-bit AES encryption algorithm.
- **esp-aes 192** specifies ESP with the 192-bit AES encryption algorithm.
- **esp-aes 256** specifies ESP with the 256-bit AES encryption algorithm.

For other accepted transform values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference.*

Verifying the AES Transform Set

To verify the configuration of the transform set, enter the **show crypto ipsec transform-set** command:

```
Router# show crypto ipsec transform-set

Transform set transform-1:{esp-256-aes esp-md5-hmac}
will negotiate = {Tunnel, }
```

For more complete configuration information about AES support, refer to this URL:

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

For an AES configuration example, see the "Advanced Encryption Standard Configuration Example" section on page 8-20.

Configuring Reverse Route Injection

Reverse Route Injection (RRI) provides the ability for static routes to be automatically inserted into the routing process for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote Virtual Private Network (VPN) router as the next hop, the traffic is forced through the crypto process to be encrypted.

After the static route is created on the VPN router, this information is propagated to upstream devices, allowing them to determine the appropriate VPN router to which to send returning traffic in order to maintain IPsec state flows. Being able to determine the appropriate VPN router is particularly useful if multiple VPN routers are used at a site to provide load balancing or failover or if the remote VPN devices are not accessible via a default route. Routes are created in either the global routing table or the appropriate virtual routing and forwarding (VRF) table.

RRI is applied on a per-crypto map basis, whether this is via a static crypto map or a dynamic crypto map template. For both dynamic and static maps, routes are created only at the time of IPsec SA creation. Routes are removed when the SAs are deleted. The **static** keyword can be added to the **reverse-route** command if routes are created on the basis of the content of the crypto ACLs that are permanently attached to the static crypto map.

RRI Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring RRI:

- IP routing should be enabled and static routes should be redistributed if dynamic routing protocols are to be used to propagate RRI-generated static routes.
- You can specify an interface or address as the explicit next hop to the remote VPN device. This functionality allows the overriding of a default route to properly direct outgoing encrypted packets.
- You can add a route tag value to any routes that are created using RRI. This route tag allows redistribution of groups of routes using route maps, allowing you to be selective about which routes enter your global routing table.
- RRI can be configured on the same crypto map that is applied to multiple router interfaces.
- The **reverse-route remote-peer** [**static**] command creates two routes. One route is the standard remote proxy ID and the next hop is the remote VPN client tunnel address. The second route is the actual route to that remote tunnel endpoint and is used when a recursive lookup requires that the remote endpoint be reachable by the next hop. Creation of the second route for the actual next hop is important in the VRF case in which a default route must be overridden by a more explicit route.

  To reduce the number of routes created and support some platforms that do not readily facilitate route recursion, the **reverse-route** {*ip-address*} [**static**] keyword can be used to create one route only.
- For devices using a VSPA, reverse route specifies the next hop to be the interface, subinterface, or virtual LAN (VLAN) with the crypto map applied to it.

Configuring RRI Under a Static Crypto Map

To configure RRI under a static crypto map, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-name ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 2	`Router(config-crypto-map)# reverse-route [[static] | tag tag-id [static] | remote-peer [static] | remote-peer ip-address [static]]`	Creates source proxy information for a crypto map entry. • **static**—(Optional) Creates permanent routes based on static ACLs. • **tag** *tag-id*—(Optional) Tag value that can be used as a match value for controlling redistribution via route maps. • **remote-peer** [**static**]—(Optional) Two routes are created, one for the remote endpoint and one for route recursion to the remote endpoint via the interface to which the crypto map is applied. The **static** keyword is optional. • **remote-peer** *ip-address* [**static**]—(Optional) One route is created to a remote proxy by way of a user-defined next hop. This next hop can be used to override a default route. The *ip-address* argument is required. The **static** keyword is optional.

Configuring RRI Under a Dynamic Crypto Map

To configure RRI under a dynamic crypto map, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto dynamic-map {dynamic-map-name} {dynamic-seq-name}`	Creates a dynamic crypto map entry and enters crypto map configuration mode. • *dynamic-map-name*—Name that identifies the map set. • *dynamic-seq-num*—Sequence number assigned to the crypto map entry.
Step 2	`Router(config-crypto-map)# reverse-route [tag tag-id | remote-peer | remote-peer ip-address]`	Creates source proxy information for a crypto map entry. • **tag** *tag-id*—(Optional) Tag value that can be used as a match value for controlling redistribution via route maps. • **remote-peer**—(Optional) Two routes are created, one for the remote endpoint and one for route recursion to the remote endpoint via the interface to which the crypto map is applied. • **remote-peer** *ip-address*—(Optional) One route is created to a remote proxy by way of a user-defined next hop. This next hop can be used to override a default route. The *ip-address* argument is required.

