Cisco VPN Services Port Adapter Configuration Guide

Release 12.2SXI

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-16406-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCSI, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco Nurse Connect, Cisco Stackpower, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0903R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco VPN Services Port Adapter Configuration Guide

CONTENTS

Preface

This preface describes the objectives and organization of this document and explains how to find additional information on related products and services. This preface contains the following sections:

Objectives

This document describes the configuration and troubleshooting of the VPN Services Port Adapter (VSPA) and its associated Services SPA Carrier-600 (SSC-600) in the Catalyst 6500 Series switch.

This document does not contain the instructions to install the VSPA in the Catalyst 6500 Series switch. For information on installing the module, refer to the *Cisco VPN Services Port Adapter Hardware Installation Guide*.

Audience

This publication is for experienced network administrators who configure and maintain VPN systems and the Catalyst 6500 Series switch.

Document Revision History

Table 1 records technical changes to this document. The table shows the Cisco IOS software release number and document revision number for the change, the date of the change, and a brief summary of the change.

Table 1 Document Revision History

Cisco IOS Release No.	Revision	Date	Change Summary
12.2(33)SXI	OL-16406-01	November 2008	First release.

Organization

This document contains the following chapters:

Chapter	Title	Description
Chapter 1	Overview of the VPN Services Port Adapter	Provides an introduction to the VSPA and the command-line interface (CLI).
Chapter 2	Overview of the IPsec Features	Provides a summary of the IPsec modes and features supported by the VSPA.
Chapter 3	Configuring VPNs in Crypto-Connect Mode	Describes the general configuration of the VSPA in crypto-connect mode.
Chapter 4	Configuring VPNs in VRF Mode	Describes the general configuration of the VSPA in VRF mode.
Chapter 5	Configuring IPsec VPN Fragmentation and MTU	Describes the general configuration of IPSec VPN fragmentation and MTU for the VSPA.
Chapter 6	Configuring Quality of Service	Describes the configuration of the Quality of Service (QoS) features of the VSPA.
Chapter 7	Configuring IKE Features	Describes the configuration of the Internet Key Exchange (IKE) features of the VSPA.
Chapter 8	Configuring Enhanced IPsec Features	Describes the configuration of enhanced security and performance features of the VSPA.
Chapter 9	Configuring PKI	Describes the configuration of the Public Key Infrastructure (PKI) features of the VSPA.
Chapter 10	Configuring Advanced VPNs	Describes the configuration of the DMVPN and EasyVPN advanced features of the VSPA.
Chapter 11	Configuring Duplicate Hardware and IPsec Failover	Describes the configuration of IPsec failover using the duplicate hardware capabilities of the VSPA.
Chapter 12	Configuring Monitoring and Accounting	Describes the configuration of the monitoring and accounting features of the VSPA.
Chapter 13	Troubleshooting	Describes techniques that you can use to troubleshoot the operation of the VSPA.

Document Conventions

This document uses the following conventions:

Note Means reader *take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Tip Means *the following information will help you solve a problem*.

Command descriptions use these conventions:

Convention	Description
boldface font	Commands and keywords are in boldface.
italic font	Arguments for which you supply values are in italics.
[ ]	Elements in square brackets are optional.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Screen examples use these conventions:

`screen font`	Terminal sessions and information that the switch displays are in screen font.
`boldface screen font`	Information that you must enter is in boldface screen font.
`italic screen font`	Arguments for which you supply values are in italic screen font.
< >	Non-printing characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or number sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation

Additional information related to the VSPA can be found in the following resources:

- For VSPA hardware installation instructions and guidelines, see the *Cisco VPN Services Port Adapter Hardware Installation Guide* at this URL:

  http://www.cisco.com/en/US/docs/interfaces_modules/services_modules/ivm/hardware/ivmhw_book.html
- For detailed information about the security configuration concepts described in this guide, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html
- For detailed information about the cryptographic commands used in this guide, see the *Cisco IOS Security Command Reference* at this URL:
- http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html
- For information about configuring the Catalyst 6500 Series switch, see the *Catalyst 6500 Series Switch Software Configuration Guide, Release 12.2SXH* at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/book.html
- For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/ffun_c.html
- To understand the Cisco IOS command-line interface and Cisco IOS command modes, see the *Cisco IOS Configuration Fundamentals Command Reference* at this URL:

  http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html

  To determine whether a feature is supported by a Cisco IOS release, or to locate the software document for that feature, or to check the minimum software requirements of Cisco IOS software with the hardware installed on your switch, Cisco maintains the Software Advisor tool on Cisco.com. You must be a registered user on Cisco.com to access this tool. To access Software Advisor, click **Login** at Cisco.com, type “Software Advisor” in the SEARCH box, and click **GO**. Click the link for the Software Advisor tool.

Note You can access Cisco IOS software configuration and hardware installation and maintenance documentation on the World Wide Web at http://www.cisco.com. Translated documentation is available at the following URL: http://www.cisco.com/public/countries_languages.shtml.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What’s New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What’s New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Notices

The following notices pertain to this software license.

OpenSSL/Open SSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

License Issues

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.
4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS”' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

Original SSLeay License:

CHAPTER 1

Overview of the VPN Services Port Adapter

This chapter provides an overview of the features of the VPN Services Port Adapter (VSPA).

This chapter includes the following sections:

Overview of the VSPA

The VPN Services Port Adapter (VSPA) is a Gigabit Ethernet IP Security (IPsec) cryptographic module that you can install in a Catalyst 6500 Series switch using the Services SPA Carrier-600 (SSC-600). The VSPA provides hardware acceleration for IPsec encryption and decryption, generic routing encapsulation (GRE), and Internet Key Exchange (IKE) key generation.

The VSPA acts as a bump-in-the-wire (BITW) in the data path to perform policy enforcement and bulk encryption and forwarding while the supervisor module performs session establishment, key management, and other features. BITW is an IPsec implementation that starts egress packet processing after the IP stack has finished with the packet and completes ingress packet processing before the IP stack receives the packet.

The VSPA can use multiple Fast Ethernet or Gigabit Ethernet ports on other Catalyst 6500 Series switch modules to connect to the Internet through WAN routers. Physical ports may be attached to the VSPA through a VLAN called the port VLAN. Packets received from the WAN routers pass through the VSPA for IPsec processing. The packets are output on a dedicated VLAN called the interface VLAN or inside VLAN. Depending on the configuration mode (VRF mode or crypto-connect mode), the interface VLAN or port VLAN may be configured explicitly or may be allocated implicitly by the system.

On the LAN side, traffic between the LAN ports can be routed or bridged on multiple Fast Ethernet or Gigabit Ethernet ports. Because the LAN traffic is not encrypted or decrypted, it does not pass through the VSPA.

The VSPA does not route, maintain routing information, or change the MAC header of a packet (except for the VLAN ID from one VLAN to another).

System Components

The cryptographic module consists of the following two components:

Description	Model Number
Services SPA Carrier-600 (SSC-600)	WS-SSC-600
VPN Services Port Adapter (VSPA)	WS-IPSEC-3

For details about the hardware installation and the physical characteristics of the VSPA and the SSC-600, see the *Cisco VPN Services Port Adapter Hardware Installation Guide*.

SSC-600

The SSC-600 inserts into a Catalyst 6500 Series switch chassis slot in the same manner as a line card and provides two subslots that are used to contain one or two VSPAs.

The SSC-600 supports online insertion and removal (OIR) with VSPAs present in the subslots. The VSPA also supports OIR and can be inserted or removed independently from the SSC-600.

VSPA

The VSPA inserts into a subslot of the SSC-600. The SSC-600 can hold one or two VSPAs.

The VSPA supports online insertion and removal (OIR). VSPAs can be inserted or removed independently from the SSC-600. The SSC-600 also supports online insertion and removal (OIR) with VSPAs inserted in its subslots.

Software Requirements

The Cisco IOS Release requirements for the VSPA are as follows:

Model	Cisco IOS Release
VSPA (WS-IPSEC-3)	12.2(33)SXI or later

In addition to the required Cisco IOS Release, you must be running one of the following crypto images on your switch:

- Supervisor Engine 720 (including 10G)

    - s72033-adventerprisek9_wan-mz
    - s72033-advipservicesk9_wan-mz
    - s72033-adventerprisek9_wan-vz
    - s72033-advipservicesk9_wan-vz
- Supervisor Engine 32 (including 10G)
    - s3223-adventerprisek9_wan-mz
    - s3223-advipservicesk9_wan-mz
    - s3223-adventerprisek9_wan-vz
    - s3223-advipservicesk9_wan-vz

Interoperability

This section lists the supervisor engines, service modules, and line cards that are compatible with the VSPA.

Table 1-1 lists the supervisor engine support for each release.

Table 1-1 Supervisor Engine Support for the VSPA by Release

		Cisco IOS Release 12.2
Supervisor	**Description**	SXI
WS-SUP720-3B	Supervisor Engine 720 Fabric MSFC3 PFC3B	Y
WS-SUP720-3BXL	Supervisor Engine 720 Fabric MSFC3 PFC3BXL	Y
VS-S720-10G-3C	Supervisor Engine 720 with 2 ports 10GbE MSFC3 PFC3C	Y
VS-S720-10G-3CXL	Supervisor Engine 720 with 2 ports 10GbE MSFC3 PFC3CXL	Y
WS-SUP32-GE-3B	Supervisor Engine 32 with 8 GbE uplinks and PFC3B	Y
WS-SUP32-10GE-3B	Supervisor Engine 32 with 2 ports 10GbE and PFC3B	Y

Table 1-2 lists the service module support for each release.

Table 1-2 Service Module Support by Release

	Cisco IOS Release 12.2
Service Module	**SXI**
Firewall Services Module (WS-SVC-FWM-1-K9)	Y

Table 1-2 Service Module Support by Release (continued)

Service Module	Cisco IOS Release 12.2
	SXI
Intrusion Detection System Module 2 (WS-SVC-IDS2BUNK9)	N
Network Analysis Module 2 (WS-SVC-NAM-2)	Y

Table 1-3 lists the SIP and SSC support for each release.

Table 1-3 SIP and SSC Support by Release

Line Card or Module	Cisco IOS Release 12.2
	SXI
7600-SIP-200	Y
7600-SIP-400	Y
7600-SIP-600	Y
7600-SSC-400	Y
WS-SSC-600	Y

Table 1-4 lists the Ethernet line card and module support for each release.

Table 1-4 Ethernet Line Card and Module Support by Release

Line Card or Module	Cisco IOS Release 12.2
	SXI
SPA-1X10GE	SIP-600
SPA-10X1GE	SIP-600
SPA-2X1GE	SIP-400
SPA-2XT3/E3	N
SPA-4X1FE-TX-V2	N
SPA-5X1GE	SIP-600
SPA-5X1GE-V2	N
SPA-8X1FE-TX-V2	N
WS-X6148-GE-TX	Y
WS-X6148-RJ-21	Y
WS-X6148-RJ-21V	Y
WS-X6148-RJ-45	Y
WS-X6148-RJ-45V	Y

Table 1-4 Ethernet Line Card and Module Support by Release (continued)

Line Card or Module	Cisco IOS Release 12.2
	SXI
WS-X6408A-GBIC	Y
WS-X6416-GBIC	Y
WS-X6502-10GE	Y
WS-X6516-GBIC	Y
WS-X6516-GE-TX	Y
WS-X6516A-GBIC	Y
WS-X6548-GE-TX	Y
WS-X6548-RJ-45	Y
WS-X6704-10GE	Y
WS-X6708-10GE	Y
WS-X6716-10GE	Y
WS-X6748-GE-TX	Y
WS-X6748-SFP	Y

Table 1-5 lists the ATM line card and module support for each release.

Table 1-5 ATM Line Card and Module Support by Release

Line Card or Module	Cisco IOS Release 12.2
	SXI
SPA-1XCHSTM1/OC3	N
SPA-1XOC48-ATM	SIP-400
SPA-2XOC3-ATM	SIP-200 SIP-400
SPA-4XOC3-ATM	N

Table 1-6 lists the POS line card and module support for each release.

Table 1-6 POS Line Card and Module Support by Release

Line Card or Module	Cisco IOS Release 12.2
	SXI
SPA-1XOC12-POS	SIP-400
SPA-1XOC48POS/RPR	N

Table 1-6 POS Line Card and Module Support by Release (continued)

Line Card or Module	Cisco IOS Release 12.2
	SXI
SPA-2XOC3-POS	SIP-200 SIP-400
SPA-OC192POS-XFP	SIP-600

Table 1-7 lists the serial line card and module support for each release.

Table 1-7 Serial Line Card and Module Support by Release

Line Card or Module	Cisco IOS Release 12.2
	SXI
SPA-2XCT3/DS0	SIP-200 SIP-400
SPA-2XT3/E3	N
SPA-4XCT3/DS0	N
SPA-4XT3/E3	N
SPA-8XCHT1/E1	N
WS-6182-2PA	N
WS-6802-2PA	N
WS-X6582-2PA With the following PAs: PA-A3-OC3MM PA-POS-OC3MM PA-POS-2OC3 PA-MC-2T3+ PA-1FE-TX PA-2FE-TX	Y

Note The VSPA does not support OSM modules.

Restrictions

The VSPA is subject to the following restrictions:

- The SSC-600 and the VSPA require Cisco IOS Release 12.2(33)SXI or a later release.
- The VSPA is supported only on the SSC-600.
- The SSC-600 supports only the VSPA. It does not support any other modules.

- You can install the VSPA in all Catalyst 6500 Series switch models, including the E and non-E switch chassis, except the Catalyst 6503.

  For more information on the Catalyst 6500 Series switch, see the *Catalyst 6500 Series Switches Installation Guide* at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/hardware/Chassis_Installation/Cat6500/6500_ins.html
- The MSFC DRAM requirements are as follows:

  – Up to 8,000 tunnels with 512-MB DRAM
  – Up to 16,000 tunnels with 1-GB DRAM

  These numbers allow for available memory for routing protocols and other applications. However, your particular use of the MSFC may demand more memory than the quantities that are listed above. In an extreme case, you could have one tunnel but still require 1-GB DRAM for other protocols and applications running on the MSFC.
- A maximum of 10 VSPAs per chassis are supported.
- VSPA state information is not maintained between the active and standby supervisor engine during normal operation. During a supervisor engine switchover in an SSO environment, the VSPA will reboot.

Supported MIBs

The following MIB is supported for the SSC-600 and the VSPA on a Catalyst 6500 Series switch:

- CISCO-IPSEC-FLOW-MONITOR-MIB

Note Gigabit Ethernet port SNMP statistics (for example, ifHCOutOctets and ifHCInOctets) are not provided for the internal VSPA trunk ports because these ports are not externally operational ports and are used only for configuration.

For more information about MIB support on a Catalyst 6500 Series switch, refer to the *Cisco 7600 Series Router MIB Specifications Guide*, at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/technical_references/7600_mib_guides/MIB_Guide_ver_6/mibgde6.html

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 this URL:

http://www.cisco.com/register

Using the Command-Line Interface

The software interface for the VSPA is the Cisco IOS command-line interface (CLI). To understand the Cisco IOS command-line interface and Cisco IOS command modes, see the *Cisco IOS Configuration Fundamentals Command Reference* at this URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html

Commands specific to the Cisco IOS software release 12.2SX are described in the *Cisco IOS Master Command List, Release 12.2SX* at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/122sxmcl/12_2sx_mcl_book.html

For detailed information on configuring the security features of the VSPA, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

For detailed information on configuring the Catalyst 6500 Series switch, see the *Catalyst 6500 Series Switch Software Configuration Guide, Release 12.2SXH* at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/book.html

Identifying Slots, Subslots, and Ports

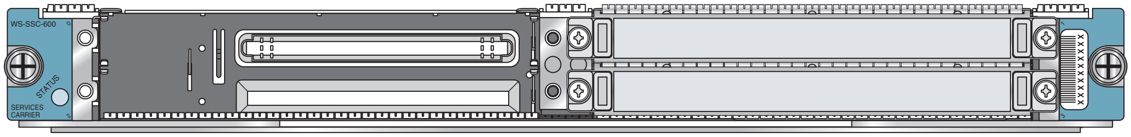
Some CLI commands, such as the **show idprom module** and **show hw-module subslot** commands, allow you to display information about the VSPA and the SSC-600. These commands require you to specify the physical location of the SSC-600 in the format *slot*, or the physical location of the VSPA in the format *slot/subslot*.

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SSC-600 is installed.
- *subslot*—Specifies the secondary slot of the SSC-600 where the VSPA is installed.

The subslot numbering is indicated by a small numeric label beside the subslot on the faceplate of the SSC-600. In the horizontal card orientation shown in Figure 1-1, the SSC-600 subslot locations are as follows:

- Subslot 0—Left subslot (top subslot if vertical)
- Subslot 1—Right subslot (bottom subslot if vertical)

Figure 1-1 SSC-600 Faceplate



For example, to display the operational status of the VSPA installed in the first subslot of the SSC-600 in slot 6 of a Catalyst 6500 Series switch, enter the following command:

```
Router# show hw-module subslot 6/0 oir
```

Some CLI commands require you to specify the inside and outside ports of the VSPA in the format *slot/subslot/port*. Although the VSPA ports are not actual Gigabit Ethernet ports, and do not share all properties of external Gigabit Ethernet interfaces, they can be addressed for configuration as Gigabit Ethernet trunk ports, using port numbers as follows:

- Port 1—Inside port, attached to interface VLAN
- Port 2—Outside port, attached to port VLAN

For example, to configure the outside port of a VSPA in the first subslot (subslot 0) of an SSC-600 in slot 6 of a Catalyst 6500 Series switch, enter the following command:

```
Router(config)# interface GigabitEthernet6/0/2
```

VSPA Hardware Configuration Guidelines

The hardware configuration guidelines for the VSPA are as follows:

- A VSPA in a chassis is active only if power to the subslot is enabled. Use the [**no**] **hw-module subslot** *slot/subslot* **shutdown** [**powered** | **unpowered**] command in global configuration mode to enable or disable power to the VSPA. The **powered** option resets power to the specified subslot, and the **unpowered** option disables power to the specified subslot. Use the [**no**] **power enable module** *slot* command to enable or disable power to the SSC-600.

- When you remove a VSPA that has some ports participating in crypto connections, the crypto configuration remains intact. When you reinsert the same type of VSPA into the same slot, the crypto connections will be reestablished. To move the VSPA to a different slot, you must first manually remove the crypto connections before removing the VSPA. You can enter the **no crypto connect vlan** command from any interface when the associated physical port is removed.
- When you reboot a VSPA that has crypto connections, the existing crypto configuration remains intact. The crypto connections will be reestablished after the VSPA reboots. When a crypto connection exists but the associated interface VLAN is missing from the VSPA inside port, the crypto connection is removed after the VSPA reboots.
- When you remove a port VLAN or an interface VLAN with the **no interface vlan** command, the associated crypto connection is also removed.

Displaying the Module Hardware Type

There are several commands on the Catalyst 6500 Series switch that provide VSPA hardware information.

- To verify the module hardware type that is installed in your switch, use the **show module** command.
- To display hardware information for the VSPA, use the **show crypto eli** command.
- To display platform and network interface controller statistics for the VSPA, use the **show crypto engine accelerator statistic** command.

For more information about these commands, see the *Cisco IOS Master Command List, Release 12.2SX* at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/122sxmcl/12_2sx_mcl_book.html

Table 1-8 shows the hardware description that appears in the **show module** command output for a VSPA on the Catalyst 6500 Series switch.

