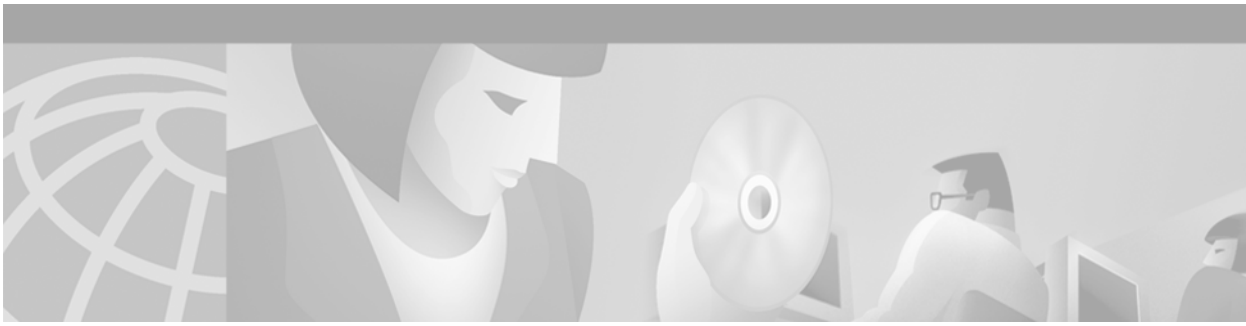

IPSec Network Security Commands

This chapter describes IP Security (IPSec) network security commands. IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec provides a robust security solution and is standards-based. IPSec provides data authentication and anti-replay services in addition to data confidentiality services.

For configuration information, refer to the chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide.*

clear crypto sa

To delete IP Security security associations, use the **clear crypto sa** EXEC command.

clear crypto sa

clear crypto sa peer {*ip-address* | *peer-name*}

clear crypto sa map *map-name*

clear crypto sa entry *destination-address protocol spi*

clear crypto sa counters

Syntax Description

peer	Deletes any IPSec security associations for the specified peer.
ip-address	Specifies a remote peer's IP address.
peer-name	Specifies a remote peer's name as the fully qualified domain name, for example remotepeer.example.com.
map	Deletes any IPSec security associations for the named crypto map set.
map-name	Specifies the name of a crypto map set.
entry	Deletes the IPSec security association with the specified address, protocol, and SPI.
destination-address	Specifies the IP address of your peer or the remote peer.
protocol	Specifies either the Encapsulation Security Protocol or Authentication Header.
spi	Specifies an SPI (found by displaying the security association database).
counters	Clears the traffic counters maintained for each security association; **counters** does not clear the security associations themselves.

Defaults

If the **peer**, **map**, **entry**, or **counters** keyword is not used, all IPSec security associations are deleted.

Command Modes

EXEC

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command clears (deletes) IPSec security associations.

If the security associations were established via Internet Key Exchange, they are deleted and future IPSec traffic will require new security associations to be negotiated. (When IKE is used, the IPSec security associations are established only when needed.)

If the security associations are manually established, the security associations are deleted and reinstalled. (When IKE is not used, the IPSec security associations are created as soon as the configuration is completed.)

If **peer**, **map**, **entry**, or **counters** keywords are not used, all IPSec security associations will be deleted.

- The **peer** keyword deletes any IPSec security associations for the specified peer.
- The **map** keyword deletes any IPSec security associations for the named crypto map set.
- The **entry** keyword deletes the IPSec security association with the specified address, protocol, and SPI.

If any of the above commands cause a particular security association to be deleted, all the "sibling" security associations—that were established during the same IKE negotiation—are deleted as well.

The **counters** keyword simply clears the traffic counters maintained for each security association; it does not clear the security associations themselves.

If you make configuration changes that affect security associations, these changes will not apply to existing security associations but to negotiations for subsequent security associations. You can use the **clear crypto sa** command to restart all security associations so they will use the most current configuration settings. In the case of manually established security associations, if you make changes that affect security associations you must use the **clear crypto sa** command before the changes take effect.

If the router is processing active IPSec traffic, it is suggested that you only clear the portion of the security association database that is affected by the changes, to avoid causing active IPSec traffic to temporarily fail.

Note that this command only clears IPSec security associations; to clear IKE state, use the **clear crypto isakmp** command.

Examples

The following example clears (and reinitializes if appropriate) all IPSec security associations at the router:

```
clear crypto sa
```

The following example clears (and reinitializes if appropriate) the inbound and outbound IPSec security associations established along with the security association established for address 10.0.0.1 using the AH protocol with the SPI of 256:

```
clear crypto sa entry 10.0.0.1 AH 256
```

Related Commands

Command	Description
clear crypto isakmp	Clears active IKE connections.

crypto dynamic-map

To create a dynamic crypto map entry and enter the crypto map configuration command mode, use the **crypto dynamic-map** global configuration command. To delete a dynamic crypto map set or entry, use the **no** form of this command.

crypto dynamic-map *dynamic-map-name dynamic-seq-num*

no crypto dynamic-map *dynamic-map-name* [*dynamic-seq-num*]

Syntax Description

dynamic-map-name	Specifies the name of the dynamic crypto map set.
dynamic-seq-num	Specifies the number of the dynamic crypto map entry.

Defaults

No dynamic crypto maps exist.

Command Modes

Global configuration. Using this command puts you into crypto map configuration mode.

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

Use dynamic crypto maps to create policy templates that can be used when processing negotiation requests for new security associations from a remote IP Security peer, even if you do not know all of the crypto map parameters required to communicate with the remote peer (such as the peer's IP address). For example, if you do not know about all the IPSec remote peers in your network, a dynamic crypto map allows you to accept requests for new security associations from previously unknown peers. (However, these requests are not processed until the Internet Key Exchange authentication has completed successfully.)

When a router receives a negotiation request via IKE from another IPSec peer, the request is examined to see if it matches a crypto map entry. If the negotiation does not match any explicit crypto map entry, it will be rejected unless the crypto map set includes a reference to a dynamic crypto map.

The dynamic crypto map is a policy template; it will accept "wildcard" parameters for any parameters not explicitly stated in the dynamic crypto map entry. This allows you to set up IPSec security associations with a previously unknown IPSec peer. (The peer still must specify matching values for the "non-wildcard" IPSec security association negotiation parameters.)

If the router accepts the peer's request, at the point that it installs the new IPSec security associations it also installs a temporary crypto map entry. This entry is filled in with the results of the negotiation. At this point, the router performs normal processing, using this temporary crypto map entry as a normal entry, even requesting new security associations if the current ones are expiring (based upon the policy specified in the temporary crypto map entry). Once the flow expires (that is, all of the corresponding security associations expire), the temporary crypto map entry is removed.

Dynamic crypto map sets are not used for initiating IPSec security associations. However, they are used for determining whether or not traffic should be protected.

The only configuration required in a dynamic crypto map is the **set transform-set** command. All other configuration is optional.

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. After you define a dynamic crypto map set (which commonly contains only one map entry) using this command, you include the dynamic crypto map set in an entry of the "parent" crypto map set using the **crypto map** (IPSec global configuration) command. The parent crypto map set is then applied to an interface.

You should make crypto map entries referencing dynamic maps the lowest priority map entries, so that negotiations for security associations will try to match the static crypto map entries first. Only after the negotiation request does not match any of the static map entries do you want it to be evaluated against the dynamic map.

To make a dynamic crypto map the lowest priority map entry, give the map entry referencing the dynamic crypto map the highest *seq-num* of all the map entries in a crypto map set.

For both static and dynamic crypto maps, if unprotected inbound traffic matches a **permit** statement in an access list, and the corresponding crypto map entry is tagged as "IPSec," then the traffic is dropped because it is not IPSec-protected. (This is because the security policy as specified by the crypto map entry states that this traffic must be IPSec-protected.)

For static crypto map entries, if outbound traffic matches a **permit** statement in an access list and the corresponding security association (SA) is not yet established, the router will initiate new SAs with the remote peer. In the case of dynamic crypto map entries, if no SA existed, the traffic would simply be dropped (because dynamic crypto maps are not used for initiating new SAs).

Note Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** entry to include multicast or broadcast traffic, the access list should include **deny** entries for the appropriate address range. Access lists should also include **deny** entries for network and subnet broadcast traffic, and for any other traffic that should not be IPSec protected.

Examples

The following example configures an IPSec crypto map set.

Crypto map entry "mymap 30" references the dynamic crypto map set "mydynamicmap," which can be used to process inbound security association negotiation requests that do not match "mymap" entries 10 or 20. In this case, if the peer specifies a transform set that matches one of the transform sets specified in "mydynamicmap," for a flow "permitted" by the access list 103, IPSec will accept the request and set up security associations with the remote peer without previously knowing about the remote peer. If accepted, the resulting security associations (and temporary crypto map entry) are established according to the settings specified by the remote peer.

