Configuring the SA-VAM2+

This chapter contains the information and procedures needed to configure the VPN Acceleration Module 2+ (SA-VAM2+). This chapter contains the following sections:

Overview

The SA-VAM2+ provides encryption services for any interface in the Cisco 7301 router and the Cisco 7200VXR series routers with a NPE-225, NPE-400, NPE-G1 or NPE-G2 processor. If you have previously configured IPSec on the router and you install a SA-VAM2+, the SA-VAM2+ automatically performs encryption services. If you install a second SA-VAM2+, both SA-VAM2+s should be automatically enabled.

Note   The Cisco 7301 router supports a single SA-VAM2+.

When installing two SA-VAM2+s on the Cisco 7200VXR series routers, per packet load balancing is not supported. With dual SA-VAM2+s installed, load balancing is done on a per IPSec tunnel basis, rather than on a per packet basis.

There are no interfaces to configure on the SA-VAM2+.

This section only contains basic configuration information for enabling encryption and IPSec tunneling services. Refer to the "IP Security and Encryption" part of the *Security Configuration Guide* and the *Security Command Reference* guide for detailed configuration information on IPSec, IKE, and CA.

Configuration Tasks

On power up if the enabled LED is on, the SA-VAM2+ is fully functional and does not require any configuration commands. However, for the SA-VAM2+ to provide encryption services, you must complete the steps in the following sections:

> **Note** You can configure a static crypto map, create a dynamic crypto map, or add a dynamic crypto map into a static crypto map. Refer to the online publication, *Configuring the VPN Acceleration Module* at http://www.cisco.com/univercd/cc/td/doc/product/core/7100/7100pacn/vam1/vamconf.htm.

Optionally, you can configure certification authority (CA) interoperability (refer to the "Configuring Certification Authority Interoperability" chapter in the *Security Configuration Guide*).

Using the EXEC Command Interpreter

You modify the configuration of your router through the software command interpreter called the *EXEC* (also called enable mode). You must enter the privileged level of the EXEC command interpreter with the **enable** command before you can use the **configure** command to configure a new interface or change the existing configuration of an interface. The system prompts you for a password if one has been set.

The system prompt for the privileged level ends with a pound sign (#) instead of an angle bracket (>). At the console terminal, use the following procedure to enter the privileged level:

Step 1 At the user-level EXEC prompt, enter the **enable** command. The EXEC prompts you for a privileged-level password as follows:

```
Router> enable

Password:
```

Step 2 Enter the password (the password is case sensitive). For security purposes, the password is not displayed. When you enter the correct password, the system displays the privileged-level system prompt (#):

```
Router#
```

This completes the procedure for entering the privileged level of the EXEC command interpreter.

Enabling SA-VAM2+

SA-VAM2+ is enabled by default.

To disable SA-VAM2+, use the following commands, starting in global configuration mode:

	Command	Purpose
Step 1	`no crypto engine accelerator <slot number>`	Disables SA-VAM2+.
Step 2	`crypto engine accelerator <slot number>`	Enables SA-VAM2+.

This completes the procedure for disabling and enabling OIR.

Configuring an IKE Policy

If you do not specify a value for a parameter, the default value is assigned. For information on default values, refer to the "IP Security and Encryption" chapter of the *Security Command Reference* publication.

To configure an IKE policy, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto isakmp policy priority`	Defines an IKE policy and enters Internet Security Association Key Management Protocol (ISAKMP) policy configuration (config-isakmp) mode.
Step 2	`Router(config-isakmp)# encryption {des | 3des | aes | aes 128 | aes 192 | aes 256}`	Specifies the encryption algorithm within an IKE policy. • des—Specifies 56-bit DES as the encryption algorithm. • 3des—Specifies 168-bit DES as the encryption algorithm. • **aes**—Specifies 128-bit AES as the encryption algorithm. • **aes 128**—Specifies 128-bit AES as the encryption algorithm. • **aes 192**—Specifies 192-bit AES as the encryption algorithm. • **aes 256**—Specifies 256-bit AES as the encryption algorithm.
Step 3	`Router(config-isakmp)# authentication {rsa-sig | rsa-encr | pre-share}`	(Optional) Specifies the authentication method within an IKE policy. • **rsa-sig**—Specifies Rivest, Shamir, and Adelman (RSA) signatures as the authentication method. • **rsa-encr**—Specifies RSA encrypted nonces as the authentication method. • **pre-share**—Specifies preshared keys as the authentication method. **Note** If this command is not enabled, the default value (**rsa-sig**) will be used.