Table 1-8 Module Hardware Description in show module Command

Module	Description in show module Command
VSPA	WS-IPSEC-3

Example of the show module Command

The following example of the **show module** command reports an operational SSC-600 in slot 4 and an operational VSPA in slot 4, subslot 0:

```
Router# show module 4
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  4    0  2-subslot Services SPA Carrier-600     WS-SSC-600         JAB113100EN

Mod MAC addresses                       Hw     Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
  4  001a.a2ff.1320 to 001a.a2ff.1327   0.302 12.2(SIERRA_ 12.2(SIERRA_ Ok

Mod  Sub-Module                  Model              Serial       Hw     Status
---- --------------------------- ------------------ ----------- ------- -------
 4/0 IPSec Accelerator 3         WS-IPSEC-3         PRTA6104008  0.38   Ok
```

```
Mod  Online Diag Status
---- -------------------
  4  Pass
 4/0 Pass
```

Example of the show crypto eli Command

The following example shows output from the **show crypto eli** command on a Catalyst 6500 Series switch with a VSPAs installed in subslot 0 of an SSC-600 that is installed in slot 3. The output displays how many IKE-SAs and IPsec sessions are active and how many Diffie-Hellman keys are in use for each VSPA.

```
Router# show crypto eli

Hardware Encryption : ACTIVE
 Number of hardware crypto engines = 3

 CryptoEngine WS-IPSEC-3[3/0] details: state = Active
 Capability      :
     IPSEC: DES, 3DES, AES, RSA, IPv6

 IKE-Session   :     0 active, 16383 max, 0 failed
 DH            :     0 active,  9999 max, 0 failed
 IPSec-Session :     0 active, 65534 max, 0 failed
```

Example of the show crypto engine accelerator statistic Command

The following example shows output from the **show crypto engine accelerator statistic** command on a Catalyst 6500 Series switch with a VSPA in subslot 0 of a SSC-600 that is installed in slot 4. The output displays platform statistics for the VSPA and also displays the network interface controller statistics.

```
Router# show crypto engine accelerator statistic slot 4/0 detail

VPN module in slot 4/0:

Decryption Side Data Path Statistics
====================================
Packets RX...............: 7
Packets TX...............: 4
IPSec Transport Mode.....: 4
IPSec Tunnel Mode........: 0
AH Packets...............: 0
ESP Packets..............: 4
GRE Decapsulations.......: 0
NAT-T Decapsulations.....: 0
Clear....................: 0

Packets Drop.............: 3
Authentication Errors....: 0
Decryption Errors........: 0
Replay Check Failed......: 0
Policy Check Failed......: 0
GRE Errors...............: 0
SPD Errors...............: 0
HA Standby Drop..........: 0
Hard Life Drop...........: 0
Invalid SA...............: 0
```

```
Reassembly Frag RX.......: 0

Decryption Side Controller Statistics
=====================================
Frames RX................: 24
Bytes RX.................: 5592
Mcast/Bcast Frames RX....: 0
RX Less 128Bytes.........: 12
RX Less 512Bytes.........: 12
RX Less 1KBytes..........: 0
RX Less 9KBytes..........: 0
RX Frames Drop...........: 0

Frames TX................: 4
Bytes TX.................: 552

Encryption Side Data Path Statistics
====================================
Packets RX...............: 24
Packets TX...............: 4
IPSec Transport Mode.....: 4
IPSec Tunnel Mode........: 0
GRE Encapsulations.......: 0
NAT-T Encapsulations.....: 0
LAF prefragmented........: 0
Fragmented...............: 0
Clear....................: 0

Packets Drop.............: 20
Encryption Errors........: 0
HA Standby Drop..........: 0
Hard life Drop...........: 0
Invalid SA...............: 0
ICMP Unreachable DF set..: 0

Reassembly Frag RX.......: 0

Encryption Side Controller Statistics
=====================================
Frames RX................: 24
Bytes RX.................: 5456
Mcast/Bcast Frames RX....: 0
RX Less 128Bytes.........: 16
RX Less 512Bytes.........: 8
RX Less 1KBytes..........: 0
RX Less 9KBytes..........: 0
RX Frames Drop...........: 0

Frames TX................: 4
Bytes TX.................: 416
```

CHAPTER 2

Overview of the IPsec Features

This chapter provides an overview of the IPsec features of the VSPA.

This chapter includes the following sections:

Overview of Basic IPsec and IKE Configuration Concepts

This section reviews some basic IPsec and IKE concepts that are used throughout the configuration of the VSPA, such as security associations (SAs), access lists (ACLs), crypto maps, transform sets, and IKE policies. The information presented here is introductory and should not be considered complete.

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

Information About IPsec Configuration

IPsec provides secure tunnels between two peers, such as two routers or switches. More accurately, these tunnels are sets of security associations (SAs) that are established between two IPsec peers. The SAs define which protocols and algorithms should be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (Authentication Header (AH) or Encapsulating Security Payload (ESP)). Multiple IPsec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams might be authenticated only while other data streams must both be encrypted and authenticated.

Note The use of the term “tunnel” in this subsection does not refer to using IPsec in tunnel mode.

With IPsec, you define what traffic should be protected between two IPsec peers by configuring ACLs and applying these ACLs to interfaces by way of crypto maps. (The ACLs used for IPsec, or crypto ACLs, are used only to determine which traffic should be protected by IPsec, not which traffic should be blocked or permitted through the interface. Separate ACLs define blocking and permitting at the interface.)

If you want certain traffic to receive one combination of IPsec protection (for example, authentication only) and other traffic to receive a different combination of IPsec protection (for example, both authentication and encryption), you must create two different crypto ACLs to define the two different types of traffic. These different ACLs are then used in different crypto map entries, which specify different IPsec policies.

Crypto ACLs associated with IPsec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Indicate the data flow to be protected by the new SAs (specified by a single permit entry) when initiating negotiations for IPsec security associations.
- Process inbound traffic in order to filter out and discard traffic that should have been protected by IPsec.
- Determine whether or not to accept requests for IPsec security associations on behalf of the requested data flows when processing IKE negotiation from the IPsec peer. Negotiation is performed only for ipsec-isakmp crypto map entries. In order to be accepted, if the peer initiates the IPsec negotiation, it must specify a data flow that is “permitted” by a crypto ACL associated with an ipsec-isakmp crypto map entry.

Crypto map entries created for IPsec combine the various parts used to set up IPsec SAs, including:

- Which traffic should be protected by IPsec (per a crypto ACL)
- The granularity of the flow to be protected by a set of SAs
- Where IPsec-protected traffic should be sent (the name of the remote IPsec peer)
- The local address to be used for the IPsec traffic
- What IPsec SA should be applied to this traffic (selecting from a list of one or more transform sets)
- Whether SAs are manually established or are established via IKE
- Other parameters that might be necessary to define an IPsec SA

Crypto map entries are searched in order—the switch attempts to match the packet to the access list specified in that entry.

Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms, and other settings to apply to IPsec-protected traffic.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. During IPsec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of both peers’ IPsec SAs. (With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.)

Note To minimize the possibility of packet loss during rekeying, we recommend using time-based rather than volume-based IPsec SA expiration. By setting the lifetime volume to the maximum value using the **set security-association lifetime kilobytes 536870912** command, you can usually force time-based SA expiration.

Information About IKE Configuration

IKE is a key management protocol standard that is used in conjunction with the IPsec standard. IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard. IKE is enabled by default.

IKE is a hybrid protocol that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

You configure IKE by creating IKE policies at each peer using the **crypto isakmp policy** command. An IKE policy defines a combination of security parameters to be used during the IKE negotiation and mandates how the peers are authenticated.

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

If you do not configure any policies, your router uses the default policy, which is always set to the lowest priority, and which contains each parameter's default value.

There are five parameters to define in each IKE policy:

- Encryption algorithm
- Hash algorithm
- Authentication method
- Diffie-Hellman group identifier
- Security association lifetime

For more information about IKE, see the "Overview of IKE" section on page 7-2.

Configuring VPNs with the VSPA

To configure a VPN using the VSPA, you have two basic options: crypto-connect mode or Virtual Routing and Forwarding (VRF) mode. In either mode, you may also configure GRE tunneling to encapsulate a wide variety of protocol packet types, including multicast packets, inside the VPN tunnel.

Note Switching between crypto-connect mode and VRF mode requires a reload.

Note We recommend that you do not make changes to the VPN configuration while VPN sessions are active. To avoid system disruption, we recommend that you plan a scheduled maintenance time and clear all VPN sessions using the **clear crypto sessions** command before making VPN configuration changes.

Crypto-Connect Mode

Traditionally, VPNs are configured on the VSPA by attaching crypto maps to interface VLANs and then crypto-connecting a physical port to the interface VLAN. This method, known as crypto-connect mode, is similar to the method used to configure VPNs on routers running Cisco IOS software. When you configure VPNs on the VSPA using crypto-connect mode, you attach crypto maps to VLANs (using interface VLANs); when you configure VPNs on switches running Cisco IOS software, you configure individual interfaces.

Note With the VSPA, crypto maps are still attached to individual interfaces but the set of interfaces allowed is restricted to interface VLANs.

Crypto-connect mode VPN configuration is described in Chapter 3, "Configuring VPNs in Crypto-Connect Mode."

VRF Mode

The VRF-aware IPsec feature, known as VRF mode, allows you to map IPsec tunnels to VPN routing and forwarding instances (VRFs) using a single public-facing address. A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table. A separate set of routing and CEF tables is maintained for each VPN customer.

When you configure a VPN on the VSPA using VRF mode, the model of interface VLANs is preserved, but the **crypto connect vlan** command is not used. Instead, a route must be installed so that packets destined for that particular subnet in that particular VRF are directed to that interface VLAN.

When configuring a VPN using VRF mode, you have these additional tunneling options: tunnel protection (TP) using GRE, and Virtual Tunnel Interface (VTI). With either of these options, you can terminate tunnels in VRFs (normal VRF mode) or in the global context.

VRF mode VPN configuration is described in Chapter 4, "Configuring VPNs in VRF Mode."

Overview of the VSPA Features

The VSPA provides hardware acceleration for policy enforcement and bulk encryption and forwarding. The following features are supported:

- IPv4
  - crypto maps
  - static VTI
  - GRE/DMVPN
  - 16K tunnels
- IPv6
  - static VTI
  - IPv6-in-IPv6 (6in6) tunnels
- IKE acceleration
- AES/DES/3-DES encryption algorithms and SHA-1/MD5 hashing algorithms
- Packet classification in IPv4

IPsec Feature Support

The following tables display supported and unsupported IPsec features of the VSPA in each VPN mode according to the software release:

- IPsec Features Common To All VPN Modes, page 2-5
- IPsec Features in Crypto-Connect Mode, page 2-8
- IPsec Features in VRF Mode, page 2-9

Note This configuration guide describes VSPA features and applications that have been tested and are supported. Features and applications that do not explicitly appear in the Feature Table and in the following chapters should be considered unsupported. Contact your Cisco account team before implementing a configuration that is not described in this document.

IPsec Features Common To All VPN Modes

Table 2-1 displays supported and unsupported IPsec features common to all VPN modes.

Table 2-1 IPsec Feature Support By Release in All VPN Modes

Feature Name	Software Release 12.2
	SXI
IPsec tunnels using software-based cryptography	N
Enhanced generic router encapsulation (GRE) takeover (if supervisor engine cannot process)	Y
Multicast over GRE	Y
Multicast over multipoint GRE (mGRE) / DMVPN	N
Multicast Scalability Enhancement (single SPA mode)	N
Advanced Encryption Standard (AES)	Y
ISAKMP keyring	Y
SafeNet Client support	N
Peer filtering (SafeNet Client support)	N
Certificate to ISAKMP profile mapping	Y
Encrypted preshared key	Y
IKE Aggressive Mode Initiation	N
Call Admission Control (CAC) for IKE	Y
Dead Peer Detection (DPD) on-demand	Y
DPD periodic message option	Y
IPsec prefragmentation (Look-Ahead Fragmentation, or LAF)	Y
Reverse Route Injection (RRI)	Y
Reverse route with optional parameters	N

Table 2-1 IPsec Feature Support By Release in All VPN Modes (continued)

Feature Name	Software Release 12.2
	SXI
Adjustable IPsec anti-replay window size	Y
IPsec preferred peer	Y
Per-crypto map (and global) IPsec security association (SA) idle timers	Y
Distinguished name-based crypto maps	Y
Sequenced Access Control Lists (ACLs) (crypto ACLs)	Y
Deny policy configuration enhancements (drop, clear)	Y
Disable volume lifetime per interface	Y
VSPA quality of service (QoS) queueing	Y
Multiple RSA key pair support	Y
Protected private key storage	Y
Trustpoint CLI	Y
Query mode per trustpoint	Y
Local certificate storage location	Y
Direct HTTP enroll with CA servers	Y
Manual certificate enrollment (TFTP and cut-and-paste)	Y
Certificate autoenrollment	Y
Key rollover for Certificate Authority (CA) renewal	Y
Public-key infrastructure (PKI) query multiple servers	Y
Online Certificate Status Protocol (OCSP)	Y
Optional OCSP nonces	Y
Certificate security attribute-based access control	Y
PKI AAA authorization using entire subject name	Y
PKI local authentication using subject name	Y
Source interface selection for outgoing traffic with certificate authority	Y
Persistent self-signed certificates as Cisco IOS CA server	N
Certificate chain verification	N
Multi-tier certificate support	Y
Easy VPN Server enhanced features	N
Easy VPN Server basic features	Y
Interoperate with Easy VPN Remote using preshared key	Y
Interoperate with Easy VPN Remote using RSA signature	Y
Stateless failover using the Hot Standby Router Protocol (HSRP)	Y

Table 2-1 IPsec Feature Support By Release in All VPN Modes (continued)

Feature Name	Software Release 12.2
	SXI
Chassis-to-chassis stateful failover using HSRP and SSP in site-to-site IPsec using preshared keys with crypto maps	N
Chassis-to-chassis failover (IPsec stateful failover) with DMVPN, GRE/TP, VTI, Easy VPN, or PKI	N
Blade-to-Blade stateful failover	Y
IPsec VPN Monitoring (IPsec Flow MIB)	Y
IPsec VPN Accounting (start / stop / interim records)	Y
Crypto Conditional Debug support	Y
show crypto engine accelerator statistic command	Y
Other **show crypto engine** commands	N
clear crypto engine accelerator counter command	Y
Crypto commands applied to a loopback interface	N
Policy Based Routing (PBR) on tunnel interface or interface VLAN	N
ACL on tunnel interface	Y
MQC QoS on tunnel interface (service policy)	Y
mls qos command on all tunnel interfaces: IPsec, GRE, mGRE	N
QoS pre-classify CLI	N
NAT (GRE taken over in crypto-connect mode) on interface VLAN with crypto maps	N
16 K Tunnels (16 K IKE & IPsec tunnels)	Y
Switching between VRF and crypto-connect modes requires reboot	Y
GRE keepalives on tunnel protection (TP) tunnels	N
GRE keepalives on mGRE/DMVPN tunnels	N
IPsec Network Address Translation Transparency (NAT-T) (transport mode, ESP only)	Y
Dynamic Multipoint VPN Phase 2 (DMVPN) (mGRE; TP & NHRP)	Y
DMVPN Phase 3	N
DMVPN hub router behind a NAT gateway—tunnel mode	N
DMVPN hub router behind a NAT gateway—transport mode (not spoke-to-spoke)	Y
DMVPN spoke router behind a NAT gateway—tunnel mode	N
DMVPN spoke router behind a NAT gateway—transport mode (not spoke-to-spoke)	Y

Table 2-1 IPsec Feature Support By Release in All VPN Modes (continued)

Feature Name	Software Release 12.2
	SXI
Multicast transit traffic over DMVPN tunnels	N
Non-IP traffic over TP (DMVPN, point-to-point GRE, sVTI) tunnels	N
ip tcp adjust-mss command for IPv4	Y
Support for Supervisor Engine 2	N
Support for the VPNSM	N
All serial PPP interfaces with crypto-connect mode must have **ip unnumber null 0** command	Y
Manual key	N
Tunnel Endpoint Discovery	N
Transport adjacency and nested tunnels	N
Transit IPsec packets	Y
VSPA supported with virtual switching system (VSS)	N
IPv4 header options through IPsec tunnels	N
Invalid SPI recovery	Y
IPsec compression	N
Multilink or dialer interfaces	N
Group Encrypted Transport VPN (GETVPN)	N
IPsec Passive Mode	N

IPsec Features in Crypto-Connect Mode

Table 2-2 displays supported and unsupported IPsec features in crypto-connect mode.

Table 2-2 Features Supported or Unsupported In Crypto-Connect VPN Mode

Feature Name	Software Release 12.2
	SXI
Point-to-point GRE with tunnel protection and VTI	N
Path MTU discovery (PMTUD)	Y
PMTUD with NAT-T	N
IPsec static virtual tunnel interface (sVTI)	N
The use of VRFs in conjunction with crypto features	N
IPX and Appletalk over point-to-point GRE	Y

IPsec Features in VRF Mode

Table 2-3 displays supported and unsupported IPsec features in VRF mode.

Table 2-3 Features Supported or Unsupported In VRF Mode

Feature Name	Software Release 12.2
	SXI
Global VRF	Y
Front-door VRF (FVRF)	Y
FVRF on an mGRE tunnel configured on a DMVPN hub	Y
FVRF on an mGRE tunnel configured on a DMVPN spoke	N
Overlapping IP address space in VRFs	Y
Secondary IP addresses on interfaces	N
MPLS over GRE/IPsec (tag switching on tunnel interfaces)	N
PE-PE encryption (IPsec only) over MPLS	N
PE-PE encryption (tunnel protection) over MPLS	N
MPLS PE-CE encryption (Tag2IP) with GRE/TP	Y
MPLS PE-CE encryption (Tag2IP) with sVTI	Y
MPLS PE-CE encryption (Tag2IP) with crypto map	N
Crypto maps in VRF-lite	Y
Per-VRF AAA with RADIUS	Y
Per-VRF AAA with TACACS	N
IPsec static virtual tunnel interface (sVTI)	Y
Multicast over sVTI	Y
Ingress and egress features (ACL, QOS) on sVTI, GRE/TP, and mGRE tunnel	Y
Ingress features (ACL, PBR, inbound service policy) on the outside interface	N
Outbound service policy on the outside interface	Y
TP support in the global context	Y
IPsec SA using crypto map created in transport mode	N
Path MTU discovery (PMTUD)	Y
IPv6 IPsec sVTI IPv6-in-IPv6	Y
OSPFv3 with authentication	N
IPv4-in-IPv6, IPv6-in-IPv4	N
Multicast over IPv6 sVTI	N
AH encapsulation over IPv6 sVTI	N
ACL, QOS, NAT, VPN, MPLS, HSRPv6, **tcp adjust-mss** command on IPv6 sVTI	N

Table 2-3 Features Supported or Unsupported In VRF Mode (continued)

Feature Name	Software Release 12.2
	SXI
Certificates, PMTUD, DPD, PDPD for IPv6	N
IPv6 extension headers on encrypted packets	N
IPv6 extension headers on cleartext packets	Y

CHAPTER 3

Configuring VPNs in Crypto-Connect Mode

This chapter provides information about configuring IPsec VPNs in crypto-connect mode, one of the two VPN configuration modes supported by the VSPA. For information on the other VPN mode, Virtual Routing and Forwarding (VRF) mode, see Chapter 4, "Configuring VPNs in VRF Mode."