The access list associated with "mydynamicmap 10" is also used as a filter. Inbound packets that match a **permit** statement in this list are dropped for not being IPSec protected. (The same is true for access lists associated with static crypto maps entries.) Outbound packets that match a **permit** statement without an existing corresponding IPSec SA are also dropped.

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
 set peer 10.0.0.2
crypto map mymap 20 ipsec-isakmp
 match address 102
 set transform-set my_t_set1 my_t_set2
 set peer 10.0.0.3
```

```
crypto map mymap 30 ipsec-isakmp dynamic mydynamicmap
!
crypto dynamic-map mydynamicmap 10
 match address 103
 set transform-set my_t_set1 my_t_set2 my_t_set3
```

Related Commands

Command	Description
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.
set peer (IPSec)	Specifies an IPSec peer in a crypto map entry.
set pfs	Specifies that IPSec should ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations.
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto dynamic-map	Displays a dynamic crypto map set.
show crypto map (IPSec)	Displays the crypto map configuration.

crypto engine accelerator

To enable the IP Security (IPSec) accelerator, use the **crypto engine accelerator** command in global configuration mode. To disable the IPSec accelerator and perform IPSec encryption and decryption in the Cisco IOS software, use the **no** form of this command.

crypto engine accelerator [*slot*]

no crypto engine accelerator [*slot*]

Syntax Description

slot	(Optional) The slot number on the crypto engine.

Defaults

The hardware accelerator for IPSec encryption is enabled.

Command Modes

Global configuration

Command History

Release	Modification
12.1(3)T	This command was introduced on the Cisco 1700 series and any other Cisco router that supports hardware accelerators for IPSec encryption.
12.1(3)XL	This command was implemented on the Cisco uBR905 cable access router.

Usage Guidelines

This command is normally not needed for typical operations because the hardware accelerator for IPSec encryption is enabled by default.

Note The hardware accelerator *should not* be disabled except on instruction from Cisco TAC personnel.

Examples

The following example disables the onboard hardware accelerator of the router. If IPSec encryption is configured, all current connections are brought down. Future encryption will be performed by the Cisco IOS software, which has the same functionality as the hardware accelerator, but performance is significantly slower.

```
Router(config)# no crypto engine accelerator

 Warning! all current connections will be torn down.
 Do you want to continue? [yes/no]: y
 ...Crypto accelerator in slot 0 disabled
 ...switching to SW IPsec crypto engine
```

Related Commands

Command	Description
show crypto engine accelerator sa-database	Displays active (in-use) entries in the platform-specific VPN module database.

crypto ipsec security-association lifetime

To change global lifetime values used when negotiating IPSec security associations, use the **crypto ipsec security-association lifetime** global configuration command. To reset a lifetime to the default value, use the **no** form of this command.

crypto ipsec security-association lifetime {**seconds** *seconds* | **kilobytes** *kilobytes*}

no crypto ipsec security-association lifetime {**seconds** | **kilobytes**}

Syntax Description

seconds *seconds*	Specifies the number of seconds a security association will live before expiring. The default is 3600 seconds (one hour).
kilobytes *kilobytes*	Specifies the volume of traffic (in kilobytes) that can pass between IPSec peers using a given security association before that security association expires. The default is 4,608,000 kilobytes.

Defaults

3600 seconds (one hour) and 4,608,000 kilobytes (10 megabits per second for one hour).

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

IPSec security associations use shared secret keys. These keys and their security associations time out together.

Assuming that the particular crypto map entry does not have lifetime values configured, when the router requests new security associations during security association negotiation, it will specify its global lifetime value in the request to the peer; it will use this value as the lifetime of the new security associations. When the router receives a negotiation request from the peer, it will use the smaller of the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

There are two lifetimes: a “timed” lifetime and a “traffic-volume” lifetime. The security association expires after the first of these lifetimes is reached.

If you change a global lifetime, the change is only applied when the crypto map entry does not have a lifetime value specified. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command. Refer to the **clear crypto sa** command for more details.

To change the global timed lifetime, use the **crypto ipsec security-association lifetime seconds** form of the command. The timed lifetime causes the security association to time out after the specified number of seconds have passed.

To change the global traffic-volume lifetime, use the **crypto ipsec security-association lifetime kilobytes** form of the command. The traffic-volume lifetime causes the security association to time out after the specified amount of traffic (in kilobytes) has been protected by the security associations' key.

Shorter lifetimes can make it harder to mount a successful key recovery attack, since the attacker has less data encrypted under the same key to work with. However, shorter lifetimes require more CPU processing time for establishing new security associations.

The lifetime values are ignored for manually established security associations (security associations installed using an **ipsec-manual** crypto map entry).

How These Lifetimes Work

The security association (and corresponding keys) will expire according to whichever occurs sooner, either after the number of seconds has passed (specified by the **seconds** keyword) or after the amount of traffic in kilobytes has passed (specified by the **kilobytes** keyword).

A new security association is negotiated *before* the lifetime threshold of the existing security association is reached, to ensure that a new security association is ready for use when the old one expires. The new security association is negotiated either 30 seconds before the **seconds** lifetime expires or when the volume of traffic through the tunnel reaches 256 kilobytes less than the **kilobytes** lifetime (whichever occurs first).

If no traffic has passed through the tunnel during the entire life of the security association, a new security association is not negotiated when the lifetime expires. Instead, a new security association will be negotiated only when IPSec sees another packet that should be protected.

Examples

The following example shortens both lifetimes, because the administrator feels there is a higher risk that the keys could be compromised. The timed lifetime is shortened to 2,700 seconds (45 minutes), and the traffic-volume lifetime is shortened to 2,304,000 kilobytes (10 megabits per second for one half hour).

