C7200 VSA (VPN Services Adapter)

Revised: September 12, 2006, OL-6695-02

Feature History

Release	Modification
Release 1.0	This feature was introduced on the Cisco 7204VXR and Cisco 7206VXR routers with the NPE-G2 processor.

This feature module describes the VPN Services Adapter (VSA) feature. It includes the following sections:

Feature Overview

The C7200 VSA (VPN Services Adapter) is a full-width service adapter supported in the I/O slot of the Cisco 7204VXR and Cisco 7206VXR routers with the NPE-G2 processor.

The VSA features hardware acceleration for Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES), providing increased performance for site-to-site and remote-access IPSec VPN services. The Cisco VSA supports full Layer 3 routing, quality of service (QoS), multicast and multiprotocol traffic, and broad support of integrated LAN/WAN media.

The AES is a Federal Information Processing Standard (FIPS) Publication that specifies a cryptographic algorithm for use by U.S. government organizations to protect sensitive information. AES is used on a voluntary basis by organizations, institutions, and individuals outside of the U.S. government and the United States. AEF is used by many European organizations.

The VSA provides hardware-accelerated support for multiple encryption functions:

- 128/192/256-bit AES in hardware
- DES standard mode with 56-bit key: Cipher Block Chaining (CBC)
- Performance to 900 Mbps encrypted throughput with 300-byte packets and 1000 tunnels
- 5000 tunnels for DES/3DES/AES
- Secure Hash Algorithm1 (SHA-1) and Message Digest 5 (MD5) hash algorithms
- Rivest, Shamir, Adelman (RSA) public-key algorithm
- Diffie-Hellman Groups 1, 2 and 5

Benefits

The VSA provides the following benefits:

- IPSec performance to 900 Mbps encrypted throughput
- IPSec scalability, consistent throughput from 1 to 5000 tunnels
- Avoids competition for valuable PCI bandwidth points with other interface port adapters, effectively increasing useful bandwidth to other port adapters
- Up to 50 tunnels per second
- The number of tunnels depends on the corresponding memory of the NPE (for example, currently 5000 tunnels with 1 GB of memory)
- RSA encryption
- Accelerated Crypto performance
- Accelerated Internet Key Exchange (IKE): RFCs 2401-2411 and 2451
- Support for automatic authentication using digital certificates
- Encryption services to any interface port adapter installed in the router
- LAN/WAN interface selection: Works with all Cisco 7200VXR-NPE-G2 compatible port adapters
- QoS, multiprotocol, and multicast feature interoperation
- Support for full Layer 3 routing, such as Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP) across the IPSec VPN
- VPN initialization improvements

Hardware Requirements

The hardware required to ensure proper operation of the VSA is as follows:

- The VSA is compatible with the Cisco NPE-G2 processor on the Cisco 7204VXR or Cisco 7206VXR routers.

Restrictions

The VSA has the following restrictions:

- VSA does not interoperate with other ISA or VAM/VAM2/VAM2+ crypto cards in the same router; the VAM/VAM2/VAM2+ crypto cards become disabled when the VSA is active in the Cisco 7200VXR series routers with the NPE-G2 processor.
- Only a single VSA card is supported on the Cisco 7200VXR series routers with the NPE-G2 processor.

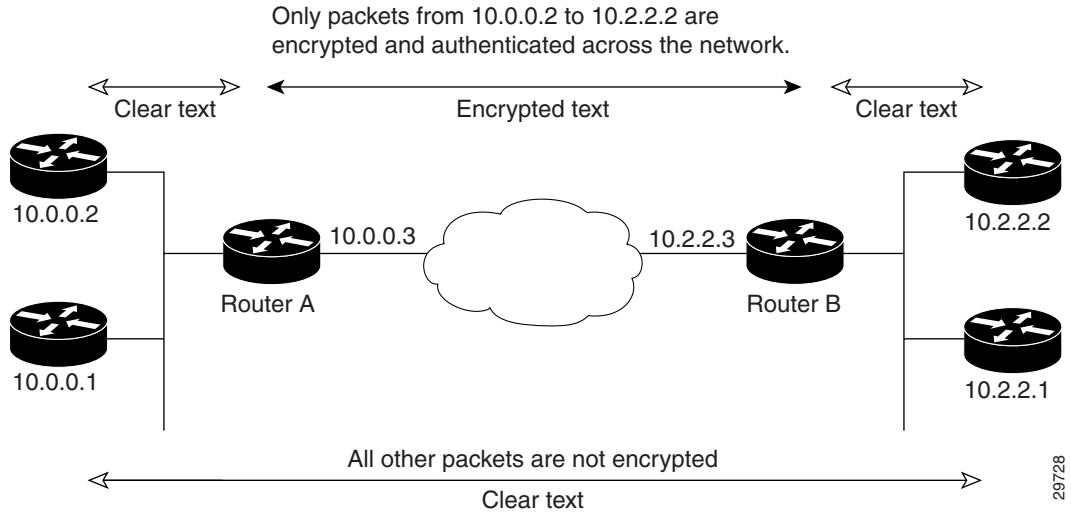
Note Only Cisco 7200VXR series routers with the NPE-G2 processor are supported.