	Command	Purpose
Step 4	Router(config-isakmp)# **lifetime** *seconds*	(Optional) Specifies the lifetime of an IKE security association (SA). *seconds*—Number of seconds that each SA should exist before expiring. Use an integer from 60 to 86,400 seconds. **Note** If this command is not enabled, the default value (86,400 seconds [one day]) will be used.
Step 5	Router(config-isakmp)# **hash** {**sha** | **md5**}	(Optional) Specifies the hash algorithm within an IKE policy. • **sha**—Specifies SHA-1 (HMAC variant) as the hash algorithm. • **md5**—Specifies MD5 (HMAC variant) as the hash algorithm. **Note** If this command is not enabled, the default value (**sha**) will be used.
Step 6	Router(config-isakmp)# **group** {**1** | **2** | **5**}	(Optional) Specifies the Diffie-Hellman (DH) group identifier within an IKE policy. **1**—Specifies the 768-bit DH group. **2**—Specifies the 1024-bit DH group. **5**—Specifies the 1536-bit DH group. **Note** If this command is not enabled, the default value (768-bit) will be used.

For detailed information on creating IKE policies, refer to the "Configuring Internet Key Exchange Security Protocol" chapter in the *Security Configuration Guide* publication.

Configuring a Transform Set

See the *Advanced Encryption Standard (AES)* feature module for more information on configuring a transform set.

This section includes the following topics:

- Defining a Transform Set
- IPSec Protocols: AH and ESP
- Selecting Appropriate Transforms
- The Crypto Transform Configuration Mode
- Changing Existing Transforms
- Transform Example

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

Defining a Transform Set

A transform set is a combination of security protocols and algorithms. During the IPSec security association negotiation, peers agree to use a specific transform set to protect a particular data flow.

To define a transform set, use the following commands, starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto ipsec transform-set` *transform-set-name* *transform1* [*transform2* [*transform3*]]	Defines a transform set and enters crypto transform configuration mode. • *transform-set-name*—Specifies the name of the transform set to create (or modify). • *transform1* [*transform2* [*transform3*] [*transform4*]]—Defines the IPSec security protocols and algorithms. Accepted transform values are described in Table 4-1.
Step 2	`Router(cfg-crypto-tran)# mode` [**tunnel** | **transport**]	(Optional) Changes the mode associated with the transform set. The mode setting is only applicable to traffic whose source and destination addresses are the IPSec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)
Step 3	`end`	Exits the crypto transform configuration mode to enabled mode.
Step 4	`clear crypto sa` or `clear crypto sa peer` {*ip-address* | *peer-name*} or `clear crypto sa map` *map-name* or `clear crypto sa spi` *destination-address protocol spi*	Clears existing IPSec security associations so that any changes to a transform set take effect on subsequently established security associations (SAs). (Manually established SAs are reestablished immediately.) Using the **clear crypto sa** command without parameters clears out the full SA database, which clears out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database.

Table 4-1 shows allowed transform combinations for the AH and ESP protocols.

Table 4-1   Allowed Transform Combinations

Transform type	Transform	Description
AH Transform (Pick up to one.)	ah-md5-hmac	AH with the MD5 (Message Digest 5) (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (Secure Hash Algorithm) (HMAC variant) authentication algorithm

Table 4-1   *Allowed Transform Combinations (continued)*

Transform type	Transform	Description
ESP Encryption Transform (Note: If an **ESP Authentication Transform** is used, you must pick one.)	esp-aes	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm
	esp-aes 128	ESP with the 128-bit AES encryption algorithm
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-des	ESP with the 56-bit Data Encryption Standard (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 Lempel-Ziv-Stac (LZS) algorithm

Examples of acceptable transform combinations are as follows:

- **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: AH and ESP

Both the AH and ESP protocols implement security services for IPSec.

AH provides data authentication and antireplay services.

ESP provides packet encryption and optional data authentication and antireplay 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, refer to 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 slightly slower.

- Note that some transforms might not be supported by the IPSec peer.

> **Note** If a user enters an IPSec transform that the hardware (the IPSec peer) does not support, a warning message will be displayed immediately after the **crypto ipsec transform-set** command is entered.

- 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 follow:

- **esp-aes** and **esp-sha-hmac**

- **ah-sha-hmac** and **esp-aes** 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, refer to 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 SAs, but will be used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the **clear crypto sa** command.