```
crypto ipsec security-association lifetime seconds 2700
crypto ipsec security-association lifetime kilobytes 2304000
```

Related Commands

Command	Description
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations.
show crypto ipsec security-association lifetime	Displays the security-association lifetime value configured for a particular crypto map entry.

crypto ipsec transform-set

To define a transform set—an acceptable combination of security protocols and algorithms—use the **crypto ipsec transform-set** global configuration command. To delete a transform set, use the **no** form of the command.

crypto ipsec transform-set *transform-set-name transform1* [*transform2* [*transform3*]]

no crypto ipsec transform-set *transform-set-name*

Syntax Description

transform-set-name	Specifies the name of the transform set to create (or modify).
transform1 *transform2* *transform3*	Specifies up to three "transforms." These transforms define the IPSec security protocols and algorithms. Accepted transform values are described in the "Usage Guidelines" section.

Defaults

No default behavior or values.

Command Modes

Global configuration. This command invokes the crypto transform configuration mode.

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

A transform set is an acceptable combination of security protocols, algorithms and other settings to apply to IP Security protected traffic. During the IPSec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

You can configure multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry is used in the IPSec security association negotiation to protect the data flows specified by that crypto map entry's access list. During the negotiation, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of both peer's IPSec security associations.

When IKE is not used to establish security associations, a single transform set must be used. The transform set is not negotiated.

Before a transform set can be included in a crypto map entry it must be defined using this command.

A transform set specifies one or two IPSec security protocols (either Encapsulation Security Protocol or Authentication Header or both) and specifies which algorithms to use with the selected security protocol. The ESP and AH IPSec security protocols are described in the section "IPSec Protocols: Encapsulation Security Protocol and Authentication Header."

To define a transform set, you specify one to three "transforms"—each transform represents an IPSec security protocol (ESP or AH) plus the algorithm you want to use. When the particular transform set is used during negotiations for IPSec security associations, the entire transform set (the combination of protocols, algorithms, and other settings) must match a transform set at the remote peer.

In a transform set you could specify the AH protocol, the ESP protocol, or both. If you specify an ESP protocol in a transform set, you can specify just an ESP encryption transform or both an ESP encryption transform and an ESP authentication transform.

Table 23 lists the acceptable transform combination selections for the AH and ESP protocols.

Table 23 Allowed Transform Combinations

Transform type	Transform	Description
AH Transform *(Pick up to one.)*	**ah-md5-hmac**	AH with the MD5 (HMAC variant) authentication algorithm
	a**h-sha-hmac**	AH with the SHA (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm
ESP Encryption Transform *(Pick up to one.)*	e**sp-des**	ESP with the 56-bit DES encryption algorithm
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES)
	esp-null	Null encryption algorithm
ESP Authentication Transform *(Pick up to one.)*	**esp-md5-hmac**	ESP with the MD5 (HMAC variant) authentication algorithm
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP Compression Transform *(Pick up to one.)*	**comp-lzs**	IP compression with the LZS algorithm.

Examples of acceptable transform combinations are:

- **ah-md5-hmac**
- **esp-des**
- **esp-3des** and **esp-md5-hmac**
- **ah-sha-hmac** and **esp-des** and **esp-sha-hmac**
- **comp-lzs**

The parser will prevent you from entering invalid combinations; for example, once you specify an AH transform it will not allow you to specify another AH transform for the current transform set.

IPSec Protocols: Encapsulation Security Protocol and Authentication Header

Both the Encapsulation Security Protocol (ESP) and Authentication Header (AH) protocols implement security services for IPSec.

ESP provides packet encryption and optional data authentication and anti-replay services.

AH provides data authentication and anti-replay services.

ESP encapsulates the protected data—either a full IP datagram (or only the payload)—with an ESP header and an ESP trailer. AH is embedded in the protected data; it inserts an AH header immediately after the outer IP header and before the inner IP datagram or payload. Traffic that originates and terminates at the IPSec peers can be sent in either tunnel or transport mode; all other traffic is sent in tunnel mode. Tunnel mode encapsulates and protects a full IP datagram, while transport mode encapsulates/protects the payload of an IP datagram. For more information about modes, see the **mode (IPSec)** command description.

Selecting Appropriate Transforms

The following tips may help you select transforms that are appropriate for your situation:

- If you want to provide data confidentiality, include an ESP encryption transform.
- If you want to ensure data authentication for the outer IP header as well as the data, include an AH transform. (Some consider the benefits of outer IP header data integrity to be debatable.)
- If you use an ESP encryption transform, also consider including an ESP authentication transform or an AH transform to provide authentication services for the transform set.
- If you want data authentication (either using ESP or AH) you can choose from the MD5 or SHA (HMAC keyed hash variants) authentication algorithms. The SHA algorithm is generally considered stronger than MD5, but is slower.
- Note that some transforms might not be supported by the IPSec peer.
- In cases where you need to specify an encryption transform but do not actually encrypt packets, you can use the **esp-null** transform.

Suggested transform combinations:

- **esp-des** and **esp-sha-hmac**
- **ah-sha-hmac** and **esp-des** and **esp-sha-hmac**

The Crypto Transform Configuration Mode

After you issue the **crypto ipsec transform-set** command, you are put into the crypto transform configuration mode. While in this mode, you can change the mode to tunnel or transport. (These are optional changes.) After you have made these changes, type **exit** to return to global configuration mode. For more information about these optional changes, see the **match address (IPSec)** and mode (IPSec) command descriptions.

Changing Existing Transforms

If one or more transforms are specified in the **crypto ipsec transform-set** command for an existing transform set, the specified transforms will replace the existing transforms for that transform set.

If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command.

Examples

The following example defines two transform sets. The first transform set will be used with an IPSec peer that supports the newer ESP and AH protocols. The second transform set will be used with an IPSec peer that only supports the older transforms.

```
crypto ipsec transform-set newer esp-3des esp-sha-hmac
crypto ipsec transform-set older ah-rfc-1828 esp-rfc1829
```

Related Commands

Command	Description
mode (IPSec)	Changes the mode for a transform set.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto ipsec transform-set	Displays the configured transform sets.

crypto map (global IPSec)

To create or modify a crypto map entry and enter the crypto map configuration mode, use the **crypto map** global configuration command. To delete a crypto map entry or set, use the **no** form of this command.

crypto map *map-name seq-num* **ipsec-manual**

crypto map *map-name seq-num* **ipsec-isakmp** [**dynamic** *dynamic-map-name*] [**discover**]

no crypto map *map-name* [*seq-num*]

Note Issue the **crypto map** *map-name seq-num* command without a keyword to modify an existing crypto map entry.

Syntax Description

map-name	The name that identifies the crypto map set. This is the name assigned when the crypto map was created.
seq-num	The number you assign to the crypto map entry. See additional explanation for using this argument in the "Usage Guidelines" section.
ipsec-manual	Indicates that Internet Key Exchange will not be used to establish the IP Security security associations for protecting the traffic specified by this crypto map entry.
ipsec-isakmp	Indicates that IKE will be used to establish the IPSec security associations for protecting the traffic specified by this crypto map entry.
dynamic	(Optional) Specifies that this crypto map entry is to reference a preexisting dynamic crypto map. Dynamic crypto maps are policy templates used in processing negotiation requests from a peer IPSec device. If you use this keyword, none of the crypto map configuration commands will be available.
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.

Defaults

No crypto maps exist.

Peer discovery is not enabled.

Command Modes

Global configuration. Using this command puts you into crypto map configuration mode, unless you use the **dynamic** keyword.

Command History

Release	Modification
11.2	This command was introduced.
11.3 T	The following keywords and arguments were added: • **ipsec-manual** • **ipsec-isakmp** • **dynamic** • *dynamic-map-name*
12.0(5)T	The **discover** keyword was added to support Tunnel Endpoint Discovery (TED).

Usage Guidelines

Use this command to create a new crypto map entry or to modify an existing crypto map entry.

Once a crypto map entry has been created, you cannot change the parameters specified at the global configuration level because these parameters determine which of the configuration commands are valid at the crypto map level. For example, once a map entry has been created as **ipsec-isakmp**, you cannot change it to **ipsec-manual** or **cisco**; you must delete and reenter the map entry.

After you define crypto map entries, you can assign the crypto map set to interfaces using the **crypto map (interface IPSec)** command.

What Crypto Maps Are For

Crypto maps provide two functions: (1) filtering and classifying traffic to be protected and (2) defining the policy to be applied to that traffic. The first use affects the flow of traffic on an interface; the second affects the negotiation performed (via IKE) on behalf of that traffic.

IPSec crypto maps link together definitions of the following:

- What traffic should be protected
- Which IPSec peers the protected traffic can be forwarded to—these are the peers with which a security association can be established
- Which transform sets are acceptable for use with the protected traffic
- How keys and security associations should be used or managed (or what the keys are, if IKE is not used)

Multiple Crypto Map Entries with the Same *map-name* Form a Crypto Map Set

A crypto map set is a collection of crypto map entries, each with a different *seq-num* but the same *map-name*. Therefore, for a given interface, you could have certain traffic forwarded to one IPSec peer with specified security applied to that traffic, and other traffic forwarded to the same or a different IPSec peer with different IPSec security applied. To accomplish this you would create two crypto maps, each with the same *map-name*, but each with a different *seq-num*.

The *seq-num* Argument

The number you assign to the *seq-num* argument should not be arbitrary. This number is used to rank multiple crypto map entries within a crypto map set. Within a crypto map set, a crypto map entry with a lower *seq-num* is evaluated before a map entry with a higher *seq-num*; that is, the map entry with the lower number has a higher priority.

For example, imagine that there is a crypto map set that contains three crypto map entries: mymap 10, mymap 20, and mymap 30. The crypto map set named mymap is applied to interface Serial 0. When traffic passes through the Serial 0 interface, the traffic is evaluated first for mymap 10. If the traffic matches a **permit** entry in the extended access list in mymap 10, the traffic will be processed according to the information defined in mymap 10 (including establishing IPSec security associations when necessary). If the traffic does not match the mymap 10 access list, the traffic will be evaluated for mymap 20, and then mymap 30, until the traffic matches a **permit** entry in a map entry. (If the traffic does not match a **permit** entry in any crypto map entry, it will be forwarded without any IPSec security.)

Dynamic Crypto Maps

Refer to the "Usage Guidelines" section of the **crypto dynamic-map** command for a discussion on dynamic crypto maps.

You should make crypto map entries which reference dynamic map sets the lowest priority map entries, so that inbound security association negotiations requests will try to match the static maps first. Only after the request does not match any of the static maps do you want it to be evaluated against the dynamic map set.

To make a crypto map entry referencing a dynamic crypto map set the lowest priority map entry, give the map entry the highest *seq-num* of all the map entries in a crypto map set.

Create dynamic crypto map entries using the **crypto dynamic-map** command. After you create a dynamic crypto map set, add the dynamic crypto map set to a static crypto map set with the **crypto map** (IPSec global configuration) command using the **dynamic** keyword.

Tunnel Endpoint Discovery

Use the **discover** keyword to enable Tunnel Endpoint Discovery (TED), which allows the initiating router to dynamically determine an IPSec peer for secure IPSec communications.

Examples

The following example shows the minimum required crypto map configuration when IKE will be used to establish the security associations:

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
```

The following example shows the minimum required crypto map configuration when the security associations are manually established:

```
crypto transform-set someset ah-md5-hmac esp-des
crypto map mymap 10 ipsec-manual
 match address 102
 set transform-set someset
 set peer 10.0.0.5
 set session-key inbound ah 256 98765432109876549876543210987654
 set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc
 set session-key inbound esp 256 cipher 0123456789012345
 set session-key outbound esp 256 cipher abcdefabcdefabcd
```

The following example configures an IPSec crypto map set that includes a reference to a dynamic crypto map set.