This chapter includes the following topics:

- Configuring Ports in Crypto-Connect Mode, page 3-2
- Configuring GRE Tunneling in Crypto-Connect Mode, page 3-21
- Configuration Examples, page 3-27

For general information on configuring IPsec VPNs with the VSPA, see the "Overview of Basic IPsec and IKE Configuration Concepts" section on page 2-1

Note The procedures in this chapter assume you have familiarity with security configuration concepts, such as VLANs, ISAKMP policies, preshared keys, transform sets, access control lists, and crypto maps. For more information about these and other security configuration concepts, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

For more information about the commands used in this chapter, see the *Cisco IOS Security Command Reference*, *Release 12.2*, at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

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

Note Some illustrations in this chapter refer to the IPsec VPN SPA. In these instances, the VSPA performs the equivalent function.

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

Configuring Ports in Crypto-Connect Mode

Before beginning your crypto-connect mode port configurations, you should read the following subsections:

- Understanding Port Types in Crypto-Connect Mode, page 3-2
- Crypto-Connect Mode Configuration Guidelines and Restrictions, page 3-5

Then perform the procedures in the following subsections:

- Configuring the VSPA Inside Port and Outside Port, page 3-6
- Configuring an Access Port, page 3-7
- Configuring a Routed Port, page 3-10
- Configuring a Trunk Port, page 3-14
- Configuring VSPA Connections to WAN Interfaces, page 3-19
- Displaying the VPN Running State, page 3-20

Note The configuration procedures in this section do not provide GRE tunneling support. For information on how to configure GRE tunneling support in crypto connect mode, see the "Configuring GRE Tunneling in Crypto-Connect Mode" section on page 3-21.

Note The procedures in this section do not provide detailed information on configuring the following Cisco IOS features: IKE policies, preshared key entries, Cisco IOS ACLs, and crypto maps. For detailed information on configuring these features, refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Understanding Port Types in Crypto-Connect Mode

To configure IPsec VPNs in crypto-connect mode, you should understand the following concepts:

- Switch Outside Ports and Inside Ports, page 3-3
- VSPA Outside Port and Inside Port, page 3-3
- Port VLAN and Interface VLAN, page 3-3
- Access Ports, Trunk Ports, and Routed Ports, page 3-4

Switch Outside Ports and Inside Ports

The Fast Ethernet or Gigabit Ethernet ports on the Catalyst 6500 Series switch that connect to the WAN routers are referred to as switch outside ports. These ports connect the LAN to the Internet or to remote sites. Cryptographic policies are applied to the switch outside ports.

The Fast Ethernet or Gigabit Ethernet ports on the Catalyst 6500 Series switch that connect to the LAN are referred to as switch inside ports.

The VSPA sends encrypted packets to the switch outside ports and decrypted packets to the Policy Feature Card (PFC) for Layer 3 forwarding to the switch inside ports.

VSPA Outside Port and Inside Port

The VSPA appears to the CLI as a module with two Gigabit Ethernet ports. The VSPA has no external connectors; the Gigabit Ethernet ports connect the VSPA to the switch backplane and Switch Fabric Module (SFM) (if installed).

One Gigabit Ethernet port handles all the traffic going to and coming from the switch outside ports. This port is referred to as the VSPA outside port. The other Gigabit Ethernet port handles all traffic going to and coming from the LAN or switch inside ports. This port is referred to as the VSPA inside port.

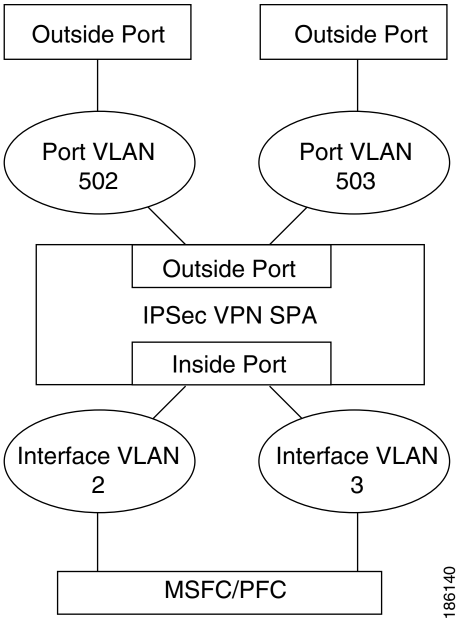
Port VLAN and Interface VLAN

Your VPN configuration can have one or more switch outside ports. To handle the packets from multiple switch outside ports, you must direct the packets from multiple switch outside ports to the VSPA outside port by placing the switch outside ports in a VLAN with the outside port of the VSPA. This VLAN is referred to as the port VLAN. The port VLAN is a Layer 2-only VLAN. You do not configure Layer 3 addresses or features on this VLAN; the packets within the port VLAN are bridged by the PFC.

Before the switch can forward the packets using the correct routing table entries, the switch needs to know which interface a packet was received on. For each port VLAN, you must create another VLAN so that the packets from every switch outside port are presented to the switch with the corresponding VLAN ID. This VLAN contains only the VSPA inside port and is referred to as the interface VLAN. The interface VLAN is a Layer 3-only VLAN. You configure the Layer 3 address and Layer 3 features, such as ACLs and the crypto map, to the interface VLAN.

You associate the port VLAN and the interface VLAN together using the **crypto engine** *slot* command on the interface VLAN followed by the **crypto connect vlan** command on the port VLAN. Figure 3-1 shows an example of the port VLAN and interface VLAN configurations.

Figure 3-1 Port VLAN and Interface VLAN Configuration Example



Port VLAN 502 and port VLAN 503 are the port VLANs that are associated with two switch outside ports.

Interface VLAN 2 and interface VLAN 3 are the interface VLANs that correspond to port VLAN 502 and port VLAN 503, respectively.

You configure the IP address, ACLs, and crypto map that apply to one switch outside port on interface VLAN 2. You configure the features that apply to another switch outside port on interface VLAN 3.

Packets coming from the WAN through the switch outside port belonging to VLAN 502 are directed by the PFC to the VSPA outside port. The VSPA decrypts the packets and changes the VLAN to interface VLAN 2, and presents the packet to the switch through the VSPA inside port. The PFC then routes the packet to the proper destination.

Packets going from the LAN to the outside ports are first routed by the PFC. Based on the route, the PFC routes the packets to one of the interface VLANs and directs the packet to the VSPA inside port. The VSPA applies the cryptographic policies that are configured on the corresponding interface VLAN, encrypts the packet, changes the VLAN ID to the corresponding port VLAN, and sends the packet to the switch outside port through the VSPA outside port.

Access Ports, Trunk Ports, and Routed Ports

When you configure VPNs on the VSPA using crypto-connect mode, you attach crypto maps to interface VLANs. Using the **crypto connect vlan** command, you then attach an interface VLAN either to a Layer 2 port VLAN associated with one or more physical ports, or directly to a physical port. The physical ports can be ATM, POS, serial, or Ethernet ports.

When youuse the **crypto connect vlan** command to attach an interface VLAN to a port VLAN that is attached to one or more Ethernet ports configured in switchport mode, the Ethernet ports can be configured as either access ports or trunk ports:

- Access ports—Access ports are switch ports that have an external or VLAN Trunk Protocol (VTP) VLAN associated with them. You can associate more than one port to a defined VLAN.

- Trunk ports—Trunk ports are switch ports that carry many external or VTP VLANs, on which all packets are encapsulated with an 802.1Q header.

When you use the **crypto connect vlan** command to attach an interface VLAN to a physical Ethernet port without defining a port VLAN, a hidden port VLAN is automatically created and associated with the port. In this configuration, the Ethernet port is a routed port:

- Routed ports—By default, every Ethernet port is a routed port until it is configured as a switch port. A routed port may or may not have an IP address assigned to it, but its configuration does not include the **switchport** command.

Crypto-Connect Mode Configuration Guidelines and Restrictions

Follow these guidelines and restrictions to prevent VSPA misconfiguration when configuring VPN ports in crypto-connect mode:

- Ethernet ports installed in a Cisco 7600 SIP-400 in the chassis cannot be configured as switch ports.
- Removing a line in a crypto ACL causes all crypto maps using that ACL to be removed and reattached to the VSPA. This action causes intermittent connectivity problems for all the security associations (SAs) derived from the crypto maps that reference that ACL.
- Do not attach a crypto map set to a loopback interface. However, you can maintain an IPsec security association database independent of physical ingress and egress interfaces with the VSPA by entering the **crypto map local-address** command.

  If you apply the same crypto map set to each secure interface and enter the **crypto map local-address** command with the interface as a loopback interface, you will have a single security association database for the set of secure interfaces. If you do not enter the **crypto map local-address** command, the number of IKE security associations is equal to the number of interfaces attached.
- You can attach the same crypto map to multiple interfaces only if the interfaces are all bound to the same crypto engine.
- If you configure a crypto map with an empty ACL (an ACL that is defined but has no lines) and attach the crypto map to an interface, all traffic goes out of the interface in the clear (unencrypted) state.
- Do not convert existing crypto-connected port characteristics. When the characteristics of a crypto-connected access port or a routed port change (switch port to routed port or vice versa), the associated crypto connection is deleted.
- Do not remove the interface VLAN or port VLAN from the VLAN database. All interface VLANs and port VLANs must be in the VLAN database. When you remove these VLANs from the VLAN database, the running traffic stops.

  When you enter the **crypto connect vlan** command and the interface VLAN or port VLAN is not in the VLAN database, this warning message is displayed:

  ```
  VLAN id 2 not found in current VLAN database. It may not function correctly unless
  VLAN 2 is added to VLAN database.
  ```
- When replacing a crypto map on an interface, always enter the **no crypto map** command before reapplying a crypto map on the interface.
- After a supervisor engine switchover, the installed modules reboot and come back online. During this period, the VSPA's established security associations (SAs) are temporarily lost and are reconstructed after the module comes back online. The reconstruction is through IKE (it is not instantaneous).

- Crypto ACLs support only the EQ operator. Other operators, such as GT, LT, and NEQ, are not supported.
- Noncontiguous subnets in a crypto ACL, as in the following example, are not supported:

```
deny ip   10.0.5.0   0.255.0.255   10.0.175.0   0.255.0.255
deny ip   10.0.5.0   0.255.0.255   10.0.176.0   0.255.0.255
```

- ACL counters are not supported for crypto ACLs.

Supported and Unsupported Features in Crypto-Connect Mode

A list of the supported and unsupported features in crypto-connect mode can be found in the "IPsec Feature Support" section on page 2-5.

Configuring the VSPA Inside Port and Outside Port

In most cases, you do not explicitly configure the VSPA inside and outside ports. Cisco IOS software configures these ports automatically.

VSPA Inside and Outside Port Configuration Guidelines and Restrictions

When configuring the VSPA inside and outside ports, follow these guidelines:

- Do not change the port characteristics of the VSPA inside or outside port unless it is necessary to set the trusted state. Cisco IOS software configures the ports automatically.

  Note Although the default trust state of the inside port is trusted, certain global settings may cause the state to change. To preserve the ToS bytes for VPN traffic in both directions, configure the **mls qos trust** command on both the inside and outside ports to set the interface to the trusted state. For information on the **mls qos trust** command, see the "Understanding QoS in the VSPA" section on page 6-1.

  If you accidentally change the inside port characteristics, enter the following commands to return the port characteristics to the defaults:

```
Router(config-if)# switchport
Router(config-if)# no switchport access vlan
Router(config-if)# switchport trunk allowed vlan 1,1002-1005
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# mtu 9216
Router(config-if)# flow control receive on
Router(config-if)# flow control send off
Router(config-if)# span portfast trunk
```

- Do not configure allowed VLANs on the inside trunk port. Cisco IOS software configures the VLAN list on the inside port automatically based on the **crypto engine slot** command. These VLANs are visible in the port configuration using the **show run** command.
- Do not configure allowed VLANs on the outside trunk port. Cisco IOS software configures these VLANs automatically as hidden VLANs. These VLANs are not visible in the port configuration using the **show run** command.

- Do not remove a VLAN from the VSPA inside port. The running traffic stops when you remove an interface VLAN from the VSPA inside port while the crypto connection to the interface VLAN exists. The crypto connection is not removed and the **crypto connect vlan** command still shows up in the **show running-config** command display. If you enter the **write memory** command with this running configuration, your startup-configuration file would be misconfigured.

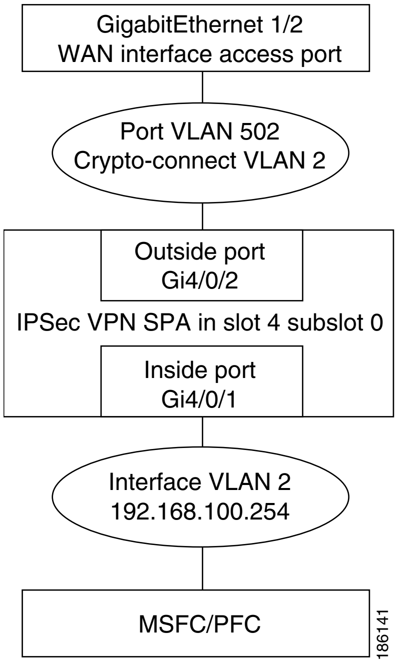
Note It is not possible to remove an interface VLAN from the VSPA inside port while the crypto connection to the interface VLAN exists. You must first remove the crypto connection.

- Do not remove a VLAN from the VSPA outside port. The running traffic stops when you remove a port VLAN from the VSPA outside port while the crypto connection to the interface VLAN exists. The crypto connection is not removed and the **crypto connect vlan** command still shows up in the **show running-config** command display. Removing a VLAN from the VSPA outside port does not affect anything in the startup-configuration file because the port VLAN is automatically added to the VSPA outside port when the **crypto connect vlan** command is entered.

Configuring an Access Port

This section describes how to configure the VSPA with an access port connection to the WAN router (see Figure 3-2).

Figure 3-2 Access Port Configuration Example



Note Ethernet ports installed in a Cisco 7600 SIP-400 in the chassis cannot be configured as switch ports.

Access Port Configuration Procedure

To configure an access port connection to the WAN router, perform the following task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp policy** *priority* ... Router(config-isakmp) # **exit**	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest. For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.
Step 2	Router(config)# **crypto isakmp key** *keystring* **address** *peer-address*	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer-address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.
Step 3	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 4	Router(config)# **access list** *access-list-number* {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Address of the host from which the packet is being sent. • *source-wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination-wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 5	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.
Step 6	Router(config)# **vlan** *inside-vlan-id*	Adds the VLAN ID into the VLAN database. • *inside-vlan-id*—VLAN identifier.
Step 7	Router(config)# **vlan** *outside-vlan-id*	Adds the VLAN ID into the VLAN database. • *outside-vlan-id*—VLAN identifier.
Step 8	Router(config)# **interface vlan** *inside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *inside-vlan-id*—VLAN identifier.
Step 9	Router(config-if)# **description inside_interface_vlan_for_crypto_map**	(Optional) Adds a comment to help identify the interface.
Step 10	Router(config-if)# **ip address** *address mask*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 11	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set. Enter the *map-name* value you created in Step 5.
Step 12	Router(config-if)# **no shutdown**	Enables the interface as a Layer 3 crypto interface VLAN.
Step 13	Router(config-if)# **crypto engine slot** *slot*/*subslot*	Assigns the crypto engine to the crypto interface VLAN. • *slot/subslot*—The slot and subslot where the VSPA is located.
Step 14	Router(config)# **interface vlan** *outside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *outside-vlan-id*—VLAN identifier.
Step 15	Router(config-if)# **description outside_access_vlan**	(Optional) Adds a comment to help identify the interface.
Step 16	Router(config-if)# **no shutdown**	Enables the interface as an outside access port VLAN.

	Command	Purpose
Step 17	`Router(config-if)# **crypto connect vlan** *inside-vlan-id*`	Connects the outside access port VLAN to the inside (crypto) interface VLAN and enters crypto-connect mode. • *inside-vlan-id*—VLAN identifier.
Step 18	`Router(config-if)# **interface gigabitethernet** *slot/subslot/port*`	Enters interface configuration mode for the secure port.
Step 19	`Router(config-if)# **description outside_secure_port**`	(Optional) Adds a comment to help identify the interface.
Step 20	`Router(config-if)# **switchport**`	Configures the interface for Layer 2 switching.
Step 21	`Router(config-if)# **switchport access vlan** *outside-vlan-id*`	Specifies the default VLAN for the interface. • *outside-vlan-id*—VLAN identifier.
Step 22	`Router(config-if)# **exit**`	Exits interface configuration mode.

For access port configuration examples, see the "Access Port in Crypto-Connect Mode Configuration Example" section on page 3-27.

Verifying the Access Port Configuration

To verify an access port configuration, enter the **show crypto vlan** command.

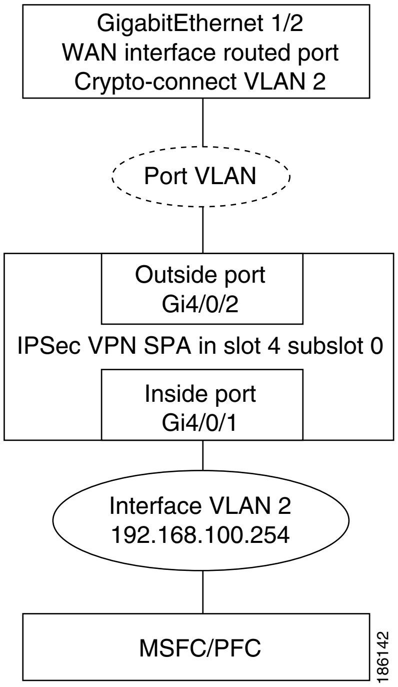
```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to VLAN 502 with crypto
map set MyMap
```

Configuring a Routed Port

This section describes how to configure the VSPA with a routed port connection to the WAN router (see Figure 3-3).

Note When a routed port without an IP address is crypto-connected to an interface VLAN, a hidden port VLAN is created automatically. This port VLAN is not explicitly configured by the user and does not appear in the running configuration.

Figure 3-3 Routed Port Configuration Example



Routed Port Configuration Guidelines

When configuring a routed port using the VSPA, follow these configuration guidelines:

- When a routed port has a crypto connection, IP ACLs cannot be attached to the routed port. Instead, you can apply IP ACLs to the attached interface VLAN.
- Unlike an access port or trunk port, the routed port does not use the **switchport** command in its configuration.

Routed Port Configuration Procedure

To configure a routed port connection to the WAN router, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp policy** *priority* ... Router(config-isakmp) # **exit**	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest. For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.
Step 2	Router(config)# **crypto isakmp key** *keystring* **address** *peer-address*	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer-address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.
Step 3	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.

	Command	Purpose
Step 4	Router(config)# **access list** *access-list-number* {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Address of the host from which the packet is being sent. • *source-wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination-wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.
Step 5	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**— Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.
Step 6	Router(config)# **vlan** *inside-vlan-id*	Adds the VLAN ID into the VLAN database. • *inside-vlan-id*—VLAN identifier.
Step 7	Router(config)# **interface vlan** *inside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *inside-vlan-id*—VLAN identifier.
Step 8	Router(config-if)# **description inside_interface_vlan_for_crypto_map**	(Optional) Adds a comment to help identify the interface.
Step 9	Router(config-if)# **ip address** *address mask*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.