For more complete configuration information for RRI, see the following URL:

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

For RRI configuration examples, see the "Reverse Route Injection Configuration Examples" section on page 8-20.

Configuring the IPsec Anti-Replay Window Size

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 of the decryptor. Any packet with a sequence number less than X minus N is discarded. Currently, N is set at 64.

At times, 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 are not replayed packets. The IPsec anti-replay window size feature allows you to expand the window size so that sequence number information can be kept for more than 64 packets.

Expanding the IPsec Anti-Replay Window Size Globally

To expand the IPsec anti-replay window globally so that it affects all SAs that are created (except for those that are specifically overridden on a per-crypto map basis), perform this task beginning in global configuration mode:

Command	Purpose
`Router(config)# crypto ipsec security-association replay window size [size]`	Expands the IPsec anti-replay window globally to the specified *size*. • *size*—(Optional) Size of the window. Values can be 64, 128, 256, 512, or 1024. This value becomes the default value.

Expanding the IPsec Anti-Replay Window at the Crypto Map Level

To expand the IPsec anti-replay window on a crypto map basis so that it affects those SAs that have been created using a specific crypto map or profile, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 2	`Router(config-crypto-map)# set security-association replay window size [size]`	Controls the SAs that are created using the policy specified by a particular crypto map, dynamic crypto map, or crypto profile. • *size*—(Optional) Size of the window. Values can be 64, 128, 256, 512, or 1024. This value becomes the default value.

Verifying the IPsec Anti-Replay Window Size Configuration at the Crypto Map Level

To verify that IPsec anti-replay window size is enabled at a crypto map, enter the **show crypto map** command for that particular map. If anti-replay window size is enabled, the display will indicate that it is enabled and indicate the configured window size. If anti-replay window size is disabled, the results will indicate that also.

The following example indicates that IPsec anti-replay window size is enabled:

```
Router# show crypto map tag TESTMAP
```

```
Crypto Map "TESTMAP" 10 ipsec-isakmp
        WARNING: This crypto map is in an incomplete state!
                (missing peer or access-list definitions)
        No matching address list set.
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
        }
        Antireplay window size = 128
        Interfaces using crypto map TESTMAP:
```

For more complete configuration information for IPsec anti-replay window size, refer to the following URL:

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

For IPsec anti-replay window size configuration examples, see the "IPsec Anti-Replay Window Size Configuration Examples" section on page 8-21.

Note Anti-replay failures detected by the VSPA can be caused by reordering, requeueing, or fragmentation elsewhere in the network. As a defense against man-in-the-middle attacks, the VSPA will drop these packets. This is the expected behavior.

Disabling the IPsec Anti-Replay Checking

To disable the IPsec anti-replay checking, enter the **crypto ipsec security-association replay disable** command in global configuration mode as follows:

Command	Purpose
`Router(config)# crypto ipsec security-association replay disable`	Disables the IPsec anti-replay checking.

To disable the IPsec anti-replay checking on a particular crypto map, enter the **set security-association replay disable** command in crypto map configuration mode as follows:

	Command	Purpose
Step 1	`Router(config)# crypto map` *map-name seq-num* `ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 2	`Router(config-crypto-map)# set security-association replay disable`	Disables IPsec anti-replay checking by a particular crypto map, dynamic crypto map, or crypto profile.

Configuring an IPsec Preferred Peer

The IP Security (IPsec) Preferred Peer feature allows you to control the circumstances by which multiple peers on a crypto map are tried in a failover scenario. If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If all connections to the current peer time out, the next time a connection is initiated, it is directed to the default peer.

This feature includes the following capabilities:

- Default peer configuration

  If a connection timeout occurs, the connection to the current peer is closed. The **set peer** command allows you to configure the first peer as the default peer. If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If the default peer is unresponsive, the next peer in the peer list becomes the current peer and future connections through the crypto map try that peer.

  This capability is useful when traffic on a physical link stops due to the failure of a remote peer. DPD indicates that the remote peer is unavailable, but that peer remains the current peer.

  A default peer facilitates the failover to a preferred peer that was previously unavailable, but has returned to service. Users can give preference to certain peers in the event of a failover. This is useful if the original failure was due to a network connectivity problem rather than failure of the remote peer.

  To configure a default peer, see the "Configuring a Default Peer" section on page 8-10.

- IPsec idle timer with default peer configuration

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

  IPsec SA idle timers increase the availability of resources by deleting SAs associated with idle peers. Because IPsec SA idle timers prevent the wasting of resources by idle peers, more resources are available to create new SAs when required. (If 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.)