Crypto map "mymap 10" allows security associations to be established between the router and either (or both) of two remote IPSec peers for traffic matching access list 101. Crypto map "mymap 20" allows either of two transform sets to be negotiated with the remote peer for traffic matching access list 102.

Crypto map entry “mymap 30” references the dynamic crypto map set “mydynamicmap,” which can be used to process inbound security association negotiation requests that do not match “mymap” entries 10 or 20. In this case, if the peer specifies a transform set that matches one of the transform sets specified in “mydynamicmap,” for a flow “permitted” by the access list 103, IPSec will accept the request and set up security associations with the remote peer without previously knowing about the remote peer. If accepted, the resulting security associations (and temporary crypto map entry) are established according to the settings specified by the remote peer.

The access list associated with “mydynamicmap 10” is also used as a filter. Inbound packets that match a **permit** statement in this list are dropped for not being IPSec protected. (The same is true for access lists associated with static crypto maps entries.) Outbound packets that match a **permit** statement without an existing corresponding IPSec SA are also dropped.

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
 set peer 10.0.0.2
crypto map mymap 20 ipsec-isakmp
 match address 102
 set transform-set my_t_set1 my_t_set2
 set peer 10.0.0.3
crypto map mymap 30 ipsec-isakmp dynamic mydynamicmap
!
crypto dynamic-map mydynamicmap 10
 match address 103
 set transform-set my_t_set1 my_t_set2 my_t_set3
```

The following example configures Tunnel Endpoint Discovery on a Cisco router:

```
crypto map testtag 10 ipsec-isakmp dynamic dmap discover
```

Related Commands

Command	Description
crypto dynamic-map	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.
set peer (IPSec)	Specifies an IPSec peer in a crypto map entry.
set pfs	Specifies that IPSec should ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations.
set security-association level per-host	Specifies that separate IPSec security associations should be requested for each source/destination host pair.
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations.

Command	Description
set session-key	Specifies the IPSec session keys within a crypto map entry.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPSec)	Displays the crypto map configuration.

crypto map (interface IPSec)

To apply a previously defined crypto map set to an interface, use the **crypto map** interface configuration command. To remove the crypto map set from the interface, use the **no** form of this command.

crypto map *map-name*

no crypto map [*map-name*]

Syntax Description

map-name	Name that identifies the crypto map set. This is the name assigned when the crypto map was created. When the **no** form of the command is used, this argument is optional. Any value supplied for the argument is ignored.

Defaults

No crypto maps are assigned to interfaces.

Command Modes

Interface configuration

Command History

Release	Modification
11.2	This command was introduced.

Usage Guidelines

Use this command to assign a crypto map set to an interface. You must assign a crypto map set to an interface before that interface can provide IPSec services. Only one crypto map set can be assigned to an interface. If multiple crypto map entries have the same *map-name* but a different *seq-num*, they are considered to be part of the same set and will all be applied to the interface. The crypto map entry with the lowest *seq-num* is considered the highest priority and will be evaluated first. A single crypto map set can contain a combination of **cisco**, **ipsec-isakmp**, and **ipsec-manual** crypto map entries.

Examples

The following example assigns crypto map set "mymap" to the S0 interface. When traffic passes through S0, the traffic will be evaluated against all the crypto map entries in the "mymap" set. When outbound traffic matches an access list in one of the "mymap" crypto map entries, a security association will be established per that crypto map entry's configuration (if no security association or connection already exists).

```
interface S0
 crypto map mymap
```

Related Commands

Command	Description
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
show crypto map (IPSec)	Displays the crypto map configuration.

crypto map local-address

To specify and name an identifying interface to be used by the crypto map for IPSec traffic, use the **crypto map local-address** global configuration command. To remove this command from the configuration, use the **no** form of this command.

crypto map *map-name* **local-address** *interface-id*

no crypto map *map-name* **local-address**

Syntax Description

map-name	Name that identifies the crypto map set. This is the name assigned when the crypto map was created.
interface-id	The identifying interface that should be used by the router to identify itself to remote peers. If Internet Key Exchange is enabled and you are using a certification authority (CA) to obtain certificates, this should be the interface with the address specified in the CA certificates.

Defaults

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

If you apply the same crypto map to two interfaces and do not use this command, two separate security associations (with different local IP addresses) could be established to the same peer for similar traffic. If you are using the second interface as redundant to the first interface, it could be preferable to have a single security association (with a single local IP address) created for traffic sharing the two interfaces. Having a single security association decreases overhead and makes administration simpler.

This command allows a peer to establish a single security association (and use a single local IP address) that is shared by the two redundant interfaces.

If applying the same crypto map set to more than one interface, the default behavior is as follows:

- Each interface will have its own security association database.
- The IP address of the local interface will be used as the local address for IPSec traffic originating from/destined to that interface.

However, if you use a local-address for that crypto map set, it has multiple effects:

- Only one IPSec security association database will be established and shared for traffic through both interfaces.
- The IP address of the specified interface will be used as the local address for IPSec (and IKE) traffic originating from or destined to that interface.

One suggestion is to use a loopback interface as the referenced local address interface, because the loopback interface never goes down.

Examples

The following example assigns crypto map set "mymap" to the S0 interface and to the S1 interface. When traffic passes through either S0 or S1, the traffic will be evaluated against the all the crypto maps in the "mymap" set. When traffic through either interface matches an access list in one of the "mymap" crypto maps, a security association will be established. This same security association will then apply to both S0 and S1 traffic that matches the originally matched IPSec access list. The local address that IPSec will use on both interfaces will be the IP address of interface loopback0.

```
interface S0
 crypto map mymap

interface S1
 crypto map mymap

crypto map mymap local-address loopback0
```

Related Commands

Command	Description
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.

match address (IPSec)

To specify an extended access list for a crypto map entry, use the **match address** crypto map configuration command. To remove the extended access list from a crypto map entry, use the **no** form of this command.

match address [*access-list-id* | *name*]

no match address [*access-list-id* | *name*]

Syntax Description

access-list-id	(Optional) Identifies the extended access list by its name or number. This value should match the *access-list-number* or *name* argument of the extended access list being matched.
name	(Optional) Identifies the named encryption access list. This name should match the *name* argument of the named encryption access list being matched.

Defaults

No access lists are matched to the crypto map entry.

Command Modes

Crypto map configuration

Command History

Release	Modification
11.2	This command was introduced.

Usage Guidelines

This command is required for all static crypto map entries. If you are defining a dynamic crypto map entry (with the **crypto dynamic-map** command), this command is not required but is strongly recommended.

Use this command to assign an extended access list to a crypto map entry. You also need to define this access list using the **access-list** or **ip access-list extended** commands.

The extended access list specified with this command will be used by IPSec to determine which traffic should be protected by crypto and which traffic does not need crypto protection. (Traffic that is permitted by the access list will be protected. Traffic that is denied by the access list will not be protected in the context of the corresponding crypto map entry.)

Note that the crypto access list is *not* used to determine whether to permit or deny traffic through the interface. An access list applied directly to the interface makes that determination.

The crypto access list specified by this command is used when evaluating both inbound and outbound traffic. Outbound traffic is evaluated against the crypto access lists specified by the interface's crypto map entries to determine if it should be protected by crypto and if so (if traffic matches a **permit** entry) which crypto policy applies. (If necessary, in the case of static IPSec crypto maps, new security associations are established using the data flow identity as specified in the **permit** entry; in the case of dynamic crypto map entries, if no SA exists, the packet is dropped.) After passing the regular access lists at the interface, inbound traffic is evaluated against the crypto access lists specified by the entries of the

interface's crypto map set to determine if it should be protected by crypto and, if so, which crypto policy applies. (In the case of IPSec, unprotected traffic is discarded because it should have been protected by IPSec.)

In the case of IPSec, the access list is also used to identify the flow for which the IPSec security associations are established. In the outbound case, the **permit** entry is used as the data flow identity (in general), while in the inbound case the data flow identity specified by the peer must be "permitted" by the crypto access list.

Examples

The following example shows the minimum required crypto map configuration when IKE will be used to establish the security associations. (This example is for a static crypto map.)

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
```

Related Commands

Command	Description
crypto dynamic-map	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
set peer (IPSec)	Specifies an IPSec peer in a crypto map entry.
set pfs	Specifies that IPSec should ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations.
set security-association level per-host	Specifies that separate IPSec security associations should be requested for each source/destination host pair.
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations.
set session-key	Specifies the IPSec session keys within a crypto map entry.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPSec)	Displays the crypto map configuration.

mode (IPSec)

To change the mode for a transform set, use the **mode** crypto transform configuration command. To reset the mode to the default value of tunnel mode, use the **no** form of the command.

mode [**tunnel** | **transport**]

no mode

Syntax Description

tunnel | **transport**	(Optional) Specifies the mode for a transform set: either tunnel or transport mode. If neither **tunnel** nor **transport** is specified, the default (tunnel mode) is assigned.