- The VSA module does not support Online Insertion and Removal (OIR). See the *C7200 VSA (VPN Services Adapter) Installation and Configuration Guide* for details.
- Per packet count details for crypto map access list are not displayed when the **show access-list** command is entered.
- An anti-replay window size of 1024 is not supported.
- Crypto Tunnel Endpoint Discovery (TED) is unsupported with VSA.
- LLQ before crypto is not supported.
- Prefragmentation on bundled SA (transform set has both AH and ESP) is not supported.

Basic IPSec Configuration Example

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 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
```

Note In the preceding example, the mode tunnel would not appear in the written configuration because this is the default value for the transform-set.

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
  match address 101
```

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
```

Related Features and Technologies

The following features and technologies are related to the VSA:

- Internet Key Exchange (IKE)
- IP Security (IPSec)

Related Documents

- The following document describes the VSA hardware:

  C7200 VSA (VPN Services Adapter) Installation and Configuration Guide

Supported Platforms

The VSA feature runs on the following platform:

- Cisco 7204VXR and Cisco 7206VXR routers with the NPE-G2 processor

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Supported Standards, MIBs, and RFCs

Standards

- No new or modified standards are supported by this feature.

MIBs

The following MIBs were introduced or modified in this feature:

- CISCO-IPSEC-FLOW-MONITOR-MIB
- CISCO-IPSEC-MIB
- CISCO-IPSEC-POLICY-MAP-MIB

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at the following URL:

http://www.cisco.com/register

RFCs

- IPSec/IKE: RFCs 2401-2411, 2451

Prerequisites

You must configure IPSec and IKE on the router and a crypto map to all interfaces that require encryption service from the VSA. See the "Configuration Examples" section on page 27 for configuration procedures.

Configuration Tasks

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

- Using the EXEC Command Interpreter, page 7
- Configuring an IKE Policy, page 8
- Configuring a Transform Set, page 12
- Configuring IPSec, page 15

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.

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 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 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.
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. Enter 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.

	Command	Purpose
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.

Verifying IKE Configurations

To view information about your IKE configurations, enter **show crypto isakmp policy** EXEC command. The following is sample output, including 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
```

Enabling/Disabling the VSA (Optional)

This section includes the following topics:

The VSA crypto card does not support OIR. The VSA boots up only during system initialization. The VSA will not work if it is inserted after the system is up and running. The VSA can be shut down by a disabling CLI command. The VSA is ready for removal after the disabling CLI command is executed.

Disabling the VSA during Operation

Before removing the VSA, we recommend that you shut down the interface so that there is no traffic running through the VSA when it is removed. Removing the VSA while traffic is flowing through the ports can cause system disruption.

Caution You could damage the VSA, if you remove the VSA without entering the CLI command.

To disable the VSA, use the following commands, starting in global configuration mode:

	Command	Purpose
Step 7	`no crypto engine accelerator 0`	Disables the VSA.
Step 8	`crypto engine accelerator 0`	VSA will be enabled after the next system reboot. **Note** See Table 3 for more details.

Enabling/Disabling Scheme

This section describes how the VSA operates without OIR support.

Table 1 describes what occurs when the system boots up after power-on or after the reload command is entered.

Table 2 describes what occurs when the system is in run-time operation.

Table 3 describes what occurs when the **crypto engine** command is entered.

Table 1 System Boots Up After Power-on or After the reload Command is Entered

Condition	System Initialization
VSA is present	The VSA subsystem comes up and initializes automatically. Other crypto engines will be disabled.
VSA is not present	The VSA subsystem will not be initialized and system will use other crypto engine if exist.