	Command	Purpose
Step 10	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set. Enter the *map-name* value you created in Step 5.
Step 11	Router(config-if)# **no shutdown**	Enables the interface as a Layer 3 crypto interface VLAN.
Step 12	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the crypto engine to the crypto interface VLAN. • *slot/subslot*—The slot and subslot where the VSPA is located.
Step 13	Router(config-if)# **interface gigabitethernet** *slot/subslot/port*	Enters interface configuration mode for the secure port.
Step 14	Router(config-if)# **description outside_secure_port**	(Optional) Adds a comment to help identify the interface.
Step 15	Router(config-if)# **crypto connect vlan** *inside-vlan-id*	Connects the routed port to the crypto interface VLAN and enters crypto-connect mode. • *inside-vlan-id*—VLAN identifier.
Step 16	Router(config-if)# **exit**	Exits interface configuration mode.

For routed port configuration examples, see the "Routed Port in Crypto-Connect Mode Configuration Example" section on page 3-30.

Verifying a Routed Port Configuration

To verify a route port configuration, enter the **show crypto vlan** command. In the following example, Gi 1/2 is the crypto-connected port:

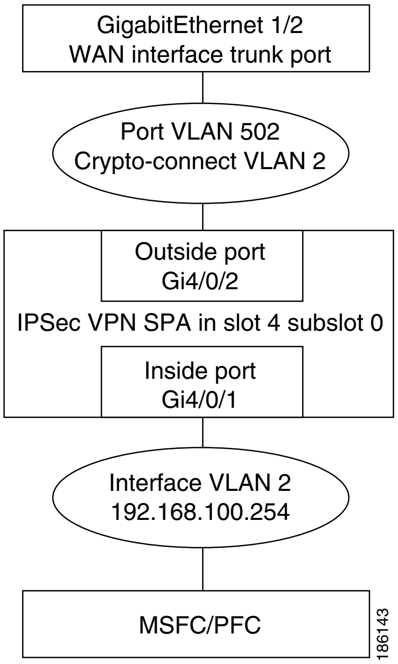
```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to Gi1/2 with crypto map
set MyMap
```

Configuring a Trunk Port

Caution When you configure an Ethernet port as a trunk port, all the VLANs are allowed on the trunk port by default. This default configuration does not work well with the VSPA and causes network loops. To avoid this problem, you must explicitly specify only the desirable VLANs.

This section describes how to configure the VSPA with a trunk port connection to the WAN router (see Figure 3-4).

Figure 3-4 Trunk Port Configuration Example



Note Ethernet ports installed in a Cisco 7600 SIP-400 in the chassis cannot be configured as switch ports.

Trunk Port Configuration Guidelines

When configuring a trunk port using the VSPA, follow these configuration guidelines:

- When you configure a trunk port for cryptographic connection, do not use the “all VLANs allowed” default. You must explicitly specify all the desirable VLANs using the **switchport trunk allowed vlan** command.
- Due to an incorrect startup configuration or through the default trunk port configuration, an interface VLAN might be associated with a trunk port. When you try to remove the interface VLAN from the VLAN list, you might receive an error message similar to the following:

```
Command rejected:VLAN 2 is crypto connected to V502.
```

To remove the interface VLAN from the VLAN list, enter the following commands:

```
Router# configure terminal
Router(config)# interface g1gabitethernet1/2
Router(config-if)# no switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1,502,1002-1005
```

Note VLANs in the VLAN list must not include any interface VLANs.

- To ensure that no interface VLANs are associated when you put an Ethernet port into the trunk mode, enter the following commands in the exact order given:

```
Router# configure terminal
Router(config)# interface g1gabitethernet1/2
Router(config)# no shut
Router(config-if)# switchport
Router(config-if)# switchport trunk allowed vlan 1
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1,502,1002-1005
```

Note VLANs in the VLAN list must not include any interface VLANs.

- A common mistake when configuring a trunk port occurs when you use the **add** option as follows:

  ```
  Router(config-if)# switchport trunk allowed vlan add 502
  ```

  If the **switchport trunk allowed vlan** command has not already been used, the **add** option does not make VLAN 502 the only allowed VLAN on the trunk port; all VLANs are still allowed after entering the command because all the VLANs are allowed by default. After you use the **switchport trunk allowed vlan** command to add a VLAN, you can then use the **switchport trunk allowed vlan add** command to add additional VLANs.
- To remove unwanted VLANs from a trunk port, use the **switchport trunk allowed vlan remove** command.

Caution Do not enter the **switchport trunk allowed vlan all** command on a secured trunk port. In addition, do not set the VSPA inside and outside ports to "all VLANs allowed."

Trunk Port Configuration Procedure

To configure a trunk port connection to the WAN switch, perform this task beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# crypto isakmp policy priority`
	`...`
	`Router(config-isakmp) # exit`
		Defines an ISAKMP policy and enters ISAKMP policy configuration mode.
		• *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Step 2	`Router(config)# crypto isakmp key keystring address peer-address`	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer-address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 3	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 4	Router(config)# **access list** *access-list-number* {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Address of the host from which the packet is being sent. • *source-wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination-wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.
Step 5	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.
Step 6	Router(config)# **vlan** *inside-vlan-id*	Adds the VLAN ID into the VLAN database. • *inside-vlan-id*—VLAN identifier.
Step 7	Router(config)# **vlan** *outside-vlan-id*	Adds the VLAN ID into the VLAN database. • *outside-vlan-id*—VLAN identifier.

	Command	Purpose
Step 8	Router(config)# **interface vlan** *inside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *inside-vlan-id*—VLAN identifier.
Step 9	Router(config-if)# **description inside_interface_vlan_for_crypto_map**	(Optional) Adds a comment to help identify the interface.
Step 10	Router(config-if)# **ip address** *address mask*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 11	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set. Enter the *map-name* value you created in Step 5.
Step 12	Router(config-if)# **no shutdown**	Enables the interface as a Layer 3 crypto interface VLAN.
Step 13	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the crypto engine to the crypto interface VLAN. • *slot/subslot*—The slot and subslot where the VSPA is located.
Step 14	Router(config)# **interface vlan** *outside-vlan-id*	Adds the specified VLAN interface as an outside trunk port VLAN and enters interface configuration mode for the specified VLAN interface. • *outside-vlan-id*—VLAN identifier.
Step 15	Router(config-if)# **description outside_trunk_port_vlan**	(Optional) Adds a comment to help identify the interface.
Step 16	Router(config-if)# **crypto connect vlan** *inside-vlan-id*	Connects the outside trunk port VLAN to the inside (crypto) interface VLAN and enters crypto connect mode. • *inside-vlan-id*—VLAN identifier.
Step 17	Router(config-if)# **no shutdown**	Enables the interface as a Layer 3 crypto interface VLAN.
Step 18	Router(config-if)# **interface gigabitethernet** *slot/subslot/port*	Enters interface configuration mode for the secure port.
Step 19	Router(config-if)# **description outside_secure_port**	(Optional) Adds a comment to help identify the interface.
Step 20	Router(config-if)# **switchport**	Configures the interface for Layer 2 switching.
Step 21	Router(config-if)# **no switchport access vlan**	Resets the access VLAN to the appropriate default VLAN for the device.
Step 22	Router(config-if)# **switchport trunk encapsulation dot1q**	Sets the trunk encapsulation to 802.1Q.
Step 23	Router(config-if)# **switchport mode trunk**	Specifies a trunk VLAN Layer 2 interface.

	Command	Purpose
Step 24	`Router(config-if)#` **`switchport trunk allowed vlan remove`** *`vlan-list`*	Removes the specified list of VLANs from those currently set to transmit from this interface. *vlan-list*—List of VLANs that transmit the interface in tagged format when in trunking mode. Valid values are from 1 to 4094.
Step 25	`Router(config-if)#` **`switchport trunk allowed vlan add`** *`outside-vlan-id`*	Adds the specified VLAN to the list of VLANs currently set to transmit from this interface. *outside-vlan-id*—VLAN identifier from step 14.
Step 26	`Router(config-if)#` **`exit`**	Exits interface configuration mode.

For trunk port configuration examples, see the "Trunk Port in Crypto-Connect Mode Configuration Example" section on page 3-32.

Verifying the Trunk Port Configuration

To verify the VLANs allowed by a trunk port, enter the **show interfaces trunk** command. The following display shows that all VLANs are allowed:

```
Router# show interfaces GigabitEthernet 1/2 trunk

Port      Mode         Encapsulation  Status        Native vlan
Gi1/2     on           802.1q         trunking      1

Port      Vlans allowed on trunk
Gi1/2     1-4094

Port      Vlans allowed and active in management domain
Gi1/2     1-4,7-8,513,1002-1005

Port      Vlans in spanning tree forwarding state and not pruned
Gi1/2     1-4,7-8,513,1002-1005
```

Configuring VSPA Connections to WAN Interfaces

The configuration of VSPA connections to WAN interfaces is similar to the configuration of Ethernet routed interfaces.

VSPA Connections to WAN Interfaces Configuration Guidelines and Restrictions

When configuring a connection to a WAN interface using a VSPA, follow these guidelines and note these restrictions:

- To configure a VSPA connection to a WAN interface, make a crypto connection from the WAN subinterface to the interface VLAN as follows:

  ```
  Router(config)# interface Vlan101
  Router(config-if)# ip address 192.168.101.1 255.255.255.0
  Router(config-if)# no mop enabled
  Router(config-if)# crypto map cwan
  Router(config-if)# crypto engine slot 4/0

  Router(config)# interface ATM6/0/0.101 point-to-point
  ```

```
Router(config-subif)# pvc 0/101
Router(config-subif)# crypto connect vlan 101
```

- You must configure a crypto connection on subinterfaces for ATM and Frame Relay.
- For ATM, there is no SVC support, no RFC-1483 bridging, and no point-to-multipoint support.
- For Frame Relay, there is no SVC support, no RFC-1490 bridging, and no point-to-multipoint support.
- For Point-to-Point Protocol (PPP) and Multilink PPP (MLPPP), you must make the physical interface passive for routing protocols, as follows:

```
Router(config)# router ospf 10
Router(config-router)# passive-interface multilink1
```

- For PPP and MLP, an **ip unnumbered Null0** command is automatically added to the port configuration to support IPCP negotiation. If you configure a **no ip address** command on the WAN port in the startup configuration, the **no ip address** command will be automatically removed in the running configuration so that it does not conflict with the automatic configuration.
- For PPP and MLP, there is no Bridging Control Protocol (BCP) support.
- When enabled on an inside VLAN, OSPF will be configured in broadcast network mode by default, even when a point-to-point interface (such as T1, POS, serial, or ATM) is crypto-connected to the inside VLAN. In addition, if OSPF is configured in point-to-point network mode on the peer router (for example, a transit router with no crypto card), OSPF will not establish full adjacency. In this case, you can manually configure OSPF network point-to-point mode in the inside VLAN:

```
Router(config)# interface vlan inside-vlan
Router(config-if)# ip ospf network point-to-point
```

For VSPA connections to WAN interfaces configuration examples, see the "VSPA Connections to WAN Interfaces Configuration Examples" section on page 3-34

Displaying the VPN Running State

Use the **show crypto vlan** command to display the VPN running state. The following examples show the **show crypto vlan** command output for a variety of VSPA configurations.

In the following example, the interface VLAN belongs to the VSPA inside port:

```
Router# show crypto vlan

  Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to Fa8/3
```

In the following example, VLAN 2 is the interface VLAN and VLAN 2022 is the hidden VLAN:

```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to VLAN 2022 with crypto
map set coral2
```

In the following example, the interface VLAN is missing on the VSPA inside port, the VSPA is removed from the chassis, or the VSPA was moved to a different subslot:

```
Router# show crypto vlan

  Interface VLAN 2 connected to VLAN 502 (no IPSec Service Module attached)
```

Configuring GRE Tunneling in Crypto-Connect Mode

In addition to choosing to configure your VPN using crypto-connect mode or VRF mode, the following additional GRE configuration options are available:

- Configuring GRE Tunneling in Crypto-Connect Mode, page 3-21
- Configuring the GRE Takeover Criteria, page 3-23
- Configuring IP Multicast over a GRE Tunnel, page 3-25

Configuring GRE Tunneling in Crypto-Connect Mode

Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to switches at remote points over an IP network.

Note The VSPA is able to accelerate packet processing for up to 2048 GRE tunnels per chassis. Any tunnels not taken over by the VSPA, or any tunnels in excess of 2048, are handled in platform hardware or by the route processor. The switch supports any number of GRE tunnels, but adding more VSPAs does not increase the 2048 tunnels per-chassis maximum that will be handled by VSPAs. If you configure more than 2048 tunnels per chassis, you could overload the route processor. Monitor the route processor CPU utilization when configuring more than 2048 tunnels per chassis.

Note If GRE encapsulation is performed by the VPN module, prefragmentation of outbound packets will be based on the IP MTU of the tunnel interface. After GRE encapsulation is performed by the VPN module, depending on the IPsec LAF (look ahead fragmentation) settings, further fragmentation may occur. The IPsec fragmentation behavior is unchanged in this release, and is based on the IPsec MTU configuration of the egress interface.

GRE Tunneling Configuration Guidelines and Restrictions

When configuring point-to-point GRE tunneling in crypto-connect mode using the VSPA, follow these guidelines:

- In a Catalyst 6500 Series switch, GRE encapsulation and decapsulation is traditionally performed by the route processor or the supervisor engine hardware. When routing indicates that encapsulated packets for a GRE tunnel will egress through an interface VLAN that is attached to a VSPA inside port, the VSPA attempts to take over the GRE tunnel interface only if the Supervisor Engine 720 is unable to process the GRE tunnel interface in hardware. If the Supervisor Engine 720 cannot process the GRE tunnel interface in hardware, the VSPA will determine if it can take over the interface. By seizing the tunnel, the VSPA takes the GRE encapsulation and decapsulation duty from the route processor. No explicit configuration changes are required to use this feature; configure GRE as you normally would. As long as routing sends the GRE-encapsulated packets over an interface VLAN, the VSPA will seize the GRE tunnel.
- If the same source address is used for more than one GRE tunnel, the supervisor engine hardware will not take over the tunnel. The VSPA will take over the tunnel if it meets the criteria discussed in the previous bullet item.

- Point-to-point GRE with tunnel protection is not supported in crypto-connect mode, but DMVPN is supported.
- If routing information changes and the GRE-encapsulated packets no longer egress through an interface VLAN, the VSPA yields the GRE tunnel. After the VSPA yields the tunnel, the route processor resumes encapsulation and decapsulation, which increases CPU utilization on the route processor.

Caution Ensure that your GRE tunnel configuration does not overload the route processor.

- A delay of up to 10 seconds occurs between routing changes and the VSPA seizing the GRE tunnel.
- The crypto map must only be applied to the interface VLAN and not to the tunnel interface.
- The following options are supported on the tunnel interface: ACLs, service policy, TTL, and ToS.
- The following options are not supported on the tunnel interface: checksum enabled, sequence check enabled, tunnel key, IP security options, policy-based routing (PBR), traffic shaping (can be applied to the crypto engine configuration within the tunnel interface configuration), QoS preclassification, and NAT.
- GRE tunneling of all non-IPv4 packets is done by the route processor even if the tunnel is seized by the VSPA.

GRE Tunneling Configuration Procedure

To configure a GRE tunnel, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`interface tunnel`** *`number`*	Creates the tunnel interface if it does not exist and enters interface configuration mode. • *number*—Number of the tunnel interface to be configured.
Step 2	`Router(config-if)# `**`ip address`** *`address`*	Sets the IP address of the tunnel interface. • *address*—IP address.
Step 3	`Router(config-if)# `**`tunnel source`** `{`*`ip-address`*` | `*`type number`*`}`	Configures the tunnel source. The source is the switch where traffic is received from the customer network. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number; for example, VLAN1.

	Command	Purpose
Step 4	`Router(config-if)# tunnel destination {hostname | ip-address}`	Sets the IP address of the destination of the tunnel interface. The destination address is the switch that transfers packets into the receiving customer network. • *hostname*—Name of the host destination. • *ip-address*—IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	`Router(config-if)# exit`	Exits interface configuration mode.

Verifying the GRE Tunneling Configuration

To verify that the VSPA has seized the GRE tunnel, enter the **show crypto vlan** command:

```
Router# show crypto vlan

Interface VLAN 101 on IPSec Service Module port 7/1/1 connected to AT4/0/0.101
    Tunnel101 is accelerated via IPSec SM in subslot 7/1
Router#
```

For complete configuration information about GRE tunneling, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_tos.html

For GRE tunneling configuration examples, see the "GRE Tunneling in Crypto-Connect Mode Configuration Example" section on page 3-38.

Configuring the GRE Takeover Criteria

You can configure the takeover criteria for Generic Routing Encapsulation (GRE) processing by using the **crypto engine gre supervisor** or **crypto engine gre vpnblade** commands. These two commands allow you to specify whether the GRE processing should be done by the supervisor engine hardware or the route processor or the VSPA.

To configure a switch to process GRE using the supervisor engine hardware or the route processor (RP), use the **crypto engine gre supervisor** command. When this command is specified, GRE processing by the supervisor engine hardware takes precedence over processing by the route processor (unless the tunnels are from duplicate sources); the RP only takes over GRE processing if the supervisor engine hardware cannot do the processing. If this command is configured, duplicate source GREs will be processed by the route processor.

To configure a switch to process GRE using the VSPA, use the **crypto engine gre vpnblade** command. If the VSPA cannot take over the GRE processing, the GRE processing will be handled either by supervisor engine hardware (which has precedence) or the route processor.

Both of these commands can be configured globally or at an individual tunnel.

Individual tunnel configuration takes precedence over the global configuration. For example, when the **crypto engine gre supervisor** command is configured at the global configuration level, the command will apply to all tunnels except those tunnels that have been configured individually using either a **crypto engine gre supervisor** command or a **crypto engine gre vpnblade** command.

At any time, only one of the two commands (**crypto engine gre supervisor** or **crypto engine gre vpnblade**) can be configured globally or individually at a tunnel. If either command is already configured, configuring the second command will overwrite the first command, and only the configuration applied by the second command will be used.

GRE Takeover Configuration Guidelines and Restrictions

When configuring GRE takeover on the VSPA, follow these guidelines and restrictions:

- For a GRE tunnel to be taken over by the VSPA, it must first satisfy the following criteria:
  - The GRE tunnel interface must be up.
  - The route to the tunnel destination must go through the VSPA.
  - The Address Resolution Protocol (ARP) entry for the next hop must exist.
  - The tunnel mode must be GRE.
  - The only supported options are **tunnel ttl** and **tunnel tos**. If any of the following command options are configured, then the tunnel will not be taken over:
    - **tunnel key**
    - **tunnel sequence-datagrams**
    - **tunnel checksum**

    All other options configured are ignored.
- If the GRE tunnels have the same source and destination addresses, then the VSPA will, at most, take over only one of them, and the determination of which specific tunnel is taken over is random.
- The VSPA will not take over GRE processing if any of the following features are configured on the tunnel interface:
  - DMVPN
  - NAT
- In crypto-connect mode, the VSPA will not take over GRE processing when the interface VLAN has no crypto map attached. The crypto map must be applied to the interface VLAN and not to the tunnel interface.
- If the VSPA cannot take over the GRE processing, the GRE processing will be handled either by the supervisor engine hardware (which has precedence) or the route processor.
- When neither the **crypto engine gre supervisor** command nor the **crypto engine gre vpnblade** command is specified globally or individually for a tunnel, the VSPA will only attempt to take over GRE processing if the following conditions apply:
  - The supervisor engine hardware does not take over GRE processing.
  - Protocol Independent Multicast (PIM) is configured on the tunnel.
  - Multiple tunnels share the same tunnel source interface and more than one tunnel is up. (If only one tunnel is up, the supervisor engine hardware can still perform the GRE processing.)
- When a new configuration file is copied to the running configuration, the new configuration will overwrite the old configuration for the **crypto engine gre vpnblade** and **crypto engine gre supervisor** commands. If the new configuration does not specify a GRE takeover criteria globally or for an individual tunnel, the existing old configuration will be used.