Defaults

Tunnel mode

Command Modes

Crypto transform configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

Use this command to change the mode specified for the transform. This setting is only used when the traffic to be protected has the same IP addresses as the IPSec peers (this traffic can be encapsulated either in tunnel or transport mode). This setting is ignored for all other traffic (all other traffic is encapsulated in tunnel mode).

If the traffic to be protected has the same IP address as the IP Security peers and transport mode is specified, during negotiation the router will request transport mode but will accept either transport or tunnel mode. If tunnel mode is specified, the router will request tunnel mode and will accept only tunnel mode.

After you define a transform set, you are put into the crypto transform configuration mode. While in this mode you can change the mode to either tunnel or transport. This change applies only to the transform set just defined.

If you do not change the mode when you first define the transform set, but later decide you want to change the mode for the transform set, you must re-enter the transform set (specifying the transform name and all its transforms) and then change the mode.

If you use this command to change the mode, the change will only affect the negotiation of subsequent IPSec security associations via crypto map entries which specify this transform set. (If you want the new settings to take effect sooner, you can clear all or part of the security association database. See the **clear crypto sa** command for more details.

Tunnel Mode

With tunnel mode, the entire original IP packet is protected (encrypted, authenticated, or both) and is encapsulated by the IPSec headers and trailers (an Encapsulation Security Protocol header and trailer, an Authentication Header, or both). Then a new IP header is prefixed to the packet, specifying the IPSec endpoints as the source and destination.

Tunnel mode can be used with any IP traffic. Tunnel mode must be used if IPSec is protecting traffic from hosts behind the IPSec peers. For example, tunnel mode is used with Virtual Private Networks (VPNs) where hosts on one protected network send packets to hosts on a different protected network via a pair of IPSec peers. With VPNs, the IPSec peers "tunnel" the protected traffic between the peers while the hosts on their protected networks are the session endpoints.

Transport Mode

With transport mode, only the payload (data) of the original IP packet is protected (encrypted, authenticated, or both). The payload is encapsulated by the IPSec headers and trailers (an ESP header and trailer, an AH header, or both). The original IP headers remain intact and are not protected by IPSec.

Use transport mode only when the IP traffic to be protected has IPSec peers as both the source and destination. For example, you could use transport mode to protect router management traffic. Specifying transport mode allows the router to negotiate with the remote peer whether to use transport or tunnel mode.

Examples

The following example defines a transform set and changes the mode to transport mode. The mode value only applies to IP traffic with the source and destination addresses at the local and remote IPSec peers.

```
crypto ipsec transform-set newer esp-des esp-sha-hmac
  mode transport
  exit
```

Related Commands

Command	Description
crypto ipsec transform-set	Defines a transform set—an acceptable combination of security protocols and algorithms.

set peer (IPSec)

To specify an IP Security peer in a crypto map entry, use the **set peer** crypto map configuration command. To remove an IPSec peer from a crypto map entry, use the **no** form of this command.

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

no set peer {*hostname* | *ip-address*}

Syntax Description

hostname	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).
ip-address	Specifies the IPSec peer by its IP address.

Defaults

No peer is defined by default.

Command Modes

Crypto map configuration

Command History

Release	Modification
11.2	This command was introduced.

Usage Guidelines

Use this command to specify an IPSec peer for a crypto map.

This command is required for all static crypto maps. If you are defining a dynamic crypto map (with the **crypto dynamic-map** command), this command is not required, and in most cases is not used (because, in general, the peer is unknown).

For **ipsec-isakmp** crypto map entries, you can specify multiple peers by repeating this command. The peer that packets are actually sent to is determined by the last peer that the router heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, Internet Key Exchange tries the next peer on the crypto map list.

For **ipsec-manual** crypto entries, you can specify only one IPSec peer per crypto map. If you want to change the peer, you must first delete the old peer and then specify the new peer.

You can specify the remote IPSec peer by its host name only if the host name is mapped to the peer's IP address in a Domain Name Server or if you manually map the host name to the IP address with the **ip host** command.

Examples

The following example shows a crypto map configuration when IKE will be used to establish the security associations. In this example, a security association could be set up to either the IPSec peer at 10.0.0.1 or the peer at 10.0.0.2.

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
```

```
set peer 10.0.0.2
```

Related Commands

Command	Description
crypto dynamic-map	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.
set pfs	Specifies that IPSec should ask for PFS when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations.
set security-association level per-host	Specifies that separate IPSec security associations should be requested for each source/destination host pair.
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations.
set session-key	Specifies the IPSec session keys within a crypto map entry.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPSec)	Displays the crypto map configuration.

set pfs

To specify that IP Security should ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations, use the **set pfs** crypto map configuration command. To specify that IPSec should not request PFS, use the **no** form of the command.

set pfs [**group1** | **group2**]

no set pfs

Syntax Description

group1	(Optional) Specifies that IPSec should use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
group2	(Optional) Specifies that IPSec should use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

Defaults

By default, PFS is not requested. If no group is specified with this command, **group1** is used as the default.

Command Modes

Crypto map configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command is only available for **ipsec-isakmp** crypto map entries and dynamic crypto map entries.

During negotiation, this command causes IPSec to request PFS when requesting new security associations for the crypto map entry. The default (**group1**) is sent if the **set pfs** statement does not specify a group. If the peer initiates the negotiation and the local configuration specifies PFS, the remote peer must perform a PFS exchange or the negotiation will fail. If the local configuration does not specify a group, a default of **group1** will be assumed, and an offer of either **group1** or **group2** will be accepted. If the local configuration specifies **group2**, that group *must* be part of the peer's offer or the negotiation will fail. If the local configuration does not specify PFS it will accept any offer of PFS from the peer.

PFS adds another level of security because if one key is ever cracked by an attacker then only the data sent with that key will be compromised. Without PFS, data sent with other keys could be also compromised.

With PFS, every time a new security association is negotiated, a new Diffie-Hellman exchange occurs. (This exchange requires additional processing time.)

The 1024-bit Diffie-Hellman prime modulus group, **group2**, provides more security than **group1**, but requires more processing time than **group1**.

Examples

The following example specifies that PFS should be used whenever a new security association is negotiated for the crypto map "mymap 10":