Transform Example

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
```

Configuring IPSec

This section includes the following topics:

- Ensuring That Access Lists Are Compatible with IPSec (required)
- Setting Global Lifetimes for IPSec Security Associations (required)
- Creating Crypto Access Lists (required)
- Creating Crypto Map Entries (required)
- Creating Dynamic Crypto Maps (required)
- Applying Crypto Map Sets to Interfaces (required)
- Verifying the Configuration (optional)

For IPSec configuration examples, refer to the .

See the "Configuring IPSec Network Security" of the *Cisco IOS Security Configuration Guide* for more information on configuring IPSec.

Ensuring That Access Lists Are Compatible with IPSec

IKE uses UDP port 500. The IPSec Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols use protocol numbers 50 and 51. Ensure that your interface access lists are configured so that protocol numbers 50, 51, and UDP port 500 traffic are not blocked at interfaces used by IPSec. In some cases you might need to add a statement to your access lists to explicitly permit this traffic.

Setting Global Lifetimes for IPSec Security Associations

You can change the global lifetime values which are used when negotiating new IPSec security associations. (These global lifetime values can be overridden for a particular crypto map entry).

These lifetimes only apply to security associations established via IKE. Manually established security associations do not expire.

To change a global lifetime for IPSec security associations, use one or more of the following commands in global configuration mode:

Step	Command	Purpose
Step 1	Router# **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **crypto ipsec security-association lifetime seconds** *seconds*	Changes global lifetime values used when negotiating IPSec security associations (SAs). To reset a lifetime to the default value, use the no form of this command. Specifies the number of seconds a security association will live before expiring. The default is 3600 seconds (one hour).
Step 4	Router(config)# **crypto ipsec security-association lifetime kilobytes** *kilobytes*	Changes the global "traffic-volume" lifetime for IPSec SAs. 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.
Step 5	Router(config)# **clear crypto sa** or Router(config)# **clear crypto sa peer** {*ip-address* | *peer-name*} or Router(config)# **clear crypto sa map** *map-name* or Router (config)# **clear crypto sa entry** *destination-address protocol spi*	(Optional) Clears existing security associations. This causes any existing security associations to expire immediately; future security associations will use the new lifetimes. Otherwise, any existing security associations will expire according to the previously configured lifetimes. **Note** Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

Creating Crypto Access Lists

Crypto access lists define which IP traffic will be protected by encryption. (These access lists are *not* the same as regular access lists, which determine what traffic to forward or block at an interface.) For example, access lists can be created to protect all IP traffic between Subnet A and Subnet Y or Telnet traffic between Host A and Host B.

To create crypto access lists, use the following command in global configuration mode:

Step	Command	Purpose

Step 1	`Router(config)# `**`access-list`**` access-list-number {`**`deny`**` | `**`permit`**`} protocol source source-wildcard destination destination-wildcard [`**`log`**`]`
	
	or
	
	`Router(config)# `**`ip access-list extended`**` name`
		Specifies conditions to determine which IP packets will be protected.[1] (Enable or disable crypto for traffic that matches these conditions.)
		
		We recommend that you configure "mirror image" crypto access lists for use by IPSec and that you avoid using the **any** keyword.

Step 2	Add **permit** and **deny** statements as appropriate.	Adds permit or deny statements to access lists.
Step 3	End	Exits the configuration command mode.

1. You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list.

For detailed information on configuring access lists, refer to the "Configuring IPSec Network Security" chapter in the *Security Configuration Guide* publication.

Creating Crypto Map Entries

You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPSec/IKE and IPSec/manual entries. Multiple interfaces can share the same crypto map set if you want to apply the same policy to multiple interfaces.

To create crypto map entries that use IKE to establish the security associations, use the following commands, starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`crypto map`**` map-name seq-num `**`ipsec-manual`**	Specifies the crypto map entry to create (or modify). This command puts you into the crypto map configuration mode.
Step 2	`Router(config-crypto-m)# `**`match address`**` access-list-id`	Names an IPSec access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. (The access list can specify only one **permit** entry when IKE is not used.)
Step 3	`Router(config-crypto-m)# `**`set peer`**` {hostname | ip-address}`	Specifies the remote IPSec peer. This is the peer to which IPSec protected traffic should be forwarded. (Only one peer can be specified when IKE is not used.)
Step 4	`Router(config-crypto-m)# `**`set transform-set`**` transform-set-name`	Specifies which transform set should be used. This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. (Only one transform set can be specified when IKE is not used.)

	Command	Purpose

Step 5	`Router(config-crypto-m)# ` **`set session-key inbound ah`** `spi hex-key-string`
	
	and
	
	`Router(config-crypto-m)# ` **`set session-key outbound ah`** `spi hex-key-string`
		Sets the AH Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the AH protocol.
		
		(This manually specifies the AH security association to be used with protected traffic.)