Configuring the GRE Takeover Criteria Globally

To configure the GRE takeover criteria globally (so that it affects all tunnels except those tunnels that have been configured individually using either a **crypto engine gre supervisor** command or a **crypto engine gre vpnblade** command), perform this task beginning in global configuration mode:

Command	Purpose

`Router(config)# crypto engine gre supervisor`
or
`Router(config)# crypto engine gre vpnblade`
	Configures a router to process GRE using the supervisor engine hardware or the route processor.
	Configures a router to process GRE using the VSPA.

Configuring the GRE Takeover Criteria at an Individual Tunnel

To configure the GRE takeover criteria at an individual tunnel (so that it affects only a specific tunnel), perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface tunnel number`	Creates the tunnel interface if it does not exist and enters interface configuration mode. • *number*—Number of the tunnel interface to be configured.

Step 2	`Router(config-if)# crypto engine gre supervisor`
	or
	`Router(config-if)# crypto engine gre vpnblade`
		Configures a router to process GRE using the supervisor engine hardware or the route processor.
		or
		Configures a router to process GRE using the VSPA.

For GRE takeover criteria configuration examples, see the "GRE Takeover Criteria Configuration Examples" section on page 3-41.

Configuring IP Multicast over a GRE Tunnel

IP multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to multiple recipients. GRE is a tunneling protocol developed by Cisco and commonly used with IPsec that encapsulates a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP network.

In some network scenarios, you might want to configure your network to use GRE tunnels to send Protocol Independent Multicast (PIM) and multicast traffic between routers. Typically, this occurs when the multicast source and receiver are separated by an IP cloud that is not configured for IP multicast routing. In such network scenarios, configuring a tunnel across an IP cloud with PIM-enabled transports multicast packets toward the receiver. The configuration of IP multicast over a GRE tunnel using the VSPA involves these key steps:

- Configuring multicast globally
- Configuring PIM at the tunnel interfaces

IP Multicast over a GRE Tunnel Configuration Guidelines and Restrictions

When configuring IP multicast over a GRE tunnel, follow these guidelines:

- If PIM is configured, and the GRE tunnel interface satisfies the rest of the tunnel takeover criteria, the GRE processing of the multicast packets will be taken over by the VSPA.
- GRE processing of IP multicast packets will be taken over by the VSPA if the GRE tunnel interface satisfies the following tunnel takeover criteria:
  - The tunnel is up.
  - There are no other tunnels with the same source destination pair.
  - The tunnel is not a multipoint GRE (mGRE) tunnel.
  - PIM is configured on the tunnel.
  - None of the following features are configured on the tunnel: tunnel key, tunnel sequence-datagrams, tunnel checksum, tunnel udlr address-resolution, tunnel udlr receive-only, tunnel udlr send-only, ip proxy-mobile tunnel reverse, or NAT. If any of these options are specified, the VSPA will not seize the GRE tunnel.
- When a tunnel is configured for multicast traffic, the **crypto engine gre supervisor** command should not be applied to the tunnel.

Configuring IP Multicast Globally

You must enable IP multicast routing globally before you can enable PIM on the router interfaces.

To enable IP multicast routing globally, use the **ip multicast-routing** command.

Configuring PIM at the Tunnel Interfaces

You must enable PIM on all participating router interfaces before IP multicast will function.

To enable PIM, use the **ip pim** command as follows:

```
Router(config-if)# ip pim {dense-mode | sparse-mode | sparse-dense-mode}
```

dense-mode enables dense mode of operation.

sparse-mode enables sparse mode of operation.

sparse-dense-mode enables the interface in either sparse mode or dense mode of operation, depending on which mode the multicast group operates in.

For IP multicast over GRE tunnels configuration examples, see the "IP Multicast over a GRE Tunnel Configuration Example" section on page 3-42.

Verifying the IP Multicast over a GRE Tunnel Configuration

To verify the IP multicast over a GRE tunnel configuration, enter the **show crypto vlan** and **show ip mroute** commands.

To verify that the tunnel has been taken over by the VSPA, enter the **show crypto vlan** command:

```
Router# show crypto vlan

Interface VLAN 100 on IPSec Service Module port Gi7/0/1 connected to Po1 with crypto map
set map_t3
Tunnel15 is accelerated via IPSec SM in subslot 7/0
```

To verify that the IP multicast traffic is hardware-switched, enter the **show ip mroute** command and look for the **H** flag:

```
Router# show ip mroute 230.1.1.5

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 230.1.1.5), 01:23:45/00:03:16, RP 15.15.1.1, flags: SJC
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16
(120.1.0.3, 230.1.1.5), 01:23:46/00:03:25, flags: T
Incoming interface: GigabitEthernet8/1, RPF nbr 0.0.0.0, RPF-MFD
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16, H
```

For IP multicast over GRE tunnels configuration examples, see the "IP Multicast over a GRE Tunnel Configuration Example" section on page 3-42.

Configuration Examples

This section provides examples of the following configurations:

- Access Port in Crypto-Connect Mode Configuration Example, page 3-27
- Routed Port in Crypto-Connect Mode Configuration Example, page 3-30
- Trunk Port in Crypto-Connect Mode Configuration Example, page 3-32
- VSPA Connections to WAN Interfaces Configuration Examples, page 3-34
- GRE Tunneling in Crypto-Connect Mode Configuration Example, page 3-38
- GRE Takeover Criteria Configuration Examples, page 3-41
- IP Multicast over a GRE Tunnel Configuration Example, page 3-42

Access Port in Crypto-Connect Mode Configuration Example

This section provides an example of the access port configuration with switch 1 shown in Figure 3-2 on page 3-7:

Switch 1 (Access Port)

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk

interface Vlan2
 !interface vlan
 ip address 11.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 !port vlan
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

Switch 2 (Access Port)

```
!
hostname router-2
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 !port vlan
 no ip address
 crypto connect vlan 2
!
```

```
ip classless
ip route 13.0.0.0 255.0.0.0 11.0.0.2
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
end
```

Routed Port in Crypto-Connect Mode Configuration Example

This section provides an example of the routed port configuration with switch 1 shown in Figure 3-3 on page 3-11:

Switch 1 (Routed Port)

```
!
hostname router-1
!
vlan 2
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 no ip address
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.1 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.0 255.0.0.0 11.0.0.2
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
end
```

Switch 2 (Routed Port)

```
!
hostname router-2
!
vlan 2
!
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 no ip address
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.2 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

Trunk Port in Crypto-Connect Mode Configuration Example

This section provides an example of the trunk port configuration with switch 1 shown in Figure 3-4 on page 3-15:

Switch 1 (Trunk Port)

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1  esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 502
 switchport mode trunk
!
interface GigabitEthernet4/0/1
```

```
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan 502
 !port vlan
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 13.0.0.0 255.0.0.0 11.0.0.2
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
end
```

Switch 2 (Trunk Port)

```
!
hostname router-2
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1  esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 13.0.0.1 255.255.255.0
```

```
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 502
 switchport mode trunk
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk

interface Vlan2
 !interface vlan
 ip address 11.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 !port vlan
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

VSPA Connections to WAN Interfaces Configuration Examples

The following are configuration examples of VSPA connections to WAN interfaces:

- VSPA Connection to an ATM Port Adapter Configuration Example, page 3-35
- VSPA Connection to a POS Port Adapter Configuration Example, page 3-36
- VSPA Connection to a Serial Port Adapter Configuration Example, page 3-37

VSPA Connection to an ATM Port Adapter Configuration Example

The following example shows the configuration of a VSPA connection to an ATM port adapter:

```
!
hostname router-1
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
crypto map testtag_1 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal
 match address acl_1
!
interface GigabitEthernet1/1
 ip address 12.0.0.2 255.255.255.0
!
interface ATM2/0/0
 no ip address
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/0/0.1 point-to-point
 atm pvc 20 0 20 aal5snap
 no atm enable-ilmi-trap
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag_1
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.1 255.255.255.255 11.0.0.2
!
ip access-list extended acl_1
 permit ip host 12.0.0.1 host 13.0.0.1
!
```

VSPA Connection to a POS Port Adapter Configuration Example

The following example shows the configuration of a VSPA connection to a POS port adapter:

```
!

hostname router-1
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
crypto map testtag_1 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal
 match address acl_1
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.2 255.255.255.0
!
interface POS2/0/0
 no ip address
 encapsulation frame-relay
 clock source internal
!
interface POS2/0/0.1 point-to-point
 frame-relay interface-dlci 16
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
```

```
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag_1
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.1 255.255.255.255 11.0.0.2
!
ip access-list extended acl_1
 permit ip host 12.0.0.1 host 13.0.0.1
```

VSPA Connection to a Serial Port Adapter Configuration Example

The following example shows the configuration of a VSPA connection to a serial port adapter:

```
!
hostname router-1
!
controller T3 2/1/0
 t1 1 channel-group 0 timeslots 1
 t1 2 channel-group 0 timeslots 1
 t1 3 channel-group 0 timeslots 1
 t1 4 channel-group 0 timeslots 1
 t1 5 channel-group 0 timeslots 1
 t1 6 channel-group 0 timeslots 1
 t1 7 channel-group 0 timeslots 1
 t1 8 channel-group 0 timeslots 1
 t1 9 channel-group 0 timeslots 1
 t1 10 channel-group 0 timeslots 1
 t1 11 channel-group 0 timeslots 1
 t1 12 channel-group 0 timeslots 1
 t1 13 channel-group 0 timeslots 1
 t1 14 channel-group 0 timeslots 1
 t1 15 channel-group 0 timeslots 1
 t1 16 channel-group 0 timeslots 1
 t1 17 channel-group 0 timeslots 1
 t1 18 channel-group 0 timeslots 1
 t1 19 channel-group 0 timeslots 1
 t1 20 channel-group 0 timeslots 1
 t1 21 channel-group 0 timeslots 1
 t1 22 channel-group 0 timeslots 1
 t1 23 channel-group 0 timeslots 1
 t1 24 channel-group 0 timeslots 1
 t1 25 channel-group 0 timeslots 1
 t1 26 channel-group 0 timeslots 1
 t1 27 channel-group 0 timeslots 1
 t1 28 channel-group 0 timeslots 1
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
crypto map testtag_1 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal
 match address acl_1
```

```
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.2 255.255.255.0
!
interface Serial2/1/0/1:0
 ip unnumbered Null0
 encapsulation ppp
 no fair-queue
 no cdp enable
 crypto connect vlan 2
!
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag_1
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.1 255.255.255.255 11.0.0.2
!
ip access-list extended acl_1
 permit ip host 12.0.0.1 host 13.0.0.1
```

GRE Tunneling in Crypto-Connect Mode Configuration Example

This section provides an example of GRE tunneling configurations.

Switch 1 (GRE Tunneling)

The following example shows the configuration of GRE tunneling for switch 1:

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1 ah-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
!
!
!
interface Tunnel1
 ip address 1.0.0.1 255.255.255.0
 tunnel source Vlan2
 tunnel destination 11.0.0.2
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
!
ip classless
ip route 13.0.0.0 255.0.0.0 Tunnel1
```

```
!
!
access-list 101 permit gre host 11.0.0.1 host 11.0.0.2
!
```

Switch 2 (GRE Tunneling)

```
!
hostname router-2
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1 ah-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
!
!
interface Tunnel1
 ip address 1.0.0.2 255.255.255.0
 tunnel source Vlan2
 tunnel destination 11.0.0.1
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
```

```
!
interface Vlan2
 ip address 11.0.0.2 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 Tunnel1
!
access-list 101 permit gre host 11.0.0.2 host 11.0.0.1
!
```

GRE Takeover Criteria Configuration Examples

The following examples show how to configure the GRE takeover criteria:

- GRE Takeover Criteria Global Configuration Example, page 3-41
- GRE Takeover Criteria Tunnel Configuration Example, page 3-41
- GRE Takeover Verification Example, page 3-41

GRE Takeover Criteria Global Configuration Example

The following example shows that the GRE takeover criteria has been set globally and the supervisor engine hardware or RP always does the GRE processing:

```
Router(config)# crypto engine gre supervisor
```

GRE Takeover Criteria Tunnel Configuration Example

The following example shows that the GRE takeover criteria has been set individually for tunnel interface 3 and the VSPA always does the GRE processing for this tunnel:

```
Router(config)# interface tunnel 3
Router(config-if)# crypto engine gre vpnblade
```

GRE Takeover Verification Example

The following example shows how to verify that the tunnel has been taken over by the VSPA:

```
Router(config)# show crypto vlan 100

Interface VLAN 100 on IPSec Service Module port GigabitEthernet4/0/1 connected to POS8/0/0
with crypto map set MAP_TO_R2
    Tunnel1 is accelerated via IPSec SM in subslot 4/0
```

The following example shows that the tunnel has not been taken over by the VSPA:

```
Router(config)# show crypto vlan 100

Interface VLAN 100 on IPSec Service Module port GigabitEthernet4/0/1 connected to POS8/0/0
with crypto map set MAP_TO_R2
```

IP Multicast over a GRE Tunnel Configuration Example

The following example shows how to configure IP multicast over GRE:

```
hostname router-1
!
vlan 2-1001
ip multicast-routing
!
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set proposal esp-3des
!
!
crypto map cm_spoke1_1 10 ipsec-isakmp
 set peer 11.1.1.1
 set transform-set proposal
 match address spoke1_acl_1
!
!
interface Tunnel1
 ip address 20.1.1.1 255.255.255.0
 ip mtu 9216
 ip pim sparse-mode
 ip hold-time eigrp 1 3600
 tunnel source 1.0.1.1
 tunnel destination 11.1.1.1
 crypto engine slot 4/0
!
interface GigabitEthernet1/1
 !switch inside port
 mtu 9216
 ip address 50.1.1.1 255.0.0.0
 ip pim sparse-mode
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,252,1002-1005
 switchport mode trunk
 mtu 9216
!
interface GigabitEthernet4/0/1
 !IPSec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPSec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,252,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 mtu 9216
 ip address 1.0.1.1 255.255.255.0
 crypto map cm_spoke1_1
 crypto engine slot 4/0
!
interface Vlan252
 mtu 9216
 no ip address
 crypto connect vlan 2
!
router eigrp 1
 network 20.1.1.0 0.0.0.255
 network 50.1.1.0 0.0.0.255
 no auto-summary
 no eigrp log-neighbor-changes
!
ip classless
ip route 11.1.1.0 255.255.255.0 1.0.1.2
!
ip pim bidir-enable
ip pim rp-address 50.1.1.1
!
ip access-list extended spoke1_acl_1
 permit gre host 1.0.1.1 host 11.1.1.1
!
```

CHAPTER 4

Configuring VPNs in VRF Mode

This chapter provides information about configuring IPsec VPNs in Virtual Routing and Forwarding (VRF) mode, one of the two VPN configuration modes supported by the VSPA. For information on the other VPN mode, crypto-connect mode, see Chapter 3, "Configuring VPNs in Crypto-Connect Mode."

This chapter includes the following topics:

For general information on configuring IPsec VPNs with the VSPA, see the "Overview of Basic IPsec and IKE Configuration Concepts" section on page 2-1

Note The procedures in this chapter assume you have familiarity with security configuration concepts, such as VLANs, ISAKMP policies, preshared keys, transform sets, access control lists, and crypto maps. For detailed information on configuring these features, refer to the following Cisco IOS documentation:

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

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

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* and the related Cisco IOS Release 12.2 software configuration guide and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xvi.

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

Configuring VPNs in VRF Mode

The VRF-Aware IPsec feature, known as VRF mode, allows you to map IPsec tunnels to VPN routing and forwarding instances (VRFs) using a single public-facing address.

A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table. A separate set of routing and CEF tables is maintained for each VPN customer.

Each IPsec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to one VRF domain, called the front door VRF (FVRF), while the inner, protected IP packet belongs to another domain called the inside VRF (IVRF). Stated another way, the local endpoint of the IPsec tunnel belongs to the FVRF while the source and destination addresses of the inside packet belong to the IVRF.

One or more IPsec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the ISAKMP profile that is attached to a crypto map entry.

With VRF mode, packets belonging to a specific VRF are routed through the VSPA for IPsec processing. Through the CLI, you associate a VRF with an interface VLAN that has been configured to point to the VSPA. An interface VLAN must be created for each VRF. Packets traveling from an MPLS cloud to the Internet that are received from an IVRF are routed to an interface VLAN, and then to the VSPA for IPsec processing. The VSPA modifies the packets so that they are placed on a special Layer 3 VLAN for routing to the WAN-side port after they leave the VSPA.

Note IVRFs are the VRFs on the unprotected (LAN) side.

Packets traveling in the inbound direction from a protected port on which the **crypto engine slot** command has been entered are redirected by a special ACL to the VSPA, where they are processed according to the Security Parameter Index (SPI) contained in the packet's IPsec header. Processing on the VSPA ensures that the decapsulated packet is mapped to the appropriate interface VLAN corresponding to the inside VRF. This interface VLAN has been associated with a specific VRF, so packets are routed within the VRF to the correct inside interface.

Note Tunnel protection is supported in VRF mode. For information on configuring tunnel protection, see the "Configuring VPNs in VRF Mode with Tunnel Protection (GRE)" section on page 4-10 and the "VRF Mode Tunnel Protection Configuration Example" section on page 4-30.

The following subsections describe how to configure a VPN in VRF mode with and without tunnel protection on the VSPA:

- Understanding VPN Configuration in VRF Mode, page 4-3
- VRF Mode Configuration Guidelines and Restrictions, page 4-4
- Configuring VPNs in VRF Mode without Tunnel Protection, page 4-5
- Configuring VPNs in VRF Mode with Tunnel Protection (GRE), page 4-10

Note For additional information on configuring VPNs in VRF mode, refer to the Cisco IOS documentation at this URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_vrf_aware_ipsec_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Understanding VPN Configuration in VRF Mode

In the traditional crypto-connect mode, a VPN is configured by attaching crypto maps to interface VLANs and then crypto-connecting a physical port to the interface VLAN. When configuring a VPN in VRF mode using the VSPA, the model of interface VLANs is preserved, but the **crypto connect vlan** CLI command is not used. When a packet comes into an interface on a specific VRF, the packet must get to the proper interface VLAN. A route must be installed so that packets destined for that particular subnet in that particular VRF are directed to that interface VLAN. This function can be achieved through the following configuration options:

- Configuring an IP address on the interface VLAN that is in the same subnet as the packets' destination IP address. For example, packets are trying to reach subnet 10.1.1.x and their destination IP address is 10.1.1.1 as follows:

```
int vlan 100
 ip vrf forwarding coke
 ip address 10.1.1.254  255.255.255.0 <-- same subnet as 10.1.1.x that we are trying
to reach.
 crypto map mymap
 crypto engine slot 4/1
```

- Configuring a static route as follows:

```
ip route vrf coke 10.1.1.0 255.255.255.0 vlan 100
```

- Configuring routing protocols. You configure BGP, OSPF, or other routing protocols so that remote switches broadcast their routes.

Note Do not configure routing protocols unless you are using tunnel protection.