```
crypto map mymap 10 ipsec-isakmp
 set pfs group2
```

Related Commands

Command	Description
crypto dynamic-map	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.
set peer (IPSec)	Specifies an IPSec peer in a crypto map entry.
set security-association level per-host	Specifies that separate IPSec security associations should be requested for each source/destination host pair.
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPSec)	Displays the crypto map configuration.

set security-association level per-host

To specify that separate IP Security security associations should be requested for each source/destination host pair, use the **set security-association level per-host** crypto map configuration command. To specify that one security association should be requested for each crypto map access list **permit** entry, use the **no** form of this command.

set security-association level per-host

no set security-association level per-host

Syntax Description

This command has no arguments or keywords.

Defaults

For a given crypto map, all traffic between two IPSec peers matching a single crypto map access list **permit** entry will share the same security association.

Command Modes

Crypto map configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command is only available for **ipsec-isakmp** crypto map entries and is not supported for dynamic crypto map entries.

When you use this command, you need to specify that a separate security association should be used for each source/destination host pair.

Normally, within a given crypto map, IPSec will attempt to request security associations at the granularity specified by the access list entry. For example, if the access list entry permits IP protocol traffic between subnet A and subnet B, IPSec will attempt to request security associations between subnet A and subnet B (for any IP protocol), and unless finer-grained security associations are established (by a peer request), all IPSec-protected traffic between these two subnets would use the same security association.

This command causes IPSec to request separate security associations for each source/destination host pair. In this case, each host pairing (where one host was in subnet A and the other host was in subnet B) would cause IPSec to request a separate security association.

With this command, one security association would be requested to protect traffic between host A and host B, and a different security association would be requested to protect traffic between host A and host C.

The access list entry can specify local and remote subnets, or it can specify a host-and-subnet combination. If the access list entry specifies protocols and ports, these values are applied when establishing the unique security associations.

Use this command with care, as multiple streams between given subnets can rapidly consume system resources.

Examples

The following example shows what happens with an access list entry of **permit ip 1.1.1.0 0.0.0.255 2.2.2.0 0.0.0.255** and a per-host level:

- A packet from 1.1.1.1 to 2.2.2.1 will initiate a security association request, which would look like it originated via **permit ip host 1.1.1.1 host 2.2.2.1**.
- A packet from 1.1.1.1 to 2.2.2.2 will initiate a security association request, which would look like it originated via **permit ip host 1.1.1.1 host 2.2.2.2**.
- A packet from 1.1.1.2 to 2.2.2.1 will initiate a security association request, which would look like it originated via **permit ip host 1.1.1.2 host 2.2.2.1**.

Without the per-host level, any of the above packets will initiate a single security association request originated via **permit ip 1.1.1.0 0.0.0.255 2.2.2.0 0.0.0.255**.

Related Commands

Command	Description
crypto dynamic-map	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.
set peer (IPSec)	Specifies an IPSec peer in a crypto map entry.
set pfs	Specifies that IPSec should ask for PFS when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations.
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPSec)	Displays the crypto map configuration.

set security-association lifetime

To override (for a particular crypto map entry) the global lifetime value, which is used when negotiating IP Security security associations, use the **set security-association lifetime** crypto map configuration command. To reset a crypto map entry's lifetime value to the global value, use the **no** form of this command.

set security-association lifetime {**seconds** *seconds* | **kilobytes** *kilobytes*}

no set security-association lifetime {**seconds** | **kilobytes**}

Syntax Description

seconds *seconds*	Specifies the number of seconds a security association will live before expiring.
kilobytes *kilobytes*	Specifies the volume of traffic (in kilobytes) that can pass between IPSec peers using a given security association before that security association expires.

Defaults

The crypto map's security associations are negotiated according to the global lifetimes.

Command Modes

Crypto map configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command is available only for **ipsec-isakmp** crypto map entries and dynamic crypto map entries.

IPSec security associations use shared secret keys. These keys and their security associations time out together.

Assuming that the particular crypto map entry has lifetime values configured, when the router requests new security associations during security association negotiation, it will specify its crypto map lifetime value in the request to the peer; it will use this value as the lifetime of the new security associations. When the router receives a negotiation request from the peer, it will use the smaller of the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

There are two lifetimes: a "timed" lifetime and a "traffic-volume" lifetime. The session keys/security association expires after the first of these lifetimes is reached.

If you change a lifetime, the change will not be applied to existing security associations, but will be used in subsequent negotiations to establish security associations for data flows supported by this crypto map entry. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command. Refer to the **clear crypto sa** command for more detail.

To change the timed lifetime, use the **set security-association lifetime seconds** form of the command. The timed lifetime causes the keys and security association to time out after the specified number of seconds have passed.

To change the traffic-volume lifetime, use the **set security-association lifetime kilobytes** form of the command. The traffic-volume lifetime causes the key and security association to time out after the specified amount of traffic (in kilobytes) has been protected by the security association's key.

Shorter lifetimes can make it harder to mount a successful key recovery attack, because the attacker has less data encrypted under the same key to work with. However, shorter lifetimes need more CPU processing time.

The lifetime values are ignored for manually established security associations (security associations installed via an **ipsec-manual** crypto map entry).

How These Lifetimes Work

Assuming that the particular crypto map entry does not have lifetime values configured, when the router requests new security associations it will specify its global lifetime values in the request to the peer; it will use this value as the lifetime of the new security associations. When the router receives a negotiation request from the peer, it will use the smaller of either the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

The security association (and corresponding keys) will expire according to whichever occurs sooner, either after the **seconds** time out or after the **kilobytes** amount of traffic is passed.

A new security association is negotiated *before* the lifetime threshold of the existing security association is reached, to ensure that a new security association is ready for use when the old one expires. The new security association is negotiated either 30 seconds before the **seconds** lifetime expires or when the volume of traffic through the tunnel reaches 256 kilobytes less than the **kilobytes** lifetime (whichever occurs first).

If no traffic has passed through the tunnel during the entire life of the security association, a new security association is not negotiated when the lifetime expires. Instead, a new security association will be negotiated only when IPSec sees another packet that should be protected.

Examples

The following example shortens the timed lifetime for a particular crypto map entry, because there is a higher risk that the keys could be compromised for security associations belonging to the crypto map entry. The traffic-volume lifetime is not changed because there is not a high volume of traffic anticipated for these security associations. The timed lifetime is shortened to 2700 seconds (45 minutes).

```
crypto map mymap 10 ipsec-isakmp
 set security-association lifetime seconds 2700
```

Related Commands

Command	Description
crypto dynamic-map	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
crypto ipsec security-association lifetime	Changes global lifetime values used when negotiating IPSec security associations.
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.

Command	Description
set peer (IPSec)	Specifies an IPSec peer in a crypto map entry.
set pfs	Specifies that IPSec should ask for PFS when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations.
set security-association level per-host	Specifies that separate IPSec security associations should be requested for each source/destination host pair.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPSec)	Displays the crypto map configuration.

set session-key

To manually specify the IP Security session keys within a crypto map entry, use the **set session-key** crypto map configuration command. This command is only available for **ipsec-manual** crypto map entries. To remove IPSec session keys from a crypto map entry, use the **no** form of this command.

set session-key {**inbound** | **outbound**} **ah** *spi hex-key-string*

set session-key {**inbound** | **outbound**} **esp** *spi cipher hex-key-string* [**authenticator** *hex-key-string*]

no set session-key {**inbound** | **outbound**} **ah**

no set session-key {**inbound** | **outbound**} **esp**

Syntax Description

inbound	Sets the inbound IPSec session key. (You must set both inbound and outbound keys.)
outbound	Sets the outbound IPSec session key. (You must set both inbound and outbound keys.)
ah	Sets the IPSec session key for the Authentication Header protocol. Use when the crypto map entry's transform set includes an AH transform.
esp	Sets the IPSec session key for the Encapsulation Security Protocol. Use when the crypto map entry's transform set includes an ESP transform.
spi	Specifies the security parameter index (SPI), a number that is used to uniquely identify a security association. The SPI is an arbitrary number you assign in the range of 256 to 4,294,967,295 (FFFF FFFF). You can assign the same SPI to both directions and both protocols. However, not all peers have the same flexibility in SPI assignment. For a given destination address/protocol combination, unique SPI values must be used. The destination address is that of the router if inbound, the peer if outbound.
hex-key-string	Specifies the session key; enter in hexadecimal format. This is an arbitrary hexadecimal string of 8, 16, or 20 bytes. If the crypto map's transform set includes a DES algorithm, specify at least 8 bytes per key. If the crypto map's transform set includes an MD5 algorithm, specify at least 16 bytes per key. If the crypto map's transform set includes an SHA algorithm, specify 20 bytes per key. Keys longer than the above sizes are simply truncated.
cipher	Indicates that the key string is to be used with the ESP encryption transform.
authenticator	(Optional) Indicates that the key string is to be used with the ESP authentication transform. This argument is required only when the crypto map entry's transform set includes an ESP authentication transform.

Defaults

No session keys are defined by default.

Command Modes Crypto map configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines Use this command to define IPSec keys for security associations via **ipsec-manual** crypto map entries. (In the case of **ipsec-isakmp** crypto map entries, the security associations with their corresponding keys are automatically established via the IKE negotiation.)

If the crypto map's transform set includes an AH protocol, you must define IPSec keys for AH for both inbound and outbound traffic. If the crypto map's transform set includes an ESP encryption protocol, you must define IPSec keys for ESP encryption for both inbound and outbound traffic. If your transform set includes an ESP authentication protocol, you must define IPSec keys for ESP authentication for inbound and outbound traffic.

When you define multiple IPSec session keys within a single crypto map, you can assign the same security parameter index (SPI) number to all the keys. The SPI is used to identify the security association used with the crypto map. However, not all peers have the same flexibility in SPI assignment. You should coordinate SPI assignment with your peer's operator, making certain that the same SPI is not used more than once for the same destination address/protocol combination.

Security associations established via this command do not expire (unlike security associations established via IKE).

Session keys at one peer must match the session keys at the remote peer.

If you change a session key, the security association using the key will be deleted and reinitialized.

Examples The following example shows a crypto map entry for manually established security associations. The transform set "t_set" includes only an AH protocol.