  When both an IPsec SA idle timer and a default peer are configured and all connections to the current peer time out, the next time a connection is initiated it is directed to the default peer configured in the **set peer** command. If a default peer is not configured and there is a connection timeout, the peer that timed out remains the current peer.

  This enhancement helps facilitate a failover to a preferred peer that was previously unavailable but is in service now.

  To configure an IPsec idle timer, see the "Configuring the IPsec Idle Timer with a Default Peer" section on page 8-11.

IPsec Preferred Peer Configuration Guidelines and Restrictions

When configuring an IPsec preferred peer, follow these guidelines and restrictions:

- When configuring a default peer, follow these guidelines and restrictions:
  - Only one peer can be designated as the default peer in a crypto map.

  - The default peer must be the first peer in the peer list.

Note The default peer feature must be used in conjunction with Dead Peer Detection (DPD). It is most effective on a remote site running DPD in periodic mode. DPD detects the failure of a device quickly and resets the peer list so that the default peer is tried for the next attempted connection.

- When configuring IPsec idle timer usage with a default peer, follow these guidelines and restrictions:
  - The IPsec idle timer usage with a default peer feature works only on the crypto map for which it is configured. You cannot configure the capability globally for all crypto maps.
  - If there is a global idle timer, the crypto map idle timer value must be different from the global value; otherwise, the idle timer is not added to the crypto map.

Configuring a Default Peer

To configure a default peer, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num [ipsec-isakmp] [dynamic dynamic-map-name] [discover] [profile profile-name]`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—(Optional) Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry. • **dynamic** *dynamic-map-name*—(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template. • **discover**—(Optional) Enables peer discovery. By default, peer discovery is not enabled. • **profile** *profile-name*—(Optional) Name of the crypto profile being created.
Step 2	`Router(config-crypto-map)# set peer {host-name [dynamic] [default] | ip-address [default]}`	Specifies an IPsec peer in a crypto map entry. Ensures that the first peer specified is defined as the default peer. • *host-name*—Specifies the IPsec peer by its host name. This is the peer's host name concatenated with its domain name (for example, myhost.example.com). • **dynamic**—(Optional) The host name of the IPsec peer will be resolved via a domain name server (DNS) lookup right before the router establishes the IPsec tunnel. • **default**—(Optional) If there are multiple IPsec peers, designates that the first peer is the default peer. • *ip-address*—Specifies the IPsec peer by its IP address.
Step 3	`Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

Configuring the IPsec Idle Timer with a Default Peer

To configure the IPsec idle timer with a default peer, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num [ipsec-isakmp] [dynamic dynamic-map-name] [discover] [profile profile-name]`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—(Optional) Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry. • **dynamic** *dynamic-map-name*—(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template. • **discover**—(Optional) Enables peer discovery. By default, peer discovery is not enabled. • **profile** *profile-name*—(Optional) Name of the crypto profile being created.
Step 2	`Router(config-crypto-map)# set security-association idle-time seconds [default]`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used. • *seconds*—Number of seconds for which the current peer can be idle before the default peer is used. Valid values are 60 to 86400. • **default**—(Optional) Specifies that the next connection is directed to the default peer.
Step 3	`Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

For complete configuration information for IPsec preferred peer, see this URL:

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

For IPsec preferred peer configuration examples, see the "IPsec Preferred Peer Configuration Examples" section on page 8-23.

Configuring IPsec Security Association Idle Timers

When a switch running Cisco IOS software creates an IPsec 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 switch 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 idle timers can be configured either globally, on a per-crypto map basis, or through an ISAKMP profile. The benefits of this feature include the following:

- Increased availability of resources
- Improved scalability of Cisco IOS IPsec deployments

IPsec Security Association Idle Timer Configuration Guidelines

When configuring idle timers on a per-crypto map basis, follow these guidelines:

- When the idle timer is configured globally, the idle timer configuration will be applied to all SAs.
- When the idle timer is configured for a crypto map, the idle timer configuration will be applied to all SAs under the specified crypto map.

Configuring the IPsec SA Idle Timer Globally

To configure the IPsec SA idle timer globally, enter the **crypto ipsec security-association idle-time** command in global configuration mode as follows:

Command	Purpose
`Router(config)# crypto ipsec security-association idle-time` *seconds*	Specifies the time, in *seconds*, that the idle timer will allow an inactive peer to maintain an SA. The range is from 60 to 86400 seconds.