Table 2 System is in Run-time Operation

Condition	System is Configured
Inserting the VSA	The VSA should never be inserted in run-time operation. The system should always be powered off before you insert the VSA. You need to power on the system to bring the VSA up.
CLI Enabling VSA	Not supported.
CLI Disabling VSA	**Hw-module slot 0 shutdown**—Not supported. **[no] crypto engine [slot | accelerator] 0**—See Table 3
Removing VSA	You must enter a disabling CLI (see Table 3) before removing the card to avoid damaging the hardware.

Table 3 crypto engine Command

Command	Description of VSA Behavior

`Crypto engine slot 0`
`Crypto engine [slot | accelerator] 0`
 	
Note The VSA can only be inserted in slot 0 (the I/O controller slot).
	This allows the VSA to come up and be registered as a crypto engine with the system.
	If you just performed this configuration and the VSA is currently disabled, perform a system reload or a reset to bring the VSA up.
	Note The current crypto engine will be still running, VSA will take over after the next system reboot.

`No crypto engine slot 0` `No crypto engine [slot | accelerator] 0`	These CLIs will disable the VSA. This is a configuration setting, so the VSA will remain disabled until you remove this configuration and system reloads or resets.

Configuring a Transform Set

See the *AES* feature module for more information on configuring a transform set.

This section includes the following topics:

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:

Note The **clear** commands in Step 4 below are in EXEC or enable mode (see "Using the EXEC Command Interpreter" section on page 7 for more details).

	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.
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.)

	Command	Purpose
Step 3	**end**	Exits the crypto transform configuration mode to enabled mode.
Step 4	Router# **clear crypto sa** or Router# **clear crypto sa peer** {*ip-address* | *peer-name*} or Router# **clear crypto sa map** *map-name* or Router# **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.) Entering 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 shows allowed transform combinations for the AH and ESP protocols.

Table 4 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
ESP Encryption Transform (Note: If an **ESP Authentication Transform** is used, you must pick one.)	**esp-aes**	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 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

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**

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 transform set will be used with an IPSec peer that supports the newer ESP and AH protocols.

```
crypto ipsec transform-set newer esp-3des esp-sha-hmac
```

Configuring IPSec

This section includes the following topics:

- Ensuring That Access Lists Are Compatible with IPSec, page 15 (required)
- Setting Global Lifetimes for IPSec Security Associations, page 16 (required)
- Creating Crypto Access Lists, page 16 (required)
- Creating Crypto Map Entries, page 17 (required)
- Creating Dynamic Crypto Maps, page 19 (required)
- Applying Crypto Map Sets to Interfaces, page 21 (required)
- Verifying IPSec Configurations, page 21 (optional)

For IPSec configuration examples, refer to the "Configuring IPSec Example" section on page 27.

See the "Configuring IPSec Network Security" chapter of the *Cisco IOS Security Configuration Guide* publication 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:

Note The **clear** commands in Step 3 below are in EXEC or enable mode (see "Using the EXEC Command Interpreter" section on page 7 for more details).

Step	Command	Purpose
Step 1	`Router(config)# crypto ipsec security-association lifetime seconds` *seconds*	Changes the global "timed" lifetime for IPSec SAs. This command causes the security association to time out after the specified number of seconds have passed.
Step 2	`Router(config)# crypto ipsec security-association lifetime kilobytes` *kilobytes*	Changes the global "traffic-volume" lifetime for IPSec SAs. This command causes the security association to time out after the specified amount of traffic (in kilobytes) have passed through the IPSec "tunnel" using the security association.

Step 3	`Router# clear crypto sa`
	or
	`Router# clear crypto sa peer {`*ip-address* `|` *peer-name*`}`
	or
	`Router# clear crypto sa map` *map-name*
	or
	`Router# clear crypto sa spi` *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 Entering 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 **spi** 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 *Cisco IOS 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 corresponding crypto map entry of the remote peer. (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 returns 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.
		
		Enter 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. Enter 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 (PFS) when requesting new security associations for this crypto map entry, or should demand PFS in requests received from the IPSec peer.
Step 8	`Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and returns 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** | **group5**]	(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 returns 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.

Verifying IPSec Configurations

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 5 in EXEC mode:

Table 5 Commands to Clear IPSec Security Associations

Command	Purpose

`Router# clear crypto sa`
or
`Router# clear crypto sa peer {ip-address | peer-name}`
or
`Router# clear crypto sa map map-name`
or
`Router# 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 To view your transform set configuration, enter the **show crypto ipsec transform-set** command.