Step 6	`Router(config-crypto-m)# ` **`set session-key inbound esp`** `spi` **`cipher`** `hex-key-string` [**`authenticator`** `hex-key-string`] and `Router(config-crypto-m)# ` **`set session-key outbound esp`** `spi` **`cipher`** `hex-key-string` [**`authenticator`** `hex-key-string`]	Sets the ESP Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol. Specifies the cipher keys if the transform set includes an ESP cipher algorithm. Specifies the authenticator keys if the transform set includes an ESP authenticator algorithm. (This manually specifies the ESP security association to be used with protected traffic.)
Step 7	`Router(config-crypto-m)# ` **`exit`**	Exits crypto-map configuration mode and return to global configuration mode.

To create crypto map entries that will use IKE to establish the security associations, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ` **`crypto map`** `map-name seq-num` **`ipsec-isakmp`**	Names the crypto map entry to create (or modify). This command puts you into the crypto map configuration mode.
Step 2	`Router(config-crypto-m)# ` **`match address`** `access-list-id`	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry.
Step 3	`Router(config-crypto-m)# ` **`set peer`** {`hostname` | `ip-address`}	Specifies a remote IPSec peer. This is the peer to which IPSec protected traffic can be forwarded. Repeat for multiple remote peers.
Step 4	`Router(config-crypto-m)# ` **`set transform-set`** `transform-set-name1` [`transform-set-name2...transform-set-name6`]	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).

Step 5	`Router(config-crypto-m)# ` **`set security-association lifetime seconds`** `seconds`
	
	and
	
	`Router (config-crypto-m)# ` **`set security-association lifetime kilobytes`** `kilobytes`
		(Optional) Specifies a security association lifetime for the crypto map entry.
		
		Use this command if you want the security associations for this crypto map entry to be negotiated using different IPSec security association lifetimes than the global lifetimes.

	Command	Purpose
Step 6	Router(config-crypto-m)# **set security-association level per-host**	(Optional) Specifies that separate security associations should be established for each source/destination host pair.
		Without this command, a single IPSec "tunnel" could carry traffic for multiple source hosts and multiple destination hosts.
		With this command, when the router requests new security associations it will establish one set for traffic between Host A and Host B, and a separate set for traffic between Host A and Host C.
		Use this command with care, as multiple streams between given subnets can rapidly consume resources.
Step 7	Router(config-crypto-m)# **set pfs** [**group1** | **group2**]	(Optional) Specifies that IPSec should ask for perfect forward secrecy when requesting new security associations for this crypto map entry, or should demand perfect forward secrecy (PFS) in requests received from the IPSec peer.
Step 8	Router(config-crypto-m)# **exit**	Exits crypto-map configuration mode and return to global configuration mode.

Creating Dynamic Crypto Maps

A dynamic crypto map entry is a crypto map entry with some parameters not configured.The missing parameters are later dynamically configured (as the result of an IPSec negotiation). Dynamic crypto maps are only available for use by IKE.

Dynamic crypto map entries are grouped into sets. A set is a group of dynamic crypto map entries all with the same *dynamic-map-name*, each with a different *dynamic-seq-num*.

To create a dynamic crypto map entry, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*	Creates a dynamic crypto map entry.
Step 2	Router(config-crypto-m)# **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]	Specifies which transform sets are allowed for the crypto map entry. List multiple transform sets in order of priority (highest priority first).
		This is the only configuration statement required in dynamic crypto map entries.

	Command	Purpose
Step 3	Router(config-crypto-m)# **match address** *access-list-id*	(Optional) Accesses list number or name of an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry.
		Note Although access-lists are optional for dynamic crypto maps, they are highly recommended
		If this is configured, the data flow identity proposed by the IPSec peer must fall within a **permit** statement for this crypto access list.
		If this is not configured, the router will accept any data flow identity proposed by the IPSec peer. However, if this is configured but the specified access list does not exist or is empty, the router will drop all packets. This is similar to static crypto maps because they also require that an access list be specified.
		Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation.
Step 4	Router(config-crypto-m)# **set peer** {*hostname* | *ip-address*}	(Optional) Specifies a remote IPSec peer. Repeat for multiple remote peers.
		This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.
Step 5	Router(config-crypto-m)# **set security-association lifetime seconds** *seconds* and Router (config-crypto-m)# **set security-association lifetime kilobytes** *kilobytes*	(Optional) If you want the security associations for this crypto map to be negotiated using shorter IPSec security association lifetimes than the globally specified lifetimes, specify a key lifetime for the crypto map entry.
Step 6	Router(config-crypto-m)# **set pfs** [**group1** | **group2**]	(Optional) Specifies that IPSec should ask for perfect forward secrecy when requesting new security associations for this crypto map entry or should demand perfect forward secrecy in requests received from the IPSec peer.
Step 7	Router(config-crypto-m)# **exit**	Exits crypto-map configuration mode and return to global configuration mode.
Step 8	Repeat these steps to create additional crypto map entries as required.	

To add a dynamic crypto map set into a crypto map set, use the following command in global configuration mode:

Command	Purpose
Router(config)# **crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name*	Adds a dynamic crypto map set to a static crypto map set.

Applying Crypto Map Sets to Interfaces

Apply a crypto map set to each interface through which IPSec traffic will flow. Crypto maps instruct the router to evaluate the interface traffic against the crypto map set and use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