Configuring the IPsec SA Idle Timer per Crypto Map

To configure the IPsec SA idle timer for a specified crypto map, use the **set security-association idle-time** command within a crypto map configuration:

	Command	Purpose
Step 1	`Router(config)# crypto map` *map-name seq-number* `ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations.
Step 2	`Router(config-crypto-map)# set security-association idle-time` *seconds*	Specifies the time, in *seconds*, that the idle timer will allow an inactive peer to maintain an SA. The range is from 60 to 86400 seconds.

For detailed information on configuring IPsec SA idle timers, refer to the following Cisco IOS documentation:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftsaidle.html

For IPsec SA idle timer configuration examples, see the "IPsec Security Association Idle Timer Configuration Examples" section on page 8-24.

Configuring Distinguished Name-Based Crypto Maps

The distinguished name-based crypto maps feature allows you to configure the switch to restrict access to selected encrypted interfaces for those peers with specific certificates, especially certificates with particular distinguished names (DNs).

Previously, if the switch accepted a certificate or a shared secret from the encrypting peer, Cisco IOS did not have a method of preventing the peer from communicating with any encrypted interface other than the restrictions on the IP address of the encrypting peer. This feature allows you to configure which crypto maps are usable to a peer based on the DN that a peer used to authenticate itself, which enables you to control which encrypted interfaces a peer with a specified DN can access. You can configure a DN-based crypto map that can be used only by peers that have been authenticated by a DN or one that can be used only by peers that have been authenticated by a hostname.

Distinguished Name-Based Crypto Map Configuration Guidelines and Restrictions

When configuring a DN-based crypto map, follow these guidelines and restrictions:

- If you restrict access to a large number of DNs, we recommend that you specify a few number of crypto maps referring to large identity sections instead of specifying a large number of crypto maps referring to small identity sections.

Configuring a DN-Based Crypto Map

To configure a DN-based crypto map that can be used only by peers that have been authenticated by a DN, or one that can be used only by peers that have been authenticated by a hostname, perform this task beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# crypto isakmp policy priority`
	`...`
	`Router(config-isakmp)# exit`
		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.
		Creates an ISAKMP policy at each peer.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Step 2	`Router(config)# crypto map map-name seq-number ipsec-isakmp`	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations.

Step 3	`Router(config-crypto-map)# set identity name`
	`...`
	`Router(config-crypto-map)# exit`
		Applies the identity to the crypto map.
		• *name*—Identity of the switch, which is associated with the given list of DNs.
		When this command is applied, only the hosts that match a configuration listed within the identity name can use the specified crypto map.
		Note If the **set identity** command does not appear within the crypto map, the encrypted connection does not have any restrictions other than the IP address of the encrypting peer.
		Specify any other policy values appropriate to your configuration.
		For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 4	`Router(config)# crypto identity name`	Configures the identity of a switch with the given list of DNs in the certificate of the switch and enters crypto identity configuration mode. • *name*—The name value specified in Step 3.
Step 5	`Router(crypto-identity)# dn name=string [,name=string]| fqdn name`	Associates the identity of the switch with either a DN or hostname (FQDN) to restrict access to peers with specific certificates. • *name*=**string**—The DN in the certificate of the switch. Optionally, you can associate more than one DN. • **fqdn** *name*—The hostname that the peer used to authenticate itself (FQDN) or the DN in the certificate of the switch. The identity of the peer must match the identity in the exchanged certificate.

For complete configuration information for distinguished name-based crypto maps, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t4/feature/guide/ftdnacl.html

For a distinguished name-based crypto map configuration example, see the "Distinguished Name-Based Crypto Maps Configuration Example" section on page 8-24.

Configuring Platform ACLs for Tunnel Interfaces

You can apply access control lists (ACLs) to tunnel interfaces.

Platform ACL on Tunnel Interfaces Configuration Guidelines and Restrictions

When configuring platform ACLs for a VSPA, follow these guidelines and note these restrictions:

- ACLs can be applied to the tunnel interface in any of these situations:
  - GRE in crypto-connect mode, whether or not GRE is taken over by the VSPA
  - GRE with tunnel protection, whether or not GRE is taken over by the VSPA
  - Static VTI
  - DMVPN, in either crypto-connect or VRF mode
- Permit and deny ACLs can be applied to GRE tunnel interfaces in either the inbound or outbound direction.
- In crypto-connect mode with GRE, when GRE is not taken over by the VSPA, apply the ACL to the interface VLAN to filter GRE-encapsulated packets, or to the tunnel interface to filter clear IP packets.
- In crypto-connect mode with GRE, when GRE is taken over by the VSPA, ACLs on the interface VLAN are not supported. Apply the ACL to the tunnel interface to filter clear IP packets.
- ACLs on tunnels are supported in blade-to-blade failover.