The following is sample output:

```
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,},
```

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 is sample output from the **show crypto ipsec transform-set** command, which 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
```

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 To view information about IPSec security associations, enter the **show crypto ipsec sa** [**map** map-name **| address | identity | detail | interface**] 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:
```

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.

Troubleshooting Tips

To verify that Cisco IOS software has recognized the VSA, enter the **show diag** command. The following is sample output when the router has the VSA in slot 0:

```
Router# show diag 0
Slot 0:
        VSA IPsec Card Port adapter
        Port adapter is analyzed
        Port adapter insertion time 00:23:25 ago
        EEPROM contents at hardware discovery:
        PCB Serial Number        : PRTA4404055
        Product (FRU) Number     : C7200-VSA
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF C1 8B 50 52 54 41 34 34 30 34 30 35 35 40
          0x10: 05 0D CB 94 43 37 32 30 30 2D 56 53 41 20 20 20
          0x20: 20 20 20 20 20 20 20 20 D9 03 C1 40 CB 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 VSA is currently processing crypto packets, enter the **show crypto engine accelerator statistic 0** command. The following is sample output:

```
Router# show crypto engine accelerator statistic 0

Device:   VSA
Location: Service Adapter: 0
 VSA Traffic Statistics

  Inbound rate: 0pps 0kb/s  Outbound rate: 0pps 0kb/s
  TXR0 PKT: 0x00000000000028B2  Byte: 0x0000000000006ACF6  Full: 0x0000000000000000
  RXR0 PKT: 0x00000000000028B2  Byte: 0x00000000000A86398
  TXR1 PKT: 0x0000000000000000  Byte: 0x0000000000000000  Full: 0x0000000000000000
  RXR1 PKT: 0x0000000000000000  Byte: 0x0000000000000000
  TXR2 PKT: 0x0000000000000000  Byte: 0x0000000000000000  Full: 0x0000000000000000
  RXR2 PKT: 0x0000000000000000  Byte: 0x0000000000000000
  Inbound Traffic:
  Decrypted  PHY I/F:0x0000000000000000  TUNNEL I/F:    0x0000000000000000
  SPI Error  PHY I/F:0x0000000000000000  TUNNEL I/F:    0x0000000000000000
  Pass clear PHY I/F:0x0000000000000000  TUNNEL I/F:    0x0000000000000000
  SPD Drop:          0x0000000000000000  IKE Bypass:    0x0000000000000000
  Outbound Traffic:
  Encry CEF: 0x0000000000000000  FS: 0x0000000000000000  PROC: 0x0000000000000000
  Pass  CEF: 0x0000000000000000  FS: 0x0000000000000000  PROC: 0x0000000000000000
  ICMP Unreachable: 0x0000000000000000  ICMP Unreach Fail: 0x0000000000000000
  SPD Drop:  0x0000000000000000
  Special Traffic:
  VAM mode  PKT:     0x0000000000000000  Exception:     0x0000000000000000
  N2 Message:    :   0x00000000000028B2  Exception:     0x0000000000000000
  IP PKT Exception:  0x0000000000000000  DJ Overflow:   0x0000000000000000
  RAE Report PKT::   0x0000000000000000  PKT Consumed:  0x0000000000000000
  TCAM WR:           0x0000000000000001  TCAM RD:       0x0000000000000000
  SARAM WR:          0x0000000000008422  SARAM RD:      0x0000000000000000
  RAE WR:            0x0000000000080000  RAE RD:        0x0000000000000000
  Warnings:
  N2 interrupt:      0x0000000000000000  Invalid Op:    0x0000000000000000
  RX CTX error:      0x0000000000000000  TX CTX low:    0x0000000000000000
  PKT CTX Low:       0x0000000000000000  PKT Info Low:  0x0000000000000000
  PKT Header Low:    0x0000000000000000  Particle Low:  0x0000000000000000
```

```
  Missing SOP:       0x0000000000000000  Missing EOP:   0x0000000000000000
  TX Drop IB:        0x0000000000000000  TX Drop OB:    0x0000000000000000
  MSG Unknown:       0x0000000000000000  MSG too Big:   0x0000000000000000
  MSG Empty:         0x0000000000000000  MSG No Buffer: 0x0000000000000000
  PKT Info Missing:  0x0000000000000000  IB SB Error:   0x0000000000000000
  TX Drop Fastsend:  0x0000000000000000  IDMA Full:     0x0000000000000000
  Particle Fallback: 0x0000000000000000  STATISTIC:     0x0000000000000000