- Configuring Reverse Route Injection (RRI). You configure RRI so that a route gets installed when the remote end initiates an IPsec session (as in remote access situations).

With VRF mode, the switch sees the interface VLAN as a point-to-point connection; the packets are placed directly onto the interface VLAN. Each VRF has its own interface VLAN.

When a crypto map is attached to an interface VLAN and the **ip vrf forwarding** command has associated that VLAN with a particular VRF, the software creates a point-to-point connection so that all routes pointing to the interface VLAN do not attempt to run the Address Resolution Protocol (ARP). Through normal routing within the VRF, packets to be processed by the VSPA are sent to the interface VLAN. You may configure features on the interface VLAN. The IP address of the interface VLAN must be on the same subnet as the desired destination subnet for packets to be properly routed.

When you enter the **ip vrf forwarding** command on an inside interface, all packets coming in on that interface are routed correctly within that VRF.

When you enable the **crypto engine mode vrf** command and enter the **crypto engine slot outside** command on an interface, a special ACL is installed that forces all incoming Encapsulating Security Payload (ESP)/Authentication Header (AH) IPsec packets addressed to a system IP address to be sent to the VSPA WAN-side port. NAT Traversal (NAT-T) packets are also directed to the VSPA by the special ACL.

Note You must enter the **vrf** *vrf_name* command from within the context of an ISAKMP profile. This command does not apply to the VRF-aware crypto infrastructure; it applies only to generic crypto processing. When the ISAKMP profile is added to a crypto map set, the VRF becomes the default VRF for all of the crypto maps in the list. Individual crypto maps may override this default VRF by specifying another policy profile that contains a different VRF. If no profile is applied to a crypto map tag, it inherits the VRF from the interface if you have configured the interface with the **ip vrf forwarding** command.

All packets destined for a protected outside interface received in this VRF context are placed on the associated interface VLAN. Similarly, all decapsulated ingress packets associated with this VRF are placed on the appropriate interface VLAN so that they may be routed in the proper VRF context.

VRF Mode Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring a VPN for the VSPA using VRF mode:

Note After enabling or disabling VRF mode using the [**no**] **crypto engine mode vrf** command, you must reload the supervisor engine. In addition, MPLS tunnel recirculation must be enabled for VRF mode. That is, you must add the **mls mpls tunnel-recir** command before entering the **crypto engine mode vrf** command.

- The procedure for configuring a VPN in VRF mode varies based on whether you are using tunnel protection or not.
- Unlike VSPA crypto-connect mode configurations, when configuring VPNs in VRF mode, you do not use the **crypto connect vlan** command.
- The **ip vrf forwarding** command is not required when configuring GRE with tunnel protection.
- Crypto ACLs support only the EQ operator. Other operators, such as GT, LT, and NEQ, are not supported.
- Noncontiguous subnets in a crypto ACL, as in the following example, are not supported:

```
deny ip   10.0.5.0   0.255.0.255   10.0.175.0   0.255.0.255
deny ip   10.0.5.0   0.255.0.255   10.0.176.0   0.255.0.255
```

- ACL counters are not supported for crypto ACLs.
- When you create an ISAKMP profile, note the following guidelines regarding the use of the **vrf** command:
  - You must use the **vrf** command if you are using the ISAKMP profile with a crypto map.
  - You are not required to use the **vrf** command if you are using the ISAKMP profile with tunnel protection.
  - You should not use the **vrf** command if you are using the ISAKMP profile with DMVPN.
- ISAKMP profiles with different VRFs are not supported in the same crypto map. All ISAKMP profiles in a crypto map must belong to the same VRF.
- When the **ip vrf forwarding** command is applied to a VLAN, any previously existing IP address assigned to that VLAN is removed. To assign an IP address to the VLAN, enter the **ip address** command after the **ip vrf forwarding** command, not preceding it.
- In VRF mode, there is no configuration difference between multiple VSPA operation and single VSPA operation. For multiple VSPA operation, the only change is to the output of the **show crypto vlan** command. The following is an example:

```
Interface Tu1 on IPSec Service Module port Gi7/1/1 connected to VRF vrf1
Interface VLAN 2 on IPSec Service Module port Gi7/1/1 connected to VRF vrf2
```

- The number of outside interfaces supported by the VSPA is determined by your system resources.

- In VRF mode, crypto map interfaces that share the same local address must be bound to the same crypto engine.
- When two tunnels share the same tunnel source address, they will be taken over by the VSPA only if one of the following two conditions are met:
  - Both tunnels share the same FVRF.
  - The **crypto engine gre vpnblade** command is entered.
- You can configure the FVRF to be the same as the IVRF.
- In VRF mode, ingress ACLs are installed on crypto engine outside interfaces. In combination with other configured ACLs, these ACLs may cause the ACL-TCAM usage to become excessive. To reduce the TCAM usage, share the TCAM resources by entering the **mls acl tcam share-global** command in the configuration. You can view the ACL usage using the **show tcam counts** command.

Supported and Unsupported Features in VRF Mode

A list of the supported and unsupported features in VRF mode can be found in the "IPsec Feature Support" section on page 2-5. Additional details are as follows:

- Remote access into a VRF (provider edge [PE]) is supported with the following:
  - Reverse Route Injection (RRI) only with crypto maps
  - Proxy AAA (one VRF is proxied to a dedicated AAA)
- Customer edge-provider edge (CE-PE) encryption using tunnel protection is supported with the following:
  - Routing update propagation between CEs
  - IGP/eBGP routing update propagation between the PE and CEs

Configuring VPNs in VRF Mode without Tunnel Protection

To configure a VPN in VRF mode with crypto maps and without tunnel protection, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mls mpls tunnel-recir`	Enables tunnel-MPLS recirculation.
Step 2	`Router(config)# crypto engine mode vrf`	Enables VRF mode for the VSPA. **Note** After enabling or disabling VRF mode using the **crypto engine mode vrf** command, you must reload the supervisor engine.
Step 3	`Router(config)# ip vrf vrf-name`	Configures a VRF routing table and enters VRF configuration mode. • *vrf-name*—Name assigned to the VRF.

	Command	Purpose
Step 4	`Router(config-vrf)#` **`rd`** *`route-distinguisher`*	Creates routing and forwarding tables for a VRF. • *route-distinguisher*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1).
Step 5	`Router(config-vrf)#` **`route-target export`** *`route-target-ext-community`*	Creates lists of export route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 3.
Step 6	`Router(config-vrf)#` **`route-target import`** *`route-target-ext-community`*	Creates lists of import route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 3.
Step 7	`Router(config-vrf)#` **`exit`**	Exits VRF configuration mode.
Step 8	`Router(config)#` **`crypto keyring`** *`keyring-name`* `[`**`vrf`** *`fvrf-name`*`]`	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring. • *fvrf-name*—(Optional) Front door virtual routing and forwarding (FVRF) name to which the keyring will be referenced. fvrf-name must match the FVRF name that was defined during virtual routing and forwarding (VRF) configuration
Step 9	`Router(config-keyring)#` **`pre-shared-key`** `{`**`address`** *`address`* `[`*`mask`*`] |` **`hostname`** *`hostname`*`}` **`key`** *`key`*	Defines a preshared key to be used for IKE authentication. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • *hostname*—Fully qualified domain name of the peer. • *key*—Specifies the secret key.
Step 10	`Router(config-keyring)#` **`exit`**	Exits keyring configuration mode.

	Command	Purpose
Step 11	`Router(config)# crypto ipsec transform-set transform-set-name transform1[transform2[transform3]]`	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. Accepted values are described in the *Cisco IOS Security Command Reference*.
Step 12	`Router(config-crypto-trans)# exit`	Exits crypto transform configuration mode.
Step 13	`Router(config)# crypto isakmp policy priority`	Defines an IKE policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 14	`Router(config-isakmp)# authentication pre-share`	Specifies the authentication method with an IKE policy. • **pre-share**—Specifies preshared keys as the authentication method.
Step 15	`Router(config-isakmp)# lifetime seconds`	Specifies the lifetime of an IKE SA. • *seconds*—Number of seconds each SA should exist before expiring. Use an integer from 60 to 86,400 seconds. Default is 86,400 (one day).
Step 16	`Router(config-isakmp)# exit`	Exits ISAKMP policy configuration mode.
Step 17	`Router(config)# crypto isakmp profile profile-name`	Defines an ISAKMP profile and enters ISAKMP profile configuration mode. • *profile-name*—Name of the user profile.
Step 18	`Router(config-isa-prof)# vrf ivrf`	Defines the VRF to which the IPsec tunnel will be mapped. • *ivrf*—Name of the VRF to which the IPsec tunnel will be mapped. Enter the same value specified in Step 2.
Step 19	`Router(config-isa-prof)# keyring keyring-name`	Configures a keyring within an ISAKMP profile. • *keyring-name*—Keyring name. This name must match the keyring name that was defined in global configuration. Enter the value specified in Step 7.

	Command	Purpose
Step 20	Router(config-isa-prof)# **match identity address** *address* [*mask*] [*vrf*]	Matches an identity from a peer in an ISAKMP profile. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • [*vrf*]—(Optional) This argument is only required when configuring a front door VRF (FVRF). This argument specifies that the address is an FVRF instance.
Step 21	Router(config-isa-prof)# **exit**	Exits ISAKMP profile configuration mode.
Step 22	Router(config)# **access list** *access-list-number* {**deny** | **permit**} **ip host** *source* **host** *destination*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Number of the host from which the packet is being sent. • *destination*—Number of the host to which the packet is being sent.
Step 23	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations.
Step 24	Router(config-crypto-map)# **set peer** {*hostname* | *ip-address*}	Specifies an IPsec peer in a crypto map entry. • {*hostname* | i*p-address*}—IPsec peer host name or IP address. Enter the value specified in Step 19.
Step 25	Router(config-crypto-map)# **set transform-set** *transform-set-name*	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set. Enter the value specified in Step 10.
Step 26	Router(config-crypto-map)# **set isakmp-profile** *profile-name*	Sets the ISAKMP profile name. • *profile-name*—Name of the ISAKMP profile. Enter the value entered in Step 16.

	Command	Purpose
Step 27	Router(config-crypto-map)# **match address** [*access-list-id* | *name*]	Specifies an extended access list for the crypto map entry. • *access-list-id*—Identifies the extended access list by its name or number. Enter the value specified in Step 21. • *name*—(Optional) Identifies the named encryption access list. This name should match the name argument of the named encryption access list being matched.
Step 28	Router(config-crypto-map)# **exit**	Exits crypto map configuration mode.
Step 29	Router(config)# **crypto map** *map-name* **local-address** *interface-id*	Specifies and names an identifying interface to be used by the crypto map for IPsec traffic. • *map-name*—Name that identifies the crypto map set. Enter the value specified in Step 22. • **local-address** *interface-id*—Name of interface that has the local address of the switch. **Note** The local address must belong to the FVRF. **Note** In VRF mode, the VPN feature supports up to 1024 local addresses. This limit is across the chassis (not per VPN module).
Step 30	Router(config)# **interface fastethernet** *slot/port*	Configures a Fast Ethernet interface and enters interface configuration mode.
Step 31	Router(config-if)# **ip vrf forwarding** *vrf-name*	Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF. Enter the value specified in Step 2.
Step 32	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 33	Router(config-if)# **no shutdown**	Enables the interface.
Step 34	Router(config-if)# **interface gigabitethernet** *slot/subslot/port*	Configures a Gigabit Ethernet interface. Match the value specified as the *interface-id* in step 28.
Step 35	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 36	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 37	Router(config-if)# **crypto engine outside**	Assigns the crypto engine to the interface.
Step 38	Router(config-if)# **no shutdown**	Enables the interface.

	Command	Purpose
Step 39	`Router(config-if)# exit`	Exits interface configuration mode.
Step 40	`Router(config)# interface vlan-id`	Configures a VLAN interface and enters interface configuration mode. • *vlan-id*—VLAN identifier.
Step 41	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF. Enter the value specified in Step 2.
Step 42	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 43	`Router(config-if)# crypto map map-name`	Applies a previously defined crypto map set to an interface. • *map-name*—Name that identifies the crypto map set. Enter the value specified in Step 22.
Step 44	`Router(config-if)# crypto engine slot slot/subslot inside`	Assigns the specified crypto engine to the interface. *slot/subslot*—The slot where the VSPA is located.
Step 45	`Router(config-if)# exit`	Exits interface configuration mode.
Step 46	`Router(config)# ip route vrf vrf-name prefix mask interface-number`	Establishes static routes for a VRF. • *vrf-name*—Name of the VRF for the static route. Enter the value specified in Step 2. • *prefix*—IP route prefix for the destination, in dotted-decimal format. • *mask*—Prefix mask for the destination, in dotted decimal format. • *interface-number*—Number identifying the network interface to use. Enter the *vlan-id* value specified in Step 40.
Step 47	`Router(config)# end`	Returns to privileged EXEC mode.

For complete configuration information for VRF-Aware IPsec, refer to this URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_vrf_aware_ipsec_ps6350_TSD_Products_Configuration_Guide_Chapter.html

For a configuration example, see the "VRF Mode Basic Configuration Example" section on page 4-21.

Configuring VPNs in VRF Mode with Tunnel Protection (GRE)

This section describes how to configure a VPN in VRF mode with tunnel protection (TP). Tunnel protection is GRE tunneling in VRF mode.

When you configure IPsec, a crypto map is attached to an interface to enable IPsec. With tunnel protection, there is no need for a crypto map or ACL to be attached to the interface. A crypto policy is attached directly to the tunnel interface. Any traffic routed by the interface is encapsulated in GRE and then encrypted using IPsec. The tunnel protection feature can be applied to point-to-point GRE.

VRF Mode Using Tunnel Protection Configuration Guidelines and Restrictions

When configuring tunnel protection on the VSPA follow these guidelines and restrictions:

- For tunnel protection to work, the VSPA must seize the GRE tunnel. Do not configure any options (such as sequence numbers or tunnel keys) that prevent the VSPA from seizing the GRE tunnel.
- Do not configure the GRE tunnel keepalive feature.
- When applied to the GRE tunnel interface, the **ip tcp adjust-mss** command is ignored. Apply the command to the ingress LAN interface instead. (CSCsl27876)
- Do not use crypto maps to protect GRE traffic in VRF mode.
- When a crypto map interface and a tunnel protection interface (either VTI or GRE/TP) share the same outside interface, they cannot share the same local source address.
- The **ip vrf forwarding** command is not required when configuring GRE with tunnel protection.
- To avoid fragmentation after encryption, set the tunnel IP MTU to be equal to or less than the egress interface MTU minus the GRE and IPsec overheads. The egress interface MTU must be the smallest MTU of all the active crypto outside interfaces.

Configuring a VPN in VRF Mode Using Tunnel Protection

To configure a VPN in VRF mode using tunnel protection, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mls mpls tunnel-recir`	Enables tunnel-MPLS recirculation.
Step 2	`Router(config)# crypto engine mode vrf`	Enables VRF mode for the VSPA. **Note** After enabling or disabling VRF mode using the **crypto engine mode vrf** command, you must reload the supervisor engine.
Step 3	`Router(config)# ip vrf vrf-name`	Configures a VRF routing table and enters VRF configuration mode. • *vrf-name*—Name assigned to the VRF.
Step 4	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables for a VRF. • *route-distinguisher*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1).

	Command	Purpose
Step 5	Router(config-vrf)# **route-target export** *route-target-ext-community*	Creates lists of export route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 3.
Step 6	Router(config-vrf)# **route-target import** *route-target-ext-community*	Creates lists of import route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 3.
Step 7	Router(config-vrf)# **exit**	Exits VRF configuration mode.
Step 8	Router(config)# **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring. • *fvrf-name*—(Optional) Front door virtual routing and forwarding (FVRF) name to which the keyring will be referenced. *fvrf-name* must match the FVRF name that was defined during virtual routing and forwarding (VRF) configuration.
Step 9	Router(config-keyring)# **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key*	Defines a preshared key to be used for IKE authentication. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • *hostname*—Fully qualified domain name of the peer. • *key*—Specifies the secret key.
Step 10	Router(config-keyring)# **exit**	Exits keyring configuration mode.
Step 11	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]]	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. Accepted values are described in the *Cisco IOS Security Command Reference*.
Step 12	Router(config-crypto-trans)# **exit**	Exits crypto transform configuration mode

	Command	Purpose
Step 13	`Router(config)#` **`crypto isakmp policy`** *`priority`*	Defines an IKE policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 14	`Router(config-isakmp)#` **`authentication pre-share`**	Specifies the authentication method with an IKE policy. • **pre-share**—Specifies preshared keys as the authentication method.
Step 15	`Router(config-isakmp)#` **`lifetime`** *`seconds`*	Specifies the lifetime of an IKE SA. • *seconds*—Number of seconds each SA should exist before expiring. Use an integer from 60 to 86,400 seconds. Default is 86,400 (one day.)
Step 16	`Router(config-isakmp)#` **`exit`**	Exits ISAKMP policy configuration mode.
Step 17	`Router(config)#` **`crypto isakmp profile`** *`profile-name`*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 18	`Router(config-isa-prof)#` **`keyring`** *`keyring-name`*	Configures a keyring within an ISAKMP profile. • *keyring-name*—Keyring name. This name must match the keyring name that was defined in global configuration. Enter the value specified in Step 7.
Step 19	`Router(config-isa-prof)#` **`match identity address`** *`address`* `[`*`mask`*`]`	Matches an identity from a peer in an ISAKMP profile. • *address* [*mask*]—IP address of the remote peer or a subnet and mask.
Step 20	`Router(config-isa-prof)#` **`exit`**	Exits ISAKMP profile configuration mode.
Step 21	`Router(config)#` **`access list`** *`access-list-number`* `{`**`deny`** `|` **`permit`**`}` **`ip host`** *`source`* **`host`** *`destination`*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Number of the host from which the packet is being sent. • *destination*—Number of the host to which the packet is being sent.
Step 22	`Router(config)#` **`crypto ipsec profile`** *`profile-name`*	Defines an IPsec profile and enters IPsec profile configuration mode. • *profile-name*—Name of the user profile.

	Command	Purpose
Step 23	Router(config-ipsec-profile)# **set transform-set** *transform-set-name*	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set. Enter the value specified in Step 10.
Step 24	Router(config-ipsec-profile)# **set isakmp-profile** *profile-name*	Sets the ISAKMP profile name. • *profile-name*—Name of the ISAKMP profile. Enter the value entered in Step 16.
Step 25	Router(config-ipsec-profile)# **exit**	Exits IPsec profile configuration mode.
Step 26	Router(config)# **interface** *tunnel-number*	Configures a tunnel interface and enters interface configuration mode. • *tunnel-number*—Name assigned to the tunnel interface.
Step 27	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF. Enter the value specified in Step 2.
Step 28	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 29	Router(config-if)# **tunnel source** *ip-address*	Sets the source address of a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel.
Step 30	Router(config-if)# **tunnel vrf** *vrf-name*	(Optional) Associates a VPN routing and forwarding instance (VRF) with a specific tunnel destination, interface or subinterface. This step is only required when configuring a front door VRF (FVRF). • *vrf-name*—Name assigned to the VRF.
Step 31	Router(config-if)# **tunnel destination** *ip*-address	Sets the destination address of a tunnel interface. • *ip-address*—IP address to use as the destination address for packets in the tunnel.
Step 32	Router(config-if)# **tunnel protection ipsec** *crypto-policy-name*	Associates a tunnel interface with an IPsec profile. • *crypto-policy-name*—Enter the value specified in Step 22.
Step 33	Router(config-if)# **crypto engine slot** *slot/subslot* **inside**	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the VSPA is located.
Step 34	Router(config-if)# **interface fastethernet** *slot/subslot*	Configures a Fast Ethernet interface.