- ACLs will be applied to transit packets, but will not be applied to packets generated by the switch.

For a platform ACL configuration example, see the "Platform ACL Configuration Example" section on page 8-25.

Configuring Sequenced Crypto ACLs

Access control lists (ACLs) are made up of access control entries (ACEs). With sequenced ACLs, ACEs can be entered with a sequence number in front of the ACE and the ACEs are then processed by sequence number. Additionally, ACEs can be deleted one at a time by using the sequence number in the front of the ACE that you want to delete. The sequence numbers do not appear in the configuration but they can be displayed using the **show access-list** command.

Note If an ACE is removed or modified, the ACL is reconfigured on the module, which might result in the closing of existing sessions.

Configuring Deny Policy Enhancements for Crypto ACLs

Use the **crypto ipsec ipv4-deny** {**jump** | **clear** | **drop**} command set as follows:

- The **jump** keyword is not supported.
- The **clear** keyword allows a deny address range to be programmed in hardware. The deny addresses are then filtered out for encryption and decryption. If the VPN mode is crypto-connect, when a deny address is hit, the search is stopped and traffic is allowed to pass in the clear (unencrypted) state. If the VPN mode is VRF, the deny address matching traffic is dropped.
- The **drop** keyword causes traffic to be dropped when a deny address is hit.

The **clear** and **drop** keywords can be used to prevent repeated address ranges from being programmed in the hardware, resulting in more efficient TCAM space utilization.

Deny Policy Enhancements for ACLs Configuration Guidelines and Restrictions

When configuring the deny policy enhancements, follow these guidelines and restrictions:

- The **crypto ipsec ipv4-deny** {**jump** | **clear** | **drop**} command is a global command that is applied to a single VSPA. The specified keyword (**jump**, **clear**, or **drop**) is propagated to the ACE software of the VSPA. The default behavior is **clear**.
- When the **clear** keyword is used with VRF mode, deny address traffic is dropped rather than passed in the clear state. VRF mode does not pass traffic in the clear state.
- If you apply the specified keyword (**jump**, **clear**, or **drop**) when crypto maps are already configured on the VSPA, all existing IPsec sessions are temporarily removed and restarted, which impacts traffic on your network.
- The number of deny entries that can be specified in an ACL are dependent on the keyword specified:
  - **jump**—Not supported.
  - **clear**—Supports up to 1000 deny entries in an ACL.
  - **drop**—Supports up to 1000 deny entries in an ACL.

For a deny policy enhancements configuration example, see the "Deny Policy Enhancements for ACLs Configuration Example" section on page 8-26.

Understanding IPv6 IPsec Support in the VSPA

The VSPA supports IPv6 encryption and decryption for configurations using a static virtual tunnel interface (sVTI) in VRF mode where all tunnel interfaces are terminated in the global context.

This section contains the following topics:

IPv6 IPsec Configuration Guidelines and Restrictions

When configuring IPsec for IPv6, follow these guidelines and note these restrictions:

- A tunnel must be either IPv4 or IPv6. If you assign both an IPv4 address and an IPv6 address to a tunnel, packets for the tunnel will be dropped.
- We recommend that you explicitly configure the tunnel source with an IPv6 address instead of an interface name to avoid ambiguity, because the tunnel source address will be used as your IKE local endpoint address. IKE supports only preshared key authentication, using the endpoint address as an ID.
- The VSPA supports a maximum IPv6 MTU of 9216. Packets that exceed the path MTU of the tunnel will be dropped and the VSPA will send a PMTU Packet Too Big message to the source host. IPv6 packets are fragmented only by the originating host.
- The number of IPv6 sVTI tunnels supported by the VSPA depends on the routing protocol used, as shown in the following table:

IPv6 Routing Protocol	Tunnels
Static	1000
BGP	500
EIGRP	500
OSPFv3	200
RIPng	200

- The following features are not supported by the VSPA for IPv6:
  - Crypto connect mode
  - IP VRF forwarding on IPv6 tunnel interfaces
  - OSPFv3 with authentication (OSPFv3 through the tunnel is supported)
  - Intermediate System-to-Intermediate System (IS-IS) routing protocol
  - Multicast
  - IPv4-in-IPv6, IPv6-in-IPv4
  - AH encapsulation

- Path MTU discovery (PMTUD)
- Dead peer detection (DPD, PDPD)
- Extension headers on encrypted packets (extension headers on cleartext packets are supported)
- Crypto certificates
- MPLS
- HSRPv6
- Unless explicitly stated, platform features (such as QoS and ACL) applied to the tunnel interface are not supported

Tunnel Source Address Selection

IPv6 allows an interface to have multiple aggregatable global unicode (AGU) addresses and a link-local address. If a single global unicast address is configured on the tunnel interface, then that address will be used as the tunnel source address. If multiple global unicast addresses are configured, then the first address will be used. If no global unicast address is configured on the tunnel interface, then the link-local address will be used.