To apply a crypto map set to an interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **crypto map** *map-name*	Applies a crypto map set to an interface.

To specify redundant interfaces and name an identifying interface, use the following command in global configuration mode:

Command	Purpose
Router(config)# **crypto map** *map-name* **local-address** *interface-id*	Permits redundant interfaces to share the same crypto map, using the same local identity.

Configuring Compression

This section includes the following topics:

- Configure IKE Policy (required)
- Configure IKE Preshared Key (required)
- Configure ipsec transform set (required)
- Configure access-list (required)
- Configure crypto map (required)
- Apply crypto map to the Interface (required)

For IPSec configuration examples, refer to the "Configuring Compression Example".

See the "Configuring IPSec Network Security" of the *Cisco IOS Security Configuration Guide* for more information on configuring IPSec.

Configure IKE Policy

To configure IKE policy, follow the steps in "Configuring an IKE Policy" on page 3, using the commands in global configuration mode.

Configure IKE Preshared Key

To specify preshared keys at a peer, use the following commands in global configuration mode:

	Command	Purpose

Step 1	`Router (config)# crypto isakmp key keystring address peer-address`
	
	`or`
	
	`Router (config)# crypto isakmp key keystring hostname peer-hostname`
		At the local peer:
		Specify the shared key to be used with a particular remote peer.
		
		If the remote peer specified their ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.

Step 2	`Router (config)# crypto isakmp key_keystring address peer-address`
	
	`or`
	
	`Router (config)# crypto isakmp key_keystring hostname peer-hostname`
		At the remote peer:
		Specify the shared key to be used with the local peer. This is the same key you just specified at the local peer.
		
		If the local peer specified their ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.

Step 3	Repeat the previous two steps for each remote peer.	

Remember to repeat these tasks at each peer that uses preshared in an IKE policy.

Configure 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.

Command	Purpose
`Router (config)# crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]`	*transform-set-name* Specify the name of the transform set to create (or modify). transform1 transform2 transform3 Specify up to three transforms (one is required) that define the IPSec security protocol(s) and algorithm(s).

Configure access-list

To establish MAC address access lists, use the access-list global configuration command. To remove a single access list entry, use the no form of this command.

Command	Purpose
Router (config)# **access-list** access-list-number {**permit** | **deny**} address mask	*access-list-number* Specify an integer from 700 to 799 that you select for the list. **permit** Permits the frame. **deny** Denies the frame. address mask Specify 48-bit MAC addresses written in dotted triplet form. The ones bits in the mask argument are the bits to be ignored in the address value.

Configure crypto map

To create crypto map entries that use IKE to establish the security associations, use the following commands, starting in global configuration mode:

	Command	Purpose
Step 1	*Router (config)#* **crypto map** *map-name seq-num* **ipsec-isakmp**	Create the crypto map and enter crypto map configuration mode.
Step 2	Router (config)# **set peer** {*hostname* | *ip-address*}	Specify a remote IPSec peer. This is the peer to which IPSec-protected traffic can be forwarded. Repeat for multiple remote peers.
Step 3	Router (config)# **set transform-set** *transform-set-name1* *[transform-set-name2...transform-set-name6]*	Specify which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).
Step 4	Router (config)# **match address** *access-list-id*	Specify an extended access list. This access list determines which traffic is protected by IPSec and which is not.

Apply crypto map to the Interface

To apply a crypto map set to an interface, use the following commands, starting in global configuration mode:

	Command	Purpose
Step 1	*Router (config)#* **interface** *type number*	Specify an interface on which to apply the crypto map and enter interface configuration mode.
Step 2	Router (config)# **crypto map** *map-name*	Apply a crypto map set to an interface.
Step 3	Router (config)# **end**	Exit interface configuration mode.

This completes the process for configuring compression on the SA-VAM2+.

Monitoring and Maintaining IPSec

To clear (and reinitialize) IPSec security associations, use one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# clear crypto sa`
 	
or
 	
`Router(config)# clear crypto sa counters`
 	
or
 	
`Router(config)# clear crypto sa peer` {*ip-address* | *peer-name*}
 	
or
 	
`Router(config)# clear crypto sa map` *map-name*
 	
or
 	
`Router(config)# clear crypto sa entry` *destination-address protocol spi*
	Clears IPSec security associations.
	
	Note Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

To view information about your IPSec configuration, use one or more of the following commands in EXEC mode:

Command	Purpose
`Router# show crypto ipsec transform-set`	Displays your transform set configuration.
`Router# show crypto map` [**interface** *interface* | **tag** *map-name*]	Displays your crypto map configuration.
`Router# show crypto ipsec sa` [**map** *map-name* | **address** | **identity**] [**detail**]	Displays information about IPSec security associations.
`Router# show crypto dynamic-map` [**tag** *map-name*]	Displays information about dynamic crypto maps.
`Router# show crypto ipsec security-association lifetime`	Displays global security association lifetime values.

IPSec Configuration Example

The following example shows a minimal IPSec configuration where the security associations will be established via IKE. For more information about IKE, see the "Configuring Internet Key Exchange Security Protocol" chapter.

An IPSec access list defines which traffic to protect:

```
access-list 101 permit ip 10.0.0.0 0.0.0.255 10.2.2.0 0.0.0.255
```

A transform set defines how the traffic will be protected. In this example, transform set "myset1" uses DES encryption and SHA for data packet authentication:

```
crypto ipsec transform-set myset1 esp-des esp-sha
```

Another transform set example is "myset2," which uses Triple DES encryptions and MD5 (HMAC variant) for data packet authentication:

```
crypto ipsec transform-set myset2 esp-3des esp-md5-hmac
```

A crypto map joins together the IPSec access list and transform set and specifies where the protected traffic is sent (the remote IPSec peer):

```
crypto map toRemoteSite 10 ipsec-isakmp
 match address 101
 set transform-set myset2
 set peer 10.2.2.5
```

The crypto map is applied to an interface:

```
interface Serial0
 ip address 10.0.0.2
 crypto map toRemoteSite
```

Note   In this example, IKE must be enabled.

Verifying IKE and IPSec Configurations

To view information about your IPSec configurations, use **show crypto ipsec transform-set** EXEC command.

Note   If a user enters an IPSec transform that the hardware (the IPSec peer) does not support, a warning message will be displayed in the **show crypto ipsec transform-set** output.

The following sample output from the **show crypto ipsec transform-set** command displays a warning message after a user tries to configure an IPSec transform that the hardware does not support:

```
Router# show crypto ipsec transform-set
Transform set transform-1:{esp-256-aes esp-md5-hmac}
   will negotiate = {Tunnel, },

WARNING:encryption hardware does not support transform
esp-aes 256 within IPSec transform transform-1
```

To view information about your IKE configurations, use **show crypto isakmp policy** EXEC command.

Note   If a user enters an IKE encryption method that the hardware does not support, a warning message will be displayed in the **show crypto isakmp policy** output.

The following sample output from the **show crypto isakmp policy** command displays a warning message after a user tries to configure an IKE encryption method that the hardware does not support:

```
Router# show crypto isakmp policy

Protection suite of priority 1
        encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).
WARNING:encryption hardware does not support the configured
encryption method for ISAKMP policy 1
        hash algorithm:        Secure Hash Standard
        authentication method: Pre-Shared Key
```

```
        Diffie-Hellman group:  #1 (768 bit)
    lifetime:           3600 seconds, no volume limit
```

Verifying the Configuration

Some configuration changes take effect only after subsequent security associations are negotiated. For the new settings to take effect immediately, clear the existing security associations.

To clear (and reinitialize) IPSec security associations, use one of the commands in Table 4-2 in global configuration mode:

Table 4-2   Commands to Clear IP Sec Security Associations

Command	Purpose

`clear crypto sa`
or
`clear crypto sa peer {ip-address | peer-name}`
or
`clear crypto sa map map-name`
or
`clear crypto sa spi destination-address protocol spi`
	Clear IPSec security associations (SAs).
	
	Using the **clear crypto sa** command without parameters clears out the full SA database, which clears out active security sessions. You may also specify the **peer**, **map**, or **spi** keywords to clear out only a subset of the SA database.

The following steps provide information on verifying your configurations:

Step 1   Enter the **show crypto ipsec transform-set** command to view your transform set configuration:

```
Router# show crypto ipsec transform-set
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,},
```

Step 2   Enter the **show crypto map** [**interface** *interface* | **tag** *map-name]* **command** to view your crypto map configuration:

```
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,}
```

Step 3   Enter the **show crypto ipsec sa** [**map** map-name | **address** | **identity** | **detail** | **interface**] command to view information about IPSec security associations:

```
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:
```

For a detailed description of the information displayed by the **show** commands, refer to the "IP Security and Encryption" chapter of the *Security Command Reference* publication.

Configuration Examples

This section provides the following configuration examples:

Configuring IKE Policies Example

In the following example, two IKE policies are created, with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority. It also creates a preshared key to be used with policy 20 with the remote peer whose IP address is 192.168.224.33.

```
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
crypto isakmp key 1234567890 address 192.168.224.33
```

Configuring IPSec Configuration Example

The following example shows a minimal IPSec configuration where the security associations will be established via IKE:

An IPSec access list defines which traffic to protect:

```
access-list 101 permit ip 10.0.0.0 0.0.0.255 10.2.2.0 0.0.0.255
```

A transform set defines how the traffic will be protected. In this example, transform set "myset1" uses DES encryption and SHA for data packet authentication:

```
crypto ipsec transform-set myset1 esp-des esp-sha
```

Another transform set example is "myset2," which uses Triple DES encryptions and MD5 (HMAC variant) for data packet authentication:

```
crypto ipsec transform-set myset2 esp-3des esp-md5-hmac
```

A crypto map joins together the IPSec access list and transform set and specifies where the protected traffic is sent (the remote IPSec peer):

```
crypto map toRemoteSite 10 ipsec-isakmp
 match address 101
 set transform-set myset2
 set peer 10.2.2.5
```

The crypto map is applied to an interface:

```
interface Serial0
 ip address 10.0.0.2
 crypto map toRemoteSite
```

Note   In this example, IKE must be enabled.

Configuring Compression Example

The following example shows a simple configuration example for configuring compression.

To configure an IKE policy:

```
crypto isakmp policy 1
hash md5
authentication pre-share
group 2
```

To configure a IKE preshared key:

```
crypto isakmp key 12abcjhrweit345 address 16.0.0.2
```

To configure an ipsec transform set:

```
crypto ipsec transform-set proposal_01 esp-3des esp-md5-hmac comp-lzs
```

To configure an access-list:

```
access-list 101 permit ip host 16.0.0.1 host 16.0.0.2
```

To configure a crypto map:

```
crypto map MAXCASE 10 ipsec-isakmp
set peer 16.0.0.2
set transform-set proposal_01
match address 101
```

To apply crypto map to the interface:

```
interface FastEthernet1/0
crypto map MAXCASE
```

Basic IPSec Configuration Illustration

The following is an example of an IPSec configuration in which the security associations are established through IKE. In this example an access list is used to restrict the packets that are encrypted and decrypted. In this example, all packets going from IP address 10.0.0.2 to IP address 10.2.2.2 are encrypted and decrypted and all packets going from IP address 10.2.2.2 to IP address 10.0.0.2 are encrypted and decrypted. Also, one IKE policy is created.

Figure 4-1   Basic IPSec Configuration
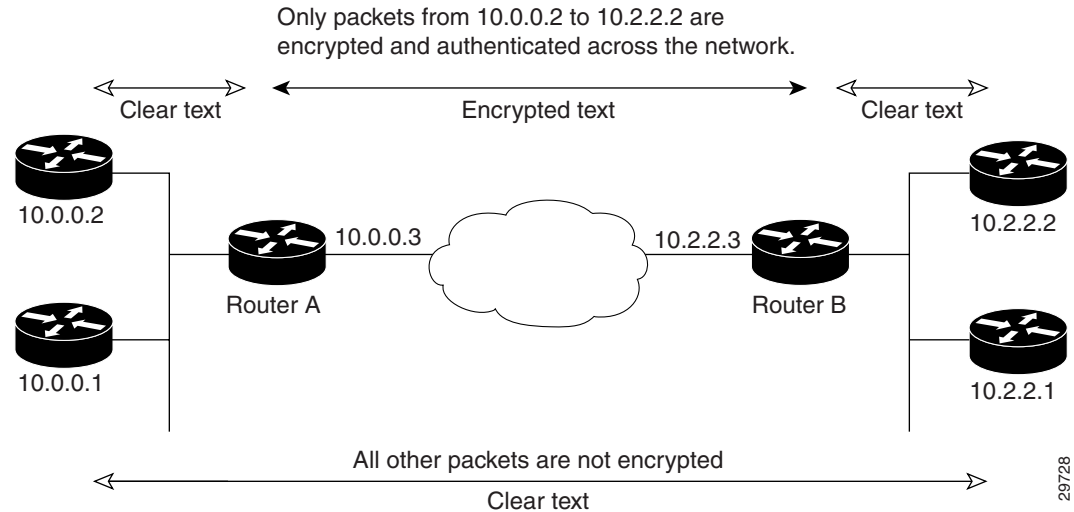
Router A Configuration

Specify the parameters to be used during an IKE negotiation:

```
crypto isakmp policy 15
  encryption des
  hash md5
  authentication pre-share
  group 2
  lifetime 5000

crypto isakmp key 1234567890 address 10.2.2.3
crypto isakmp identity address
```

Note   In the preceding example, the encryption DES of policy 15 would not appear in the written configuration because this is the default value for the encryption algorithm parameter.

A transform set defines how the traffic will be protected:

```
crypto ipsec transform-set auth1 ah-md5-hmac esp-des esp-md5-hmac
  mode tunnel
```

A crypto map joins the transform set and specifies where the protected traffic is sent (the remote IPSec peer):

```
crypto map toRemoteSite 10 ipsec-isakmp
set peer 10.2.2.3
set transform-set auth1
match address 101
```

The crypto map is applied to an interface:

```
  interface Serial0
  ip address 10.0.0.3
crypto map toRemoteSite
```

An IPSec access list defines which traffic to protect:

```
        access-list 101 permit ip host 10.0.0.2 host 10.2.2.2
        access-list 101 permit ip host 10.0.0.3 host 10.2.2.3
```

Router B Configuration

Specify the parameters to be used during an IKE negotiation:

```
crypto isakmp policy 15
  encryption des
  hash md5
  authentication pre-share
  group 2
  lifetime 5000

crypto isakmp key 1234567890 address 10.0.0.3
crypto isakmp identity address
```

A transform set defines how the traffic will be protected:

```
crypto ipsec transform-set auth1 ah-md5-hmac esp-des ah-md5-hmac
  mode tunnel
```

A crypto map joins the transform set and specifies where the protected traffic is sent (the remote IPSec peer):

```
crypto map toRemoteSite 10 ipsec-isakmp
  set peer 10.0.0.3
  set transform-set auth1
```

The crypto map is applied to an interface:

```
interface Serial0
  ip address 10.2.2.3
  crypto map toRemoteSite
```

An IPSec access list defines which traffic to protect:

```
access-list 101 permit ip host 10.2.2.2 host 10.0.0.2
access-list 101 permit ip host 10.2.2.3 host 10.0.0.3
```

Troubleshooting Tips

To verify that Cisco IOS software has recognized SA-VAM2+, enter the **show diag** command and check the output. For example, when the router has the SA-VAM2+ in slot 4, the following output appears:

```
Router#show diag 4
Slot 4:
        VAM2+ Encryption/Compression engine, Port adapter
        Port adapter is analyzed
        Port adapter insertion time 00:16:17 ago
        EEPROM contents at hardware discovery:
        Hardware Revision        :4.0
        EEPROM format version 4
        EEPROM contents (hex):
          0x00:04 FF 40 04 B0 41 04 00 FF FF FF FF FF FF FF FF
          0x10:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x20:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x30:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

```
                    0x40:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
                    0x50:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
                    0x60:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
                    0x70:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

To see if the SA-VAM2+ is currently processing crypto packets, enter the **show pas vam interface** command. The following is sample output:

```
Router# show pas vam interface
VPN Acceleration Module Version II+ in slot : 3
        Statistics for Hardware VPN Module since the last clear
        of counters 314 seconds ago
              5290894 packets in                  5290895 packets out
           1882478960 bytes in               1327439698 bytes out
                16850 paks/sec in                  16850 paks/sec out
                47940 Kbits/sec in                 33805 Kbits/sec out
              4222173 pkts compressed                  0 pkts not compressed
           1190662374 bytes before compress  405331872 bytes after compress
                2.9:1 compression ratio           2.9:1 overall
                   58 commands out                    58 commands acknowledged
             Last 5 minutes:
              4855704 packets in                  4855705 packets out
                16185 paks/sec in                  16185 paks/sec out
             46723079 bits/sec in               32921855 bits/sec out

Errors:
          ppq full errors       :       0  ppq rx errors         :       0
          cmdq full errors      :       0  cmdq rx errors        :       0
          no buffer             :       0  replay errors         :       0
          dest overflow         :       0  authentication errors :       0
          Other error           :       0  RNG self test fail    :       0
          DF Bit set            :       0  Hash Miscompare       :       0
   Unwrappable object      :       0  Missing attribute     :       0
          Invalid attrribute value:     0  Bad Attribute         :       0
          Verification Fail     :       0  Decrypt Failure       :       0
          Invalid Packet        :       0  Invalid Key           :       0
          Input Overrun         :       0  Input Underrun        :       0
          Output buffer overrun :       0  Bad handle value      :       0
          Invalid parameter     :       0  Bad function code     :       0
          Out of handles        :       0  Access denied         :       0

       Warnings:
          sessions_expired      :       0  packets_fragmented    :       0
          general               :       0  compress_bypassed     :       4

       HSP details:
          hsp_operations        :      75  hsp_sessions          :       6
```

When the SA-VAM2+ processes packets, the "packets in" and "packets out" counter changes. Counter "packets out" represents the number of packets directed to the SA-VAM2+. Counter "packets in" represents the number of packets received from the SA-VAM2+.

✎

Note   The **show pas vam interface** command output includes 'compression ratio' (or the efficiency of the tunnel bandwidth) which represents the ratio of the original packet to the compressed packet plus the ipsec headers. It does not represent the ratio of the ipsec payload before compression to the ipsec payload after compression.

This ratio may fall below 1 when small packets are not compressible, resulting in the ratio representing unencrypted packet to the encrypted packet plus the ipsec header.

To see if the IKE/IPSec packets are being redirected to the SA-VAM2+ for IKE negotiation and IPSec encryption and decryption, enter the **show crypto eli** command. The following is sample output when Cisco IOS software redirects packets to SA-VAM2+:

```
Router# show crypto eli
Hardware Encryption Layer :  ACTIVE
 Number of crypto engines = 1 .

 CryptoEngine-0 (slot-5) details.
 Capability-IPSec :IPPCP, 3DES, AES, RSA

 IKE-Session   :    0 active,  5120 max, 0 failed
 DH-Key        :    0 active,  5120 max, 0 failed
 IPSec-Session :    0 active, 10230 max, 0 failed
```

When the software crypto engine is active, the **show crypto eli** command yields no output.

During bootup or OIR, when the Cisco IOS software agrees to redirect crypto traffic to the SA-VAM2+, it prints a message similar to the following:

```
%ISA-6-INFO:Recognised crypto engine (0) at slot-1
    ...switching to hardware crypto engine
```

To disable the SA-VAM2+, use the configuration mode **no crypto engine accelerator *<slot>* command, as follows**:

```
Router(config)# no crypto engine accelerator <slot>
    Router#
    ...switching to software crypto engine
    *Oct  2 20:00:44 GMT:%VPN_HW-6-INFO:slot:4 Crypto Engine 0 in slot 4 going DOWN
    *Oct  2 20:00:44 GMT: Changing crypto engine :Service Adapter:4 state change to:
    DISABLED
    *Oct  2 20:00:44 GMT:%ISA-1-ERROR:Slot-4:VAM2+ User initiated shutdown.
```

Monitoring and Maintaining the SA-VAM2+

Use the commands that follow to monitor and maintain the SA-VAM2+:

Command	Purpose
Router# **show pas vam interface**	Verifies the SA-VAM2+ is currently processing crypto packets.
Router# **show pas vam controller**	Displays the SA-VAM2+ controller configuration.
Router# **Show version**	Displays integrated service adapter as part of the interfaces.