 Elrond statistic:
  TXDMA PKT Count:   0x00000000000028B2  Byte Count:    0x000000000006ACF6
  RXDMA PKT Count:   0x00000000000028B2  Byte Count:    0x0000000000A86398
  IPPE  PKT Count:   0x00000000000028B2  EPPE PKT Count:0x00000000000028B2
  PL3TX PKT Count:   0x00000000000028B2  Byte Count:    0x000000000009DADE
  PL3RX PKT Count:   0x00000000000028B2  Byte Count:    0x0000000000A86398
  CAM search IPPE:   0x0000000000000000  EPPE:          0x0000000000000000
  SARAM Req IPPE:    0x0000000000000000  EPPE:          0x0000000000000000
  RAE Frag Req IPPE: 0x0000000000000000  EPPE:          0x0000000000000000
  RAE ReAssembly:    0x0000000000000000  Re-Ordering:   0x0000000000000000
  REA Frag Finished: 0x0000000000000000
  Frag Drop Count:
   IPPE:             0x0000000000000000  EPPE:          0x0000000000000000
   FIFO:             0x0000000000000000  RAE:           0x0000000000000000

VSA RX Exception statistics:
   IRH Not valid           :        0   Invalid SA              :        0
   SA configuration error  :        0   Enc Dec mismatch        :        0
   Insufficient Push       :        0   Next Header mismatch    :        0
   Pad mismatch            :        0   MAC mismatch            :        0
   Atomic OP failed        :        0   L2 UDD GE 256           :        0
   Max BMI Read too small  :        0   Max BMI Read No payload :        0
   Anti replay failed      :        0   Enc Seq num overflow    :        0
   Dec IPver mismatch      :        0   Enc IPver mismatch      :        0
   TTL Decr                :        0   Selector checks         :        0
   UDP mismatch            :        0   Reserved                :        0
   Soft byte lifetime      :        0   hardbyte lifetime       :        0
   IP Parse error          :        0   Fragmentation Error     :        0
   Unknown Exception       :        0
```

When the VSA processes packets, the “packets in” and “packets out” counter changes. Counter “packets out” represents the number of packets directed to the VSA. Counter “packets in” represents the number of packets received from the VSA.

To see if the IKE/IPSec packets are being redirected to the VSA 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 VSA:

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

 CryptoEngine VSA details: state = Active
 Capability      : DES, 3DES, AES, RSA

 IKE-Session   :     0 active,  5120 max, 0 failed
 DH            :     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.

When the Cisco IOS software agrees to redirect crypto traffic to the VSA, it prints a message similar to the following:

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

To disable the VSA, use the configuration mode **no crypto engine accelerator** <*slot*> command, as follows:

```
Router(config)# no crypto engine accelerator 0
...switching to SW crypto engine
Router(config)#
*Feb  6 11:57:26.763: %VPN_HW-6-INFO_LOC: Crypto engine: slot 0  State changed to:
Disabled
*Feb  6 11:57:26.779: %PA-3-DEACTIVATED: port adapter in bay [0] powered off.
*Feb  6 11:57:26.779: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
Router(config)#end
```

Monitoring and Maintaining

Use the commands that follow to monitor and maintain the VSA:

Command	Purpose
`Router# show crypto engine accelerator statistic 0`	Verifies the VSA is currently processing crypto packets.
`Router# Show version`	Displays integrated service adapter as part of the interfaces.

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

Command	Purpose

`Router# clear crypto sa`
or
`Router# clear crypto sa peer {ip-address | peer-name}`
or
`Router# clear crypto sa map map-name`
or
`Router# clear crypto sa spi 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 **spi** 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.

Command	Purpose
`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.

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 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.

Command Reference

This section documents new commands. There are no new commands associated with the VSA. All other commands used with this feature are documented in the *Cisco IOS Release 12.3(12)M* command reference publications.

Glossary

ACL—Access Control List

AH—Authentication Header

DPD—Dead Peer Detection

ESP—Encapsulating Security Payload

GRE—Generic Routing Encapsulation

HSRP—Hot Standby Routing Protocol

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IPSec—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

ISA—Integrated Services Adapter

ISAKMP—Internet Security Association Key Management Protocol

HA—High Availability

MM—IKE Main Mode

MODECFG—Mode Configuration

QM—IKE Quick Mode

SA—security association

VAM—VPN Acceleration Module

VSA—VPN Services Adapter

VPN—Virtual Private Network

XAUTH—Extended Authentication