Configuring IPv6 IPsec

To configure IPsec for IPv6, perform this task:

	Command	Purpose
Step 1	`Router(config)# crypto engine mode vrf`	Enables VRF mode for the VSPA.
Step 2	`Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast packets.
Step 3	`Router(config)# crypto isakmp key` *enc-type-digit keystring* `address ipv6` *rem_ipv6_address/prefix*	Configures a preshared authentication key and specifies the IPv6 address of the remote peer.
Step 4	`Router(config)# crypto ipsec transform-set` *transform-set-name* `esp-aes esp-sha`	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms.
Step 5	`Router(config-crypto-trans)# exit`	Exits crypto transform configuration mode.
Step 6	`Router(config)# crypto ipsec profile` *profile-name*	Defines an IPsec profile and enters IPsec profile configuration mode. • *profile-name*—Name of the user profile.
Step 7	`Router(config-ipsec-profile)# set transform-set` *transform-set-name*	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set. Enter the value specified in Step 4.
Step 8	`Router(config-ipsec-profile)# exit`	Exits IPsec profile configuration mode.

	Command	Purpose
Step 9	`Router(config)# interface tunnel-number`	Configures a tunnel interface and enters interface configuration mode. • *tunnel-number*—Name assigned to the tunnel interface.
Step 10	`Router(config-if)# ipv6 address ipv6_address`	Specifies an IPv6 address and enables IPv6 processing on the interface.
Step 11	`Router(config-if)# tunnel source src_ipv6_address`	Specifies an IPv6 address as the source for the tunnel interface.
Step 12	`Router(config-if)# tunnel destination dst_ipv6_address`	Specifies an IPv6 address as the destination for the tunnel interface.
Step 13	`Router(config-if)# tunnel mode ipsec ipv6`	Tunnel mode is IPsec and the transport is IPv6.
Step 14	`Router(config-if)# tunnel protection ipsec profile profile-name`	Associates a tunnel interface with an IPsec profile. • *profile-name*—Name of the crypto profile.
Step 15	`Router(config-if)# crypto engine slot slot/subslot inside`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot and subslot where the VSPA is located.
Step 16	`Router(config-if)# exit`	Exits interface configuration mode.
Step 17	`Router(config)# interface gigabitethernet slot/subslot/port`	Configures the physical egress interface.
Step 18	`Router(config-if)# ipv6 address ipv6_address`	Specifies an IPv6 address and enables IPv6 processing on the interface.
Step 19	`Router(config-if)# crypto engine outside`	Assigns the crypto engine to the interface.
Step 20	`Router(config-if)# exit`	Exits interface configuration mode.

For IPv6 IPsec configuration examples, see the "IPv6 IPsec Configuration Example" section on page 8-26.

Configuration Examples

This section provides examples of the following configurations:

Advanced Encryption Standard Configuration Example

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

```
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
```

Reverse Route Injection Configuration Examples

The following examples show how to configure RRI:

RRI Under a Static Crypto Map Configuration Example

The following example shows how to configure RRI under a static crypto map. In this example, the RRI-created route has been tagged with a tag number. This tag number can then be used by a routing process to redistribute the tagged route via a route map:

```
Router(config)# crypto map mymap 1 ipsec-isakmp
Router(config-crypto-map)# reverse-route tag 5
```

RRI Under a Dynamic Crypto Map Configuration Example

The following example shows how to configure RRI under a dynamic crypto map:

```
Router(config)# crypto dynamic-map mymap 1
Router(config-crypto-map)# reverse-route remote peer 10.1.1.1
```

RRI with Existing ACLs Configuration Example

The following example shows how to configure RRI for a situation in which there are existing ACLs:

```
Router(config)# crypto map mymap 1 ipsec-isakmp
Router(config-crypto-map)# set peer 172.17.11.1
Router(config-crypto-map)# reverse-route static
Router(config-crypto-map)# set transform-set esp-3des-sha
Router(config-crypto-map)# match address 101

access-list 101 permit ip 192.168.1.0 0.0.0.255 172.17.11.0 0.0.0.255
```

RRI for Two Routes Configuration Example

The following example shows how to configure two routes, one for the remote endpoint and one for route recursion to the remote endpoint via the interface on which the crypto map is configured:

```
Router(config-crypto-map)# reverse-route remote-peer
```