```
crypto ipsec transform-set t_set ah-sha-hmac

crypto map mymap 20 ipsec-manual
 match address 102
 set transform-set t_set
 set peer 10.0.0.21
 set session-key inbound ah 300 1111111111111111111111111111111111111111
 set session-key outbound ah 300 2222222222222222222222222222222222222222
```

The following example shows a crypto map entry for manually established security associations. The transform set "someset" includes both an AH and an ESP protocol, so session keys are configured for both AH and ESP for both inbound and outbound traffic. The transform set includes both encryption and authentication ESP transforms, so session keys are created for both using the **cipher** and **authenticator** keywords.

```
crypto ipsec transform-set someset ah-sha-hmac esp-des esp-sha-hmac

crypto map mymap 10 ipsec-manual
 match address 101
 set transform-set someset
 set peer 10.0.0.1
 set session-key inbound ah 300 9876543210987654321098765432109876543210
 set session-key outbound ah 300 fedcbafedcbafedcbafedcbafedcbafedcbafedc
 set session-key inbound esp 300 cipher 0123456789012345
```

```
 authenticator 000011112222333344445555666677778888999
set session-key outbound esp 300 cipher abcdefabcdefabcd
 authenticator 99998888777766665555444433332222111110000
```

Related Commands

Command	Description
crypto map (global IPSec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.
set peer (IPSec)	Specifies an IPSec peer in a crypto map entry.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPSec)	Displays the crypto map configuration.

set transform-set

To specify which transform sets can be used with the crypto map entry, use the **set transform-set** crypto map configuration command. To remove all transform sets from a crypto map entry, use the **no** form of this command.

set transform-set *transform-set-name* [*transform-set-name2...transform-set-name6*]

no set transform-set

Syntax Description

transform-set-name	Name of the transform set. For an **ipsec-manual** crypto map entry, you can specify only one transform set. For an **ipsec-isakmp** or dynamic crypto map entry, you can specify up to 6 transform sets.

Defaults

No transform sets are included by default.

Command Modes

Crypto map configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command is required for all static and dynamic crypto map entries.

Use this command to specify which transform sets to include in a crypto map entry.

For an **ipsec-isakmp** crypto map entry, you can list multiple transform sets with this command. List the higher priority transform sets first.

If the local router initiates the negotiation, the transform sets are presented to the peer in the order specified in the crypto map entry. If the peer initiates the negotiation, the local router accepts the first transform set that matches one of the transform sets specified in the crypto map entry.

The first matching transform set that is found at both peers is used for the security association. If no match is found, IPSec will not establish a security association. The traffic will be dropped because there is no security association to protect the traffic.

For an **ipsec-manual** crypto map entry, you can specify only one transform set. If the transform set does not match the transform set at the remote peer's crypto map, the two peers will fail to correctly communicate because the peers are using different rules to process the traffic.

If you want to change the list of transform sets, re-specify the new list of transform sets to replace the old list. This change is only applied to crypto map entries that reference this transform set. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command.

Any transform sets included in a crypto map must previously have been defined using the **crypto ipsec transform-set** command.

Examples

The following example defines two transform sets and specifies that they can both be used within a crypto map entry. (This example applies only when IKE is used to establish security associations. With crypto maps used for manually established security associations, only one transform set can be included in a given crypto map entry.)

```
crypto ipsec transform-set my_t_set1 esp-des esp-sha-hmac
crypto ipsec transform-set my_t_set2 ah-sha-hmac esp-des esp-sha-hmac

crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1 my_t_set2
 set peer 10.0.0.1
 set peer 10.0.0.2
```

In this example, when traffic matches access list 101, the security association can use either transform set “my_t_set1” (first priority) or “my_t_set2” (second priority) depending on which transform set matches the remote peer’s transform sets.

show crypto dynamic-map

To view a dynamic crypto map set, use the **show crypto dynamic-map** EXEC command.

show crypto dynamic-map [**tag** *map-name*]

Syntax Description

tag *map-name*	(Optional) Displays only the crypto dynamic map set with the specified *map-name*.

Command Modes

EXEC

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

Use the **show crypto dynamic-map** command to view a dynamic crypto map set.

Examples

The following is sample output for the **show crypto dynamic-map** command:

```
Router# show crypto dynamic-map

Crypto Map Template"dyn1" 10
        Extended IP access list 152
            access-list 152 permit ip
                source: addr = 172.21.114.67/0.0.0.0
                dest:   addr = 0.0.0.0/255.255.255.255
        Current peer: 0.0.0.0
        Security association lifetime: 4608000 kilobytes/120 seconds
        PFS (Y/N): N
        Transform sets={ tauth, t1, }
```

The following partial configuration was in effect when the above **show crypto dynamic-map** command was issued:

```
crypto ipsec security-association lifetime seconds 120
!
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto ipsec transform-set tauth ah-sha-hmac
!
crypto dynamic-map dyn1 10
 set transform-set tauth t1
 match address 152
crypto map to-router local-address Ethernet0
crypto map to-router 10 ipsec-isakmp
 set peer 172.21.114.123
 set transform-set tauth t1
 match address 150
crypto map to-router 20 ipsec-isakmp dynamic dyn1
!
access-list 150 permit ip host 172.21.114.67 host 172.21.114.123
access-list 150 permit ip host 15.15.15.1 host 172.21.114.123
access-list 150 permit ip host 15.15.15.1 host 8.8.8.1
```

```
access-list 152 permit ip host 172.21.114.67 any
```

show crypto engine accelerator logs

To display information about the last 32 CryptoGraphics eXtensions (CGX) Library packet processing commands and associated parameters sent from the VPN module driver to the VPN module hardware, use the **show crypto engine accelerator logs** command in privileged EXEC mode.

show crypto engine accelerator logs

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)XC	This command was introduced on the Cisco 1720 and Cisco 1750 platforms.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.

Usage Guidelines

Use this command when encrypted traffic is sent to the router and a problem with the encryption module is suspected. Use the **debug crypto engine accelerator logs** command to enable command logging *before* using this command.

Note The **show crypto engine accelerator logs** command is intended only for Cisco Systems TAC personnel to collect debugging information.

Examples

The following is sample output for the **show crypto engine accelerator logs** command:

```
Router# show crypto engine accelerator logs

Contents of packet log (current index = 20):

tag = 0x5B02, cmd = 0x5000
param[0] = 0x000E, param[1] = 0x57E8
param[2] = 0x0008, param[3] = 0x0000
param[4] = 0x0078, param[5] = 0x0004
param[6] = 0x142C, param[7] = 0x142C
param[8] = 0x0078, param[9] = 0x000C
tag = 0x5B03, cmd = 0x4100
param[0] = 0x000E, param[1] = 0x583C
param[2] = 0x0034, param[3] = 0x0040
param[4] = 0x00B0, param[5] = 0x0004
param[6] = 0x1400, param[7] = 0x1400
param[8] = 0x0020, param[9] = 0x000C
```

```
tag = 0x5C00, cmd = 0x4100
param[0] = 0x000E, param[1] = 0x57BC
param[2] = 0x0034, param[3] = 0x0040
param[4] = 0x00B0, param[5] = 0x0004
param[6] = 0x1400, param[7] = 0x1400
param[8] = 0x0020, param[9] = 0x000C
            •
            •
            •
tag = 0x5A01, cmd = 0x4100
param[0] = 0x000E, param[1] = 0x593C
param[2] = 0x0034, param[3] = 0x0040
param[4] = 0x00B0, param[5] = 0x0004
param[6] = 0x1400, param[7] = 0x1400
param[8] = 0x0020, param[9] = 0x000C