RRI via a User-Defined Hop Configuration Example

The following example shows that one route has been created to the remote proxy via a user-defined next hop. This next hop should not require a recursive route lookup unless it will recurse to a default route.

```
Router(config-crypto-map)# reverse-route remote-peer 10.4.4.4
```

IPsec Anti-Replay Window Size Configuration Examples

The following examples show how to configure the IPsec anti-replay window size:

- IPsec Anti-Replay Window Global Configuration Example, page 8-21
- IPsec Anti-Replay Window per Crypto Map Configuration Example, page 8-22

IPsec Anti-Replay Window Global Configuration Example

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

```
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 Ethernet0/0
```

```
 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
```

IPsec Anti-Replay Window per Crypto Map Configuration Example

The following example shows that anti-replay checking is disabled for IPsec connections to 172.150.150.2, but enabled (and the default window size is 64) for IPsec connections to 172.150.150.3 and 172.150.150.4:

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname dr_whoovie
!
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 172.150.150.2
 crypto isakmp key cisco180
 address 172.150.150.3
 crypto isakmp key cisco190
 address 172.150.150.4
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 172.150.150.2
 set security-association replay disable
 set transform-set 170cisco
 match address 170
crypto map ETH0 18 ipsec-isakmp
 set peer 150.150.150.3
 set transform-set 180cisco
 match address 180
crypto map ETH0 19 ipsec-isakmp
 set peer 150.150.150.4
```

```
 set transform-set 190cisco
 match address 190
!
interface Ethernet0
 ip address 172.150.150.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no mop enabled
 crypto map ETH0
!
interface Serial0
 ip address 172.160.160.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
ip classless
ip route 172.170.170.0 255.255.255.0 172.150.150.2
ip route 172.180.180.0 255.255.255.0 172.150.150.3
ip route 172.190.190.0 255.255.255.0 172.150.150.4
no ip http server
!
access-list 170 permit ip 172.160.160.0 0.0.0.255 172.170.170.0 0.0.0.255
access-list 180 permit ip 172.160.160.0 0.0.0.255 172.180.180.0 0.0.0.255
access-list 190 permit ip 172.160.160.0 0.0.0.255 172.190.190.0 0.0.0.255
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
end
```

IPsec Preferred Peer Configuration Examples

The following examples show how to configure an IPsec preferred peer:

- Default Peer Configuration Example, page 8-23
- IPsec Idle Timer with Default Peer Configuration Example, page 8-24

Default Peer Configuration Example

The following example shows how to configure a default peer. In this example, the first peer, at IP address 1.1.1.1, is the default peer:

```
Router(config)# crypto map tohub 1 ipsec-isakmp
Router(config-crypto-map)# set peer 1.1.1.1 default
Router(config-crypto-map)# set peer 2.2.2.2
Router(config-crypto-map)# exit
```

IPsec Idle Timer with Default Peer Configuration Example

The following example shows how to configure an IPsec idle timer with a default peer. In the following example, if the current peer is idle for 600 seconds, the default peer 1.1.1.1 (which was specified in the **set peer** command) is used for the next attempted connection:

```
Router (config)# crypto map tohub 1 ipsec-isakmp
Router(config-crypto-map)# set peer 1.1.1.1 default
Router(config-crypto-map)# set peer 2.2.2.2
Router(config-crypto-map)# set security-association idle-time 600 default
Router(config-crypto-map)# exit
```

IPsec Security Association Idle Timer Configuration Examples

The following examples show how to configure the IPsec SA idle timer:

- IPsec SA Idle Timer Global Configuration Example, page 8-24
- IPsec SA Idle Timer per Crypto Map Configuration Example, page 8-24

IPsec SA Idle Timer Global Configuration Example

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

```
Router(config)# crypto ipsec security-association idle-time 600
```

IPsec SA Idle Timer per Crypto Map Configuration 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:

```
Router(config) # crypto map test 1 ipsec-isakmp
Router(config-crypto-map)# set security-association idle-time 600
```

Distinguished Name-Based Crypto Maps Configuration Example

The following example shows how to configure DN-based crypto maps that have been authenticated by DN and hostname. Comments are included inline to explain various commands.

```
! DN based crypto maps require you to configure an IKE policy at each peer.
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
 crypto isakmp key 1234567890 address 171.69.224.33
!
!The following is an IPsec crypto map (part of IPsec configuration). It can be used only
! by peers that have been authenticated by DN and if the certificate belongs to BigBiz.
crypto map map-to-bigbiz 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
```

```
 match address 124
 identity to-bigbiz
!
crypto identity to-bigbiz
dn ou=BigBiz
!
!
! This crypto map can be used only by peers that have been authenticated by hostname
!and if the certificate belongs to little.com.
crypto map map-to-little-com 10 ipsec-isakmp
 set peer 172.21.115.119
 set transform-set my-transformset
 match address 125
 identity to-little-com
!
crypto identity to-little-com
fqdn little.com
!
```

Platform ACL Configuration Example

This example shows a tunnel configuration with inbound and outbound ACLs:

```
interface Tunnel1
 ip vrf forwarding i1
 ip address 26.0.1.2 255.255.255.0
 ip access-group T1ACL_IN in
 ip access-group T1ACL_OUT out
 ip mtu 1400
 tunnel source 27.0.1.2
 tunnel destination 67.0.1.6
 tunnel vrf f1
 tunnel protection ipsec profile TUN_PROTECTION
 crypto engine slot 3/0 inside
!
!
ip access-list extended T1ACL_IN
 permit tcp any any
 permit icmp any any
 permit ip any host 50.0.1.2 precedence critical
 permit ip any host 50.0.1.2 precedence internet
 permit ip any host 50.0.1.2 precedence priority
 permit ip any host 50.0.1.2 precedence flash
 deny   ip any any
ip access-list extended T1ACL_OUT
 permit tcp any any
 permit icmp any any
 permit ip any host 60.0.1.2 precedence critical
 permit ip any host 60.0.1.2 precedence internet
 permit ip any host 60.0.1.2 precedence priority
 permit ip any host 60.0.1.2 precedence flash
 deny   ip any any
```

Deny Policy Enhancements for ACLs Configuration Example

The following example shows a configuration using the deny policy **clear** option. In this example, when a deny address is hit, the search will stop and traffic will be allowed to pass in the clear (unencrypted) state:

```
Router(config)# crypto ipsec ipv4-deny clear
```

IPv6 IPsec Configuration Example

This example shows how to configure IPv6 IPsec between a hub and a spoke:

Hub Configuration

```
ipv6 unicast-routing
crypto engine mode vrf
crypto isakmp policy 1
  encr aes 192
  group 2
  auth pre-share
  lifetime 7200
crypto isakmp key 12345 address ipv6 ::/0
crypto ipsec transform-set ts esp-aes 256 esp-sha-hmac
!
! WAN interface
!
interface TenGigabitEthernet3/2
  mtu 9216
  ipv6 enable
  ipv6 address 2001:410:4::1/48
  crypto engine outside
  no shut
!
! LAN interface
!
interface TenGigabitEthernet3/1
  mtu 9216
  ipv6 enable
  ipv6 address 2001:410:5::2/48
  no shut
crypto ipsec profile tp
  set transform-set ts
interface Loopback1
  ipv6 address 2001:410:1:1::1/128
  no shut
interface Tunnel1
  ipv6 enable
  ipv6 address 2001:410:2:1::1/64
  tunnel source Loopback1
  tunnel destination 2001:410:1:2::1
  tunnel mode ipsec ipv6
  tunnel protection ipsec profile tp
  crypto engine slot 5/0 inside
  no shut
!
ipv6 route 2001:410:1:2::/64 2001:410:4::2
ipv6 route 2001:410:8:1::1/128 Tunnel1
!
```

Spoke Configuration

```
ipv6 unicast-routing
crypto engine mode vrf
crypto isakmp policy 1
  encr aes 192
  group 2
  auth pre-share
  lifetime 7200
crypto isakmp key 12345 address ipv6 ::/0
crypto ipsec transform-set ts esp-aes 256 esp-sha-hmac
!
! WAN interface
!
interface TenGigabitEthernet2/3
  mtu 9216
  ipv6 enable
  ipv6 address 2001:410:4::2/48
  crypto engine outside
  no shut
!
! LAN interface
!
interface TenGigabitEthernet2/1
  mtu 9216
  ipv6 enable
  ipv6 address 2001:410:6::1/48
  no shut
crypto ipsec profile tp
  set transform-set ts
interface Loopback1
  ipv6 address 2001:410:1:2::1/128
  no shut
interface Tunnel1
  ipv6 enable
  ipv6 address 2001:410:2:1::2/64
  tunnel source Loopback1
  tunnel destination 2001:410:1:1::1
  tunnel mode ipsec ipv6
  tunnel protection ipsec profile tp
  crypto engine slot 4/0 inside
  no shut
!
ipv6 route 2001:410:1:1::/64 2001:410:4::1
ipv6 route 2001:410:7:1::1/128 Tunnel1
!
```