Contents of cgx log (current index = 12):

cmd = 0x0074 ret = 0x0000
param[0] = 0x0010, param[1] = 0x028E
param[2] = 0x0039, param[3] = 0x0D1E
param[4] = 0x0100, param[5] = 0x0000
param[6] = 0x0000, param[7] = 0x0000
param[8] = 0x0000, param[9] = 0x0000
cmd = 0x0062 ret = 0x0000
param[0] = 0x0035, param[1] = 0x1BE0
param[2] = 0x0100, param[3] = 0x0222
param[4] = 0x0258, param[5] = 0x0000
param[6] = 0x0000, param[7] = 0x0000
param[8] = 0x0000, param[9] = 0x0000
cmd = 0x0063 ret = 0x0000
param[0] = 0x0222, param[1] = 0x0258
param[2] = 0x0000, param[3] = 0x0000
param[4] = 0x0000, param[5] = 0x0000
param[6] = 0x0000, param[7] = 0x020A
param[8] = 0x002D, param[9] = 0x0000
            •
            •
            •
cmd = 0x0065 ret = 0x0000
param[0] = 0x0222, param[1] = 0x0258
param[2] = 0x0010, param[3] = 0x028E
param[4] = 0x00A0, param[5] = 0x0008
param[6] = 0x0001, param[7] = 0x0000
param[8] = 0x0000, param[9] = 0x0000
```

Related Commands

Command	Description
debug crypto engine acclerator logs	Enables logging of commands and associated parameters sent from the VPN module driver to the VPN module hardware using a debug flag.
crypto engine accelerator	Enables or disables the IPSec accelerator.

show crypto engine accelerator sa-database

To display active (in-use) entries in the platform-specific virtual private network (VPN) module database, use the **show crypto engine accelerator sa-database** command in privileged EXEC configuration mode.

show crypto engine accelerator sa-database

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)XC	This command was introduced on the Cisco 1720 and Cisco 1750 platforms.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.

Usage Guidelines

Use this command when encrypted traffic is sent to the router and a problem with the encryption module is suspected.

Note The **show crypto engine accelerator sa-database** command is intended only for Cisco Systems TAC personnel to collect debugging information.

Examples

The following is sample output for the **show crypto engine accelerator sa-database** command:

```
Router# show crypto engine accelerator sa-database
Flow Summary
        Index   Algorithms
        005      tunnel inbound  esp-md5-hmac esp-des ah-sha-hmac
        006      tunnel outbound esp-md5-hmac esp-des ah-sha-hmac
        007      tunnel inbound  esp-md5-hmac esp-des ah-sha-hmac
        008      tunnel outbound esp-md5-hmac esp-des ah-sha-hmac
        009      tunnel inbound  esp-md5-hmac esp-des ah-sha-hmac
        010      tunnel outbound esp-md5-hmac esp-des ah-sha-hmac
SA Summary:
        Index   DH-Index        Algorithms
        003     001(deleted)    DES SHA
        004     002(deleted)    DES SHA
DH Summary
        Index Group Config
```

Related Commands

Command	Description
debug crypto engine acclerator logs	Enables logging of commands and associated parameters sent from the VPN module driver to the VPN module hardware using a debug flag.
crypto engine accelerator	Enables or disables the IPSec accelerator.

show crypto ipsec sa

To view the settings used by current security associations, use the **show crypto ipsec sa** EXEC command.

show crypto ipsec sa [**map** *map-name* | **address** | **identity**] [**detail**]

Syntax Description

map *map-name*	(Optional) Displays any existing security associations created for the crypto map set named *map-name*.
address	(Optional) Displays the all existing security associations, sorted by the destination address (either the local address or the address of the IP Security remote peer) and then by protocol (Authentication Header or Encapsulation Security Protocol).
identity	(Optional) Displays only the flow information. It does not show the security association information.
detail	(Optional) Displays detailed error counters. (The default is the high level send/receive error counters.)

Command Modes

EXEC

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

If no keyword is used, all security associations are displayed. They are sorted first by interface, and then by traffic flow (for example, source/destination address, mask, protocol, port). Within a flow, the security associations are listed by protocol (ESP/AH) and direction (inbound/outbound).

Examples

The following is sample output for the **show crypto ipsec sa** command:

```
Router# show crypto ipsec sa

interface: Ethernet0
    Crypto map tag: router-alice, local addr. 172.21.114.123

   local  ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
   current_peer: 172.21.114.67
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0

     local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67
     path mtu 1500, media mtu 1500
     current outbound spi: 20890A6F
```

```
     inbound esp sas:
      spi: 0x257A1039(628756537)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 26, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y

     inbound ah sas:

     outbound esp sas:
      spi: 0x20890A6F(545852015)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 27, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y

     outbound ah sas:

interface: Tunnel0
    Crypto map tag: router-alice, local addr. 172.21.114.123

   local  ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
   current_peer: 172.21.114.67
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0

     local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67
     path mtu 1500, media mtu 1500
     current outbound spi: 20890A6F

     inbound esp sas:
      spi: 0x257A1039(628756537)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 26, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y

     inbound ah sas:

     outbound esp sas:
      spi: 0x20890A6F(545852015)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 27, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y

     outbound ah sas:
```

show crypto ipsec security-association lifetime

To view the security-association lifetime value configured for a particular crypto map entry, use the **show crypto ipsec security-association lifetime** EXEC command.

show crypto ipsec security-association lifetime

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.3 T	This command was introduced.

Examples

The following is sample output for the **show crypto ipsec security-association lifetime** command:

```
Router# show crypto ipsec security-association lifetime

Security-association lifetime: 4608000 kilobytes/120 seconds
```

The following configuration was in effect when the previous **show crypto ipsec security-association lifetime** command was issued:

```
crypto ipsec security-association lifetime seconds 120
```

show crypto ipsec transform-set

To view the configured transform sets, use the **show crypto ipsec transform-set** EXEC command.

show crypto ipsec transform-set [**tag** *transform-set-name*]

Syntax Description

tag *transform-set-name*	(Optional) Displays only the transform sets with the specified *transform-set-name*.

Command Modes

EXEC

Command History

Release	Modification
11.3 T	This command was introduced.

Examples

The following is sample output for the **show crypto ipsec transform-set** command:

```
Router# show crypto ipsec transform-set

Transform set combined-des-sha: { esp-des esp-sha-hmac  }
   will negotiate = { Tunnel,  },

Transform set combined-des-md5: { esp-des esp-md5-hmac  }
   will negotiate = { Tunnel,  },

Transform set t1: { esp-des esp-md5-hmac  }
   will negotiate = { Tunnel,  },

Transform set t100: { ah-sha-hmac  }
   will negotiate = { Transport,  },

Transform set t2: { ah-sha-hmac  }
   will negotiate = { Tunnel,  },
   { esp-des  }
   will negotiate = { Tunnel,  },
```

The following configuration was in effect when the previous **show crypto ipsec transform-set** command was issued:

```
crypto ipsec transform-set combined-des-sha esp-des esp-sha-hmac
crypto ipsec transform-set combined-des-md5 esp-des esp-md5-hmac
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto ipsec transform-set t100 ah-sha-hmac
 mode transport
crypto ipsec transform-set t2 ah-sha-hmac esp-des
```

show crypto map (IPSec)

To view the crypto map configuration, use the **show crypto map** EXEC command.

show crypto map [**interface** *interface* | **tag** *map-name*]

Syntax Description

interface *interface*	(Optional) Displays only the crypto map set applied to the specified interface.
tag *map-name*	(Optional) Displays only the crypto map set with the specified *map-name*.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.

Examples

The following is sample output for the **show crypto map** command:

```
Router# show crypto map

Crypto Map: "router-alice" idb: Ethernet0 local address: 172.21.114.123

Crypto Map "router-alice" 10 ipsec-isakmp
        Peer = 172.21.114.67
        Extended IP access list 141
            access-list 141 permit ip
                source: addr = 172.21.114.123/0.0.0.0
                dest:   addr = 172.21.114.67/0.0.0.0
        Current peer: 172.21.114.67
        Security-association lifetime: 4608000 kilobytes/120 seconds
        PFS (Y/N): N
        Transform sets={ t1, }
```

The following configuration was in effect when the above **show crypto map** command was issued:

```
crypto map router-alice local-address Ethernet0
crypto map router-alice 10 ipsec-isakmp
 set peer 172.21.114.67
 set transform-set t1
 match address 141
```

The following is sample output for the **show crypto map** command when manually established security associations are used:

```
Router# show crypto map

Crypto Map "multi-peer" 20 ipsec-manual
        Peer = 172.21.114.67
        Extended IP access list 120
            access-list 120 permit ip
                source: addr = 1.1.1.1/0.0.0.0
                dest:   addr = 1.1.1.2/0.0.0.0
        Current peer: 172.21.114.67
        Transform sets={ t2, }
        Inbound esp spi: 0,
         cipher key: ,
```

```
 auth_key: ,
Inbound ah spi: 256,
    key: 0102030405060708090102030405060708090102030405060708090,
Outbound esp spi: 0
 cipher key: ,
 auth key: ,
Outbound ah spi: 256,
    key: 0102030405060708090102030405060708090102030405060708090,
```

The following configuration was in effect when the above **show crypto map** command was issued:

```
crypto map multi-peer 20 ipsec-manual
 set peer 172.21.114.67
 set session-key inbound ah 256
0102030405060708090102030405060708090102030405060708090
 set session-key outbound ah 256
0102030405060708090102030405060708090102030405060708090
 set transform-set t2
 match address 120
```