	Command	Purpose
Step 35	`Router(config-if)# ip vrf forwarding vrf-name`	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 36	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 37	`Router(config-if)# no shutdown`	Enables the interface.
Step 38	`Router(config-if)# interface type slot/subslot/port`	Configures the physical egress interface.
Step 39	`Router(config-if)# ip vrf forwarding vrf-name`	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 40	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 29. • *mask*—Subnet mask.
Step 41	`Router(config-if)# crypto engine outside`	Assigns the crypto engine to the interface.
Step 42	`Router(config-if)# no shutdown`	Enables the interface.
Step 43	`Router(config-if)# exit`	Exits interface configuration mode.

For a configuration example, see the "VRF Mode Tunnel Protection Configuration Example" section on page 4-30.

Configuring an IPsec Virtual Tunnel Interface

The IPsec Virtual Tunnel Interface (VTI) provides a routable interface type for terminating IPsec tunnels that greatly simplifies the configuration process when you need to provide protection for remote access, and provides a simpler alternative to using GRE tunnels and crypto maps with IPsec. In addition, the IPsec VTI simplifies network management and load balancing.

Note IPsec VTI is not supported in crypto-connect mode.

Note the following details about IPsec VTI routing and traffic encryption:

- You can enable routing protocols on the tunnel interface so that routing information can be propagated over the virtual tunnel. The router can establish neighbor relationships over the virtual tunnel interface. Interoperability with standard-based IPsec installations is possible through the use of the IP ANY ANY proxy. The static IPsec interface will negotiate and accept IP ANY ANY proxies.
- The IPsec VTI supports native IPsec tunneling and exhibits most of the properties of a physical interface.

- In the IPsec VTI, encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static IP routing can be used to route the traffic to the virtual tunnel interface. Using IP routing to forward the traffic to encryption simplifies the IPsec VPN configuration because the use of ACLs with a crypto map in native IPsec configurations is not required. When IPsec VTIs are used, you can separate applications of NAT, ACLs, and QoS, and apply them to clear text or encrypted text, or both. When crypto maps are used, there is no easy way to specify forced encryption features.

IPsec Virtual Tunnel Interface Configuration Guidelines and Restrictions

When configuring IPsec VTI, follow these guidelines and restrictions:

- A VTI tunnel can terminate either in a VRF (normal VRF mode) or in the global context (with no **ip vrf forwarding** command on the tunnel interface).
- Only static VTI is currently supported.
- Only strict IP ANY ANY proxy is supported.
- The IPsec transform set must be configured only in tunnel mode.
- The IKE security association (SA) is bound to the virtual tunnel interface. Because it is bound to the virtual tunnel interface, the same IKE SA cannot be used for a crypto map.
- When the **mls mpls tunnel-recir** command is applied in a VTI configuration, one reserved VLAN is allocated to each tunnel. As a result, there will be a maximum limit of 1000 VTI tunnels.
- A static VTI tunnel interface supports multicast traffic.
- ACLs can be applied to GRE and static VTI tunnel interfaces participating in multicast traffic.
- Platform QoS features can be applied to GRE and static VTI tunnel interfaces participating in multicast traffic.

Configuring an IPsec Static Tunnel

To configure a static IPsec virtual tunnel interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mls mpls tunnel-recir`	Enables tunnel-MPLS recirculation.
Step 2	`Router(config)# crypto engine mode vrf`	Enables VRF mode for the VSPA. **Note** After enabling or disabling VRF mode using the **crypto engine mode vrf** command, you must reload the supervisor engine.
Step 3	`Router(config)# ip vrf vrf-name`	Configures a VRF routing table and enters VRF configuration mode. • *vrf-name*—Name assigned to the VRF.

	Command	Purpose
Step 4	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables for a VRF. • *route-distinguisher*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1).
Step 5	`Router(config-vrf)# route-target export route-target-ext-community`	Creates lists of export route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 3.
Step 6	`Router(config-vrf)# route-target import route-target-ext-community`	Creates lists of import route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 3.
Step 7	`Router(config-vrf)# exit`	Exits VRF configuration mode.
Step 8	`Router(config)# crypto keyring keyring-name [vrf fvrf-name]`	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring. • *fvrf-name*—(Optional) Front door virtual routing and forwarding (FVRF) name to which the keyring will be referenced. *fvrf-name* must match the FVRF name that was defined during virtual routing and forwarding (VRF) configuration.
Step 9	`Router(config-keyring)# pre-shared-key {address address [mask] | hostname hostname} key key`	Defines a preshared key to be used for IKE authentication. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • *hostname*—Fully qualified domain name of the peer. • *key*—Specifies the secret key.
Step 10	`Router(config-keyring)# exit`	Exits keyring configuration mode.

	Command	Purpose
Step 11	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]]	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. Accepted values are described in the *Cisco IOS Security Command Reference*.
Step 12	Router(config-crypto-trans)# **exit**	Exits crypto transform configuration mode.
Step 13	Router(config)# **crypto isakmp policy** *priority*	Defines an IKE policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 14	Router(config-isakmp)# **authentication pre-share**	Specifies the authentication method with an IKE policy. • **pre-share**—Specifies preshared keys as the authentication method.
Step 15	Router(config-isakmp)# **lifetime** *seconds*	Specifies the lifetime of an IKE SA. • *seconds*—Number of seconds each SA should exist before expiring. Use an integer from 60 to 86,400 seconds. Default is 86,400 (one day.)
Step 16	Router(config-isakmp)# **exit**	Exits ISAKMP policy configuration mode.
Step 17	Router(config)# **crypto ipsec profile** *profile-name*	Defines an IPsec profile and enters IPsec profile configuration mode. The IPsec profile defines the IP Security (IPsec) parameters that are to be used for IPsec encryption between two IPsec routers. • *profile-name*—Name of the user profile.
Step 18	Router(config-ipsec-profile)# **set transform-set** *transform-set-name* [*transform-set-name2* ...*transform-set-name6*]	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set.
Step 19	Router(config)# **interface** *type slot*/[*subslot*]/*port*	Configures an interface type. • *type*—Type of interface being configured. • *slot*/[*subslot*]/*port*—Number of the slot, subslot (optional), and port to be configured.
Step 20	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.

	Command	Purpose
Step 21	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 22	`Router(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel as IPsec and the transport as IPv4.
Step 23	`Router(config-if)# tunnel source ip-address`	Sets the source address of a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel.
Step 24	`Router(config-if)# tunnel destination ip-address`	Sets the destination address of a tunnel interface. • *ip-address*—IP address to use as the destination address for packets in the tunnel.
Step 25	`Router(config-if)# tunnel vrf vrf-name`	(Optional) Associates a VPN routing and forwarding instance (VRF) with a specific tunnel destination. This step is only required when configuring a front door VRF (FVRF). • *vrf-name*—Name assigned to the VRF.
Step 26	`Router(config-if)# tunnel protection ipsec profile name`	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command in Step 1.
Step 27	`Router(config-if)# crypto engine slot slot/subslot inside`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the VSPA is located.
Step 28	`Router(config-if)# interface type slot/subslot/port`	Configures the physical egress interface.
Step 29	`Router(config-if)# ip vrf forwarding vrf-name`	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 30	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 23. • *mask*—Subnet mask.
Step 31	`Router(config-if)# crypto engine outside`	Assigns the crypto engine to the interface.
Step 32	`Router(config-if)# no shutdown`	Enables the interface.
Step 33	`Router(config-if)# exit`	Exits interface configuration mode.

Verifying the IPsec Virtual Tunnel Interface Configuration

To confirm that your IPsec virtual tunnel interface configuration is working properly, enter the **show interfaces tunnel**, **show crypto session**, and **show ip route** commands.

The **show interfaces tunnel** command displays tunnel interface information, the **show crypto session** command displays status information for active crypto sessions, and the **show ip route** command displays the current state of the routing table.

In this display the Tunnel 0 is up and the line protocol is up. If the line protocol is down, the session is not active.

```
Router1# show interfaces tunnel 0

Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport IPSEC/IP, key disabled, sequencing disabled
Tunnel TTL 255
Checksumming of packets disabled, fast tunneling enabled
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPSec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
191320 packets input, 30129126 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out

Router1# show crypto session
Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAs: 4, origin: crypto map

Router1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.0.35.0/24 is directly connected, Ethernet3/3
S 10.0.36.0/24 is directly connected, Tunnel0
C 10.0.51.0/24 is directly connected, Tunnel0
C 10.0.149.0/24 is directly connected, Ethernet3/0
```

For more complete information about IPsec Virtual Tunnel Interface, refer to the following URL:

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

For IPsec Virtual Tunnel Interface configuration examples, see the "IPsec Virtual Tunnel Interfaces Configuration Examples" section on page 4-32.

Configuring VTI in the Global Context

You can configure IPsec VTI without having to configure VRFs. Although VRF mode must be configured globally using the **crypto engine mode vrf** command, tunnels can be terminated in the global context rather than in VRFs.

The configuration steps for VTI in the global context are similar to the steps for IPsec VTI shown in the "Configuring an IPsec Static Tunnel" section on page 4-16 with the exception that the **ip vrf forwarding** *vrf-name* command and the **tunnel vrf** *vrf-name* command are not required.

For a configuration example of IPsec VTI in the global context, see the "IPsec Virtual Tunnel Interfaces Configuration Examples" section on page 4-32.

Configuration Examples

The following sections provide examples of VRF mode configurations:

- VRF Mode Basic Configuration Example, page 4-21
- VRF Mode Remote Access Using Easy VPN Configuration Example, page 4-24
- VRF Mode PE Configuration Example, page 4-27
- VRF Mode CE Configuration Example, page 4-28
- VRF Mode Tunnel Protection Configuration Example, page 4-30
- IP Multicast in VRF Mode Configuration Example, page 4-31
- IPsec Virtual Tunnel Interfaces Configuration Examples, page 4-32

Note When the **ip vrf forwarding** command is applied to a VLAN, any previously existing IP address assigned to that VLAN is removed. To assign an IP address to the VLAN, enter the **ip address** command after the **ip vrf forwarding** command, not preceding it.

VRF Mode Basic Configuration Example

The following example shows a basic VSPA configuration using VRF mode:

Switch 1 Configuration

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
vlan 2,3
```

```
!
crypto keyring key0
  pre-shared-key address 11.0.0.2 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
!
crypto isakmp profile prof1
   vrf ivrf
   keyring key0
   match identity address 11.0.0.2 255.255.255.255
!
!
crypto ipsec transform-set proposal1  esp-3des esp-sha-hmac
!
crypto map testtag local-address Vlan3
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 set isakmp-profile prof1
 match address 101
!
interface GigabitEthernet1/1
 !switch inside port
 ip vrf forwarding ivrf
 ip address 12.0.0.1 255.255.255.0
!
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 3
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip vrf forwarding ivrf
 ip address 13.0.0.252 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0 inside
!
```

```
interface Vlan3
 ip address 11.0.0.1 255.255.255.0
 crypto engine slot 4/0 outside
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
```

Switch 2 Configuration

```
hostname router-2
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
vlan 2,3
!
crypto keyring key0
  pre-shared-key address 11.0.0.1 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
!
crypto isakmp profile prof1
   vrf ivrf
   keyring key0
   match identity address 11.0.0.1 255.255.255.255
!
!
crypto ipsec transform-set proposal1  esp-3des esp-sha-hmac
!
crypto map testtag local-address Vlan3
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 set isakmp-profile prof1
 match address 101
!
interface GigabitEthernet1/1
 !switch inside port
 ip vrf forwarding ivrf
 ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 3
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip vrf forwarding ivrf
 ip address 12.0.0.252 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0 inside
!
interface Vlan3
 ip address 11.0.0.2 255.255.255.0
 crypto engine slot 4/0 outside
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
```

VRF Mode Remote Access Using Easy VPN Configuration Example

The following examples show VRF mode configurations for remote access using Easy VPN, first using RADIUS authentication, then using local authentication:

Using RADIUS Authentication

```
aaa group server radius acs-vrf1
 server-private 192.1.1.251 auth-port 1812 acct-port 1813 key allegro
 ip vrf forwarding vrf1
!
aaa authentication login test_list group acs-vrf1
aaa authorization network test_list group acs-vrf1
aaa accounting network test_list start-stop group acs-vrf1
!
ip vrf ivrf
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
!
crypto isakmp policy 5
 encr 3des
 authentication pre-share
 group 2

crypto isakmp client configuration group test
 key world
 pool pool1
!
crypto isakmp profile test_pro
   vrf ivrf
   match identity group test
   client authentication list test_list
   isakmp authorization list test_list
   client configuration address respond
   accounting test_list
crypto ipsec transform-set t3 esp-3des esp-sha-hmac
!
crypto dynamic-map remote 1
 set transform-set t3
 set isakmp-profile test_pro
 reverse-route
!
!
crypto map map-ra local-address GigabitEthernet2/1
crypto map map-ra 10 ipsec-isakmp dynamic remote
!
interface GigabitEthernet2/1
  mtu 9216
 ip address 120.0.0.254 255.255.255.0
 ip flow ingress
 logging event link-status
 mls qos trust ip-precedence
 crypto engine slot 1/0 outside
!
interface GigabitEthernet1/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,100,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!

interface Vlan100
 ip vrf forwarding vrf1
 ip address 120.0.0.100 255.255.255.0
 no mop enabled
 crypto map map-ra
 crypto engine slot 1/0 inside

ip local pool pool1 100.0.1.1 100.0.5.250
```

Using Local Authentication

```
username t1 password 0 cisco
aaa new-model
!
aaa authentication login test_list local
aaa authorization network test_list local
!
```

```
aaa session-id common
!
ip vrf ivrf
 rd 1:2
 route-target export 1:2
 route-target import 1:2


!
crypto isakmp policy 5
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp client configuration group test
 key world
 pool pool1
crypto isakmp profile test_pro
   vrf ivrf
   match identity group test
   client authentication list test_list
   isakmp authorization list test_list
   client configuration address respond
   accounting test_list
crypto ipsec transform-set t3 esp-3des esp-sha-hmac
!
crypto dynamic-map remote 10
 set transform-set t3
 set isakmp-profile test_pro
 reverse-route

!
!
crypto map map-ra local-address GigabitEthernet2/1
crypto map map-ra 11 ipsec-isakmp dynamic remote
!
!

!
interface GigabitEthernet2/1
  mtu 9216
 ip address 120.0.0.254 255.255.255.0
 ip flow ingress
 logging event link-status
 mls qos trust ip-precedence
 crypto engine slot 1/0 outside
!
!
interface GigabitEthernet1/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,100,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan100
 ip vrf forwarding ivrf
 ip address 120.0.0.100 255.255.255.0
 ip flow ingress
 crypto map map-ra
 crypto engine slot 1/0 inside
!
!
ip local pool pool1 100.0.1.1 100.0.5.250
```

VRF Mode PE Configuration Example

The following example shows a VRF mode configuration for a provider edge (PE):

```
!
hostname PE
!
crypto isakmp enable
!
!
vlan 2-3
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
crypto keyring key0
  pre-shared-key address 11.0.0.1 key cisco
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 lifetime 500
crypto isakmp keepalive 10
crypto isakmp profile prof1
   vrf ivrf
   keyring key0
   match identity address 11.0.0.1 255.255.255.255
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag local-address Vlan3
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set security-association lifetime seconds 1000
 set transform-set proposal1
 set pfs group1
 set isakmp-profile prof1
 match address 101
!
!
```

```
!
interface GigabitEthernet3/1
 ip vrf forwarding ivrf
 ip address 13.0.0.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet3/9
 switchport
 switchport access vlan 3
 switchport mode access
 no shutdown
!
interface GigabitEthernet2/0/1
!IPsec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
interface GigabitEthernet2/0/2
!IPsec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
interface Vlan2
 ip vrf forwarding ivrf
 ip address 12.0.0.252 255.255.255.0
 crypto map testtag
 crypto engine slot 2/0 inside
 no shutdown
!
interface Vlan3
 ip address 11.0.0.2 255.255.255.0
 crypto engine outside
 no shutdown
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

VRF Mode CE Configuration Example

The following example shows a VRF mode configuration for a customer edge (CE):

```
!
hostname CE
!
crypto isakmp enable
!
vlan 2-3
!
crypto isakmp policy 1
```

```
 encr 3des
 authentication pre-share
 lifetime 500
crypto isakmp key cisco address 11.0.0.2
crypto isakmp keepalive 10
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set security-association lifetime seconds 1000
 set transform-set proposal1
 set pfs group1
 match address 101
!
!
!
interface GigabitEthernet3/1
 ip address 12.0.0.1 255.255.255.0
 load-interval 30
 no keepalive
 no shutdown
!
interface GigabitEthernet3/9
 switchport
 switchport access vlan 3
 switchport mode access
 no shutdown
!
interface GigabitEthernet4/1/1
!IPsec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk

interface GigabitEthernet4/1/2
!IPsec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 3
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 2/0
 no shutdown
!
interface Vlan3
 no ip address
 crypto connect vlan 2
 no shutdown
!
```

```
ip route 13.0.0.0 255.0.0.0 11.0.0.2
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
end
```

VRF Mode Tunnel Protection Configuration Example

The following example shows a VRF mode configuration with tunnel protection:

```
ip vrf coke
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto keyring key1
 pre-shared-key address 100.1.1.1 key happy-eddie
!
crypto isakmp policy 1
 authentication pre-share

crypto isakmp profile prof1
 keyring key1
 match identity address 100.1.1.1 255.255.255.255
!
crypto ipsec transform-set TR esp-des esp-md5-hmac
!
crypto ipsec profile tp
 set transform-set TR
 set isakmp-profile prof1
!
!
crypto engine mode vrf
!
interface Tunnel1
 ip vrf forwarding coke
 ip address 10.1.1.254 255.255.255.0
 tunnel source 172.1.1.1
 tunnel destination 100.1.1.1
 tunnel protection ipsec profile tp
 crypto engine slot 4/0 inside
!
interface GigabitEthernet4/0/1
 !IPsec VPN Module inside port
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 cdp enable
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN Module outside port
 no ip address
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 cdp enable
 spanning-tree portfast trunk
!
interface GigabitEthernet6/1
 ip address 172.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
interface FastEthernet7/13
 ip vrf forwarding coke
 ip address 13.1.1.2 255.255.255.0
!
ip route 100.1.1.1 255.255.255.255 Tunnel1
```

IP Multicast in VRF Mode Configuration Example

The following example shows how to configure IP multicast over GRE:

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
!
!
ip multicast-routing vrf ivrf
!
crypto engine mode vrf
!
!
!
crypto keyring key1
  pre-shared-key address 11.0.0.0 255.0.0.0 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp profile isa_prof
   keyring key1
   match identity address 11.0.0.0 255.0.0.0
!
crypto ipsec transform-set proposal esp-3des
!
crypto ipsec profile vpnprof
 set transform-set proposal
 set isakmp-profile isa_prof
!
!
!
interface Tunnel1
 ip vrf forwarding ivrf
 ip address 20.1.1.1 255.255.255.0
 ip mtu 9216
 ip hold-time eigrp 1 3600
 ip pim sparse-mode
 tunnel source 1.0.1.1
 tunnel destination 11.1.1.1
 tunnel protection ipsec profile vpnprof
 crypto engine slot 4/0 inside
!
interface Loopback1
 ip address 1.0.1.1 255.255.255.0
!
interface GigabitEthernet1/1
 mtu 9216
 ip vrf forwarding ivrf
 ip address 50.1.1.1 255.0.0.0
 ip pim sparse-mode
!
interface GigabitEthernet1/2
 mtu 9216
 ip address 9.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
router eigrp 1
 !
 address-family ipv4 vrf ivrf
  autonomous-system 1
  network 20.1.1.0 0.0.0.255
  network 50.1.1.0 0.0.0.255
  no auto-summary
  no eigrp log-neighbor-changes
 exit-address-family
!
router ospf 1
 log-adjacency-changes
 network 1.0.0.0 0.255.255.255 area 0
 network 9.0.0.0 0.255.255.255 area 0
!
ip pim vrf ivrf rp-address 50.1.1.1
!
```

IPsec Virtual Tunnel Interfaces Configuration Examples

The following examples show VRF mode configurations that use VTI:

- IPsec Virtual Tunnel Interface FVRF Configuration Example, page 4-33

- IPsec Virtual Tunnel Interface in the Global Context Configuration Example, page 4-34
- IPsec Virtual Tunnel Interface Multicast Configuration Example, page 4-35

IPsec Virtual Tunnel Interface FVRF Configuration Example

The following example configuration shows an FVRF VTI configuration:

```
hostname router-1
!
!
ip vrf fvrf
 rd 2000:1
 route-target export 2000:1
 route-target import 2000:1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
crypto keyring key1 vrf fvrf
  pre-shared-key address 11.1.1.1 key cisco47
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
!
crypto isakmp profile isa_prof
   keyring key1
   match identity address 11.1.1.1 255.255.255.255 fvrf

crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
!
crypto ipsec profile vpnprof
 set transform-set proposal
 set isakmp-profile isa_prof
!
!
!
!
!
interface Tunnel1
 ip vrf forwarding ivrf
 ip address 20.1.1.1 255.255.255.0
 ip ospf network broadcast
 ip ospf priority 2
 tunnel source 1.0.0.1
 tunnel destination 11.1.1.1
 tunnel mode ipsec ipv4
 tunnel vrf fvrf
 tunnel protection ipsec profile vpnprof
 crypto engine slot 4/0 inside
!
interface Loopback1
 ip vrf forwarding fvrf
 ip address 1.0.0.1 255.255.255.0
!
interface GigabitEthernet1/1
```

```
 !switch inside port
 ip vrf forwarding ivrf
 ip address 50.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 ip vrf forwarding fvrf
 ip address 9.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
interface GigabitEthernet4/0/1
 !IPsec VPN Module inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN Module outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
router ospf 1 vrf ivrf
 log-adjacency-changes
 network 20.1.1.0 0.0.0.255 area 0
 network 21.1.1.0 0.0.0.255 area 0
 network 50.0.0.0 0.0.0.255 area 0
!
ip classless
ip route vrf fvrf 11.1.1.0 255.255.255.0 9.1.1.254
```

IPsec Virtual Tunnel Interface in the Global Context Configuration Example

The following example configuration shows IPsec VTI configuration in the global context:

```
!
crypto engine mode vrf
!
crypto keyring key1
  pre-shared-key address 14.0.0.2 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share

!
crypto isakmp profile prof1
   keyring key1
   match identity address 14.0.0.2 255.255.255.255
!
crypto ipsec transform-set t-set1 esp-3des esp-sha-hmac
```

```
!
crypto ipsec profile prof1
 set transform-set t-set1
 set isakmp-profile prof1
!
!
interface Tunnel1
 ip address 122.0.0.2 255.255.255.0
 tunnel source 15.0.0.2
 tunnel destination 14.0.0.2
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile prof1
 crypto engine slot 2/0 inside
!
interface Loopback2
 ip address 15.0.0.2 255.255.255.0
!

interface GigabitEthernet1/3
 ip address 172.2.1.1 255.255.255.0
 crypto engine slot 2/0 outside
!
interface GigabitEthernet2/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet2/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
!
ip route 14.0.0.0 255.0.0.0 172.2.1.2
ip route 172.0.0.0 255.0.0.0 172.2.1.2
```

IPsec Virtual Tunnel Interface Multicast Configuration Example

The following example shows how to configure multicast over VTI:

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
!
!
ip multicast-routing vrf ivrf
!
crypto engine mode vrf
```

```
!
!
!
crypto keyring key1
  pre-shared-key address 11.0.0.0 255.0.0.0 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp profile isa_prof
   keyring key1
   match identity address 11.0.0.0 255.0.0.0
!
crypto ipsec transform-set proposal esp-3des
!
crypto ipsec profile vpnprof
 set transform-set proposal
 set isakmp-profile isa_prof
!
!
!
interface Tunnel1
 ip vrf forwarding ivrf
 ip address 20.1.1.1 255.255.255.0
 ip mtu 9216
 ip hold-time eigrp 1 3600
 ip pim sparse-mode
 tunnel source 1.0.1.1
 tunnel destination 11.1.1.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vpnprof
 crypto engine slot 4/0 inside
!
interface Loopback1
 ip address 1.0.1.1 255.255.255.0
!
interface GigabitEthernet1/1
 mtu 9216
 ip vrf forwarding ivrf
 ip address 50.1.1.1 255.0.0.0
 ip pim sparse-mode
!
interface GigabitEthernet1/2
 mtu 9216
 ip address 9.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
router eigrp 1
 !
 address-family ipv4 vrf ivrf
  autonomous-system 1
  network 20.1.1.0 0.0.0.255
  network 50.1.1.0 0.0.0.255
  no auto-summary
  no eigrp log-neighbor-changes
 exit-address-family
!
router ospf 1
 log-adjacency-changes
 network 1.0.0.0 0.255.255.255 area 0
 network 9.0.0.0 0.255.255.255 area 0
!
ip pim vrf ivrf rp-address 50.1.1.1
!
```

CHAPTER 5

Configuring IPsec VPN Fragmentation and MTU

This chapter provides information about configuring IPsec VPN fragmentation and the maximum transmission unit (MTU). It includes the following sections:

- Understanding IPsec VPN Fragmentation and MTU, page 5-1
- Configuring IPsec Prefragmentation, page 5-9
- Configuring MTU Settings, page 5-11
- Configuration Examples, page 5-13

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

Note Some illustrations in this chapter refer to the IPsec VPN SPA. In these instances, the VSPA performs the equivalent function.

Understanding IPsec VPN Fragmentation and MTU

This section includes the following topics:

- Overview of Fragmentation and MTU, page 5-1
- IPsec Prefragmentation, page 5-2
- Fragmentation in Different Modes, page 5-3

Overview of Fragmentation and MTU

When a packet is nearly the size of the maximum transmission unit (MTU) of the physical egress port of the encrypting switch, and it is encapsulated with IPsec headers, it probably will exceed the MTU of the egress port. This situation causes the packet to be fragmented after encryption (post-fragmentation), which requires the IPsec peer to perform reassembly before decryption, degrading its performance. To minimize post-fragmentation, you can set the MTU in the upstream data path to ensure that most fragmentation occurs before encryption (prefragmentation). Prefragmentation for IPsec VPNs avoids performance degradation by shifting the reassembly task from the receiving IPsec peer to the receiving end hosts.

Note In this document, prefragmentation refers to fragmentation prior to any type of encapsulation, such as IPsec or GRE. IPsec prefragmentation refers to fragmentation prior to IPsec encryption.

To ensure prefragmentation in most cases, we recommend the following MTU settings:

- The crypto interface VLAN MTU associated with the VSPA should be set to be equal or less than the egress interface MTU.
- For GRE over IPsec, the IP MTU of the GRE tunnel interface should be set below the egress interface MTU by at least the overhead of IPsec encryption and the 24-byte GRE+IP header (20-byte IP header plus 4-byte GRE header). Because options such as tunnel key (RFC 2890) are not supported, the GRE+IP IP header will always be 24 bytes.

Note The crypto interface VLAN MTU, the egress interface MTU, and the IP MTU of the GRE tunnel interface are all Layer 3 parameters.

The following are additional guidelines for IPsec prefragmentation and MTU in crypto-connect mode:

- If a packet's DF (Don't Fragment) bit is set and the packet exceeds the MTU at any point in the data path, the packet will be dropped. To prevent a packet drop, clear the DF bit by using either policy-based routing (PBR) or the **crypto df-bit clear** command.
- If GRE encapsulation is not taken over by the VSPA, and if the packets exceed the IP MTU of the GRE tunnel interface, the route processor will fragment and encapsulate the packets.

Note If the supervisor engine performs GRE encapsulation, the encapsulated packets will have the DF bit set.

For general information on fragmentation and MTU issues, see "Resolve IP Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPsec" at this URL:

http://www.cisco.com/en/US/tech/tk827/tk369/technologies_white_paper09186a00800d6979.shtml

IPsec Prefragmentation

In the IPsec prefragmentation process (also called Look-Ahead Fragmentation, or LAF), the encrypting switch can predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPsec security association (SA). IPsec prefragmentation avoids reassembly by the receiving switch before decryption and helps improve overall IPsec traffic throughput by shifting the reassembly task to the end hosts.

A packet will be fragmented before encryption if either of the following conditions is met:

- The encrypted packet will exceed the MTU of the crypto interface VLAN.
- The clear packet exceeds the tunnel MTU.

Fragmentation in Different Modes

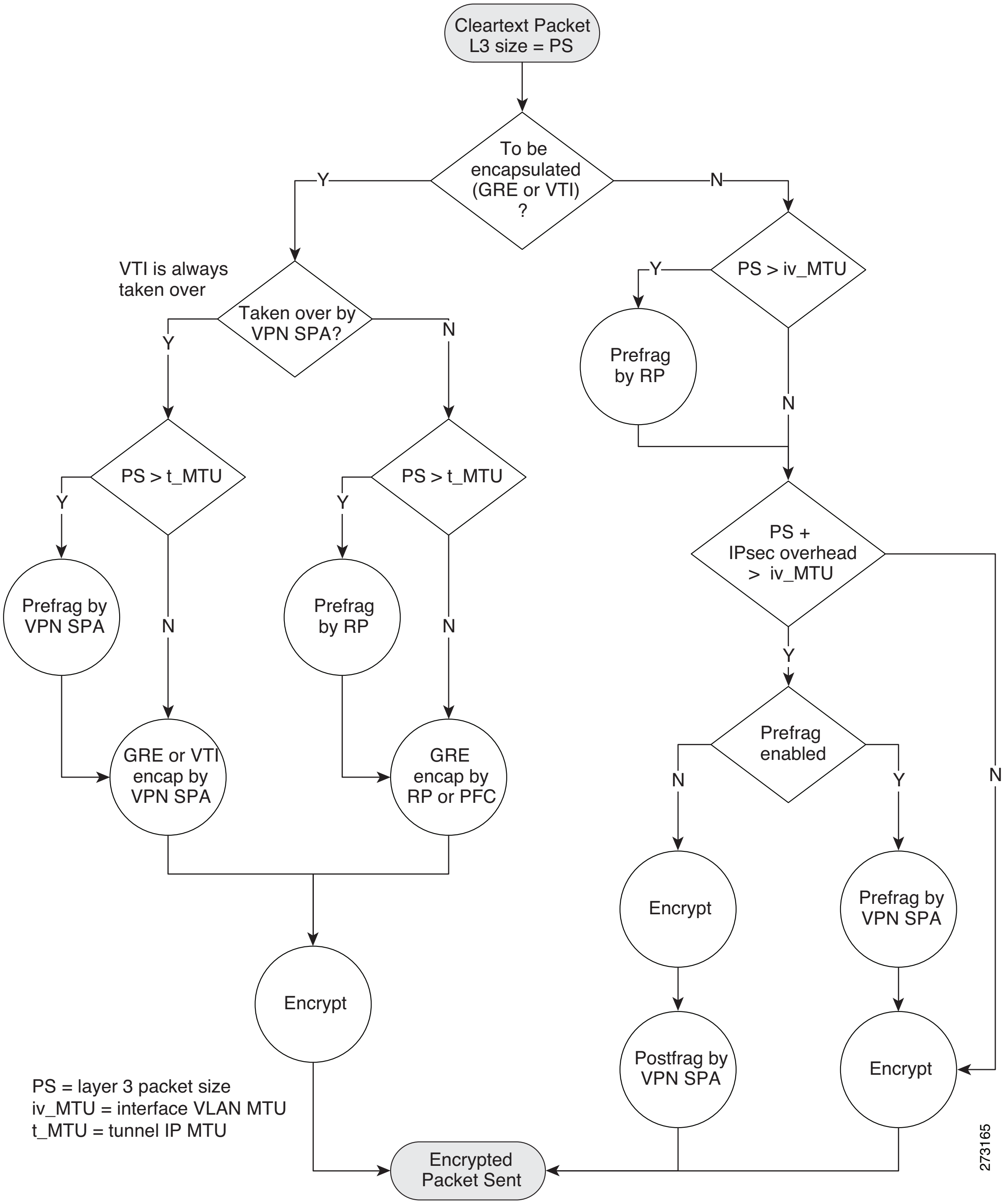
The fragmentation process differs depending on the IPsec VPN mode and whether GRE or virtual tunnel interface (VTI) is used. The process is described in the following sections:

- Overview of the Fragmentation Process, page 5-4
- Fragmentation of IPsec Packets in Crypto-Connect Mode, page 5-5
- Fragmentation of GRE Packets in Crypto-Connect Mode, page 5-6
- Fragmentation of IPsec Packets in VRF Mode, page 5-7
- Fragmentation of GRE Packets in VRF Mode, page 5-7
- Fragmentation of IPsec Packets Using VTI, page 5-8

Overview of the Fragmentation Process

Figure 5-1 shows the fragmentation process for IPsec packets in all VPN modes.

Figure 5-1 Fragmentation of IPsec Packets in All VPN Modes



These notes apply to the fragmentation process:

- The fragmentation process described in Figure 5-1 applies only when the DF (Don't Fragment) bit is not set for cleartext packets entering the flowchart. If a packet requires fragmentation and the DF bit is set, the packet will be dropped.
- VTI encapsulation is always taken over by the VSPA.
- GRE encapsulation of RP-generated packets is never taken over by the VSPA.
- GRE encapsulation of mGRE packets is never taken over by the VSPA.
- The VSPA will perform only a single fragmentation operation, either prefragmentation or postfragmentation, but not both.
- Fragmentation is based on the IP MTU of the tunnel or of the crypto interface VLAN, not the egress interface.
- Path MTU discovery (PMTUD) is supported in both crypto-connect and VRF modes.
- The **ip tcp adjust-mss** command is supported in all modes.

Fragmentation of IPsec Packets in Crypto-Connect Mode

For fragmentation of packets in crypto-connect mode, the following are the MTU setting requirements and recommendations:

- The configured IP MTU of the interface VLAN
  - Prefragmentation of traffic by the VSPA is based on this MTU.
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces configured on the port VLAN, or packets will be dropped.
- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the interface VLAN.

In the following example, a 1500-byte cleartext packet will not be fragmented by the RP, because it is within the MTU of the interface VLAN. The cleartext packet will be fragmented by the VSPA, because the IPsec overhead would cause the encrypted packet to exceed the MTU of the interface VLAN.

A 1600-byte cleartext packet will first be fragmented by the RP, because the packet exceeds the MTU of the interface VLAN. The packet will then be fragmented again by the VSPA, because the IPsec overhead added by the encryption process would cause the encrypted packet to exceed the MTU of the interface VLAN.

```
interface GigabitEthernet1/1
  ! switch inside port
  mtu 9216
  ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
  ! switch outside port
  ! mtu 1500 by default
  switchport
  switchport access vlan 502
  switchport mode access
!
interface Vlan2
  ! interface vlan
  ! mtu 1500 by default
  ip address 11.0.0.2 255.255.255.0
  crypto map testtag
  crypto engine slot 4/0
!
interface Vlan502
  ! port vlan
  no ip address
  crypto connect vlan 2
!
```

Fragmentation of GRE Packets in Crypto-Connect Mode

For fragmentation of packets in crypto-connect mode, the following are the MTU setting requirements and recommendations:

- The configured IP MTU of the crypto interface VLAN
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces configured on the port VLAN, or packets will be dropped.
- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the crypto interface VLAN.
- The configured IP MTU of the GRE tunnel interface
  - Prefragmentation of traffic by the VSPA is based on this MTU.
  - You must set this MTU so that IPsec-encrypted GRE packets will not exceed the IP MTU of the crypto interface VLAN, or packets will be dropped. This requirement applies regardless of whether the GRE tunnel is taken over by the VSPA.

In the following example, if the tunnel is taken over by the VSPA, a 1600-byte cleartext packet will be fragmented by the VSPA, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated and IPsec-encrypted by the VSPA.

If the tunnel is not taken over by the VSPA, a 1600-byte cleartext packet will be fragmented by the RP, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated by the PFC and IPsec-encrypted by the VSPA.

```
interface Tunnel1
  ip mtu 1400
  ip address 1.0.0.1 255.255.255.0
  tunnel source Vlan2
  tunnel destination 11.0.0.2
!
interface GigabitEthernet1/1
  ! switch inside port
  mtu 9216
  ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
  ! switch outside port
  ! mtu 1500 by default
  switchport
  switchport access vlan 502
  switchport mode access
!
interface Vlan2
  ! mtu 1500 by default
  ip address 11.0.0.1 255.255.255.0
  no mop enabled
  crypto map testtag
  crypto engine slot 4/0
```

```
!
interface Vlan502
  no ip address
  crypto connect vlan 2
!
```

Fragmentation of IPsec Packets in VRF Mode

For fragmentation of packets in VRF mode, the following are the MTU setting requirements and recommendations:

- The MTU of the crypto interface VLAN.
  - Prefragmentation by the VSPA will be based on this MTU.
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces, or packets will be dropped.
- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the crypto interface VLAN.

In the following example, a 1500-byte cleartext packet will not be fragmented by the RP, because it is within the MTU of the interface VLAN. The cleartext packet will be fragmented by the VSPA, because the IPsec overhead would cause the encrypted packet to exceed the MTU of the interface VLAN.

A 1600-byte cleartext packet will first be fragmented by the RP, because the packet exceeds the MTU of the interface VLAN. The packet will then be fragmented again by the VSPA, because the IPsec overhead added by the encryption process would cause the encrypted packet to exceed the MTU of the interface VLAN.

```
interface GigabitEthernet1/1
  ! switch inside port
  mtu 9216
  ip vrf forwarding ivrf
  ip address 12.0.0.1 255.255.255.0
!
!
interface GigabitEthernet1/2
  ! switch outside port
  ! mtu 1500 by default
  ip address 11.0.0.1 255.255.255.0
  crypto engine slot 4/0 outside
!
interface Vlan2
  ! mtu 1500 by default
  ip vrf forwarding ivrf
  ip address 13.0.0.252 255.255.255.0
  crypto map testtag
  crypto engine slot 4/0 inside
!
```

Fragmentation of GRE Packets in VRF Mode

For fragmentation of packets in VRF mode, the following are the MTU setting requirements and recommendations:

- The MTU of the crypto interface VLAN.
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces configured on the port VLAN, or packets will be dropped.

- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the crypto interface VLAN.
- The configured IP MTU of the GRE tunnel interface
  - Prefragmentation by the VSPA will be based on this MTU.
  - You must set this MTU so that IPsec-encrypted GRE packets will not exceed the minimum MTU of the physical egress interfaces, or packets will be dropped. This requirement applies regardless of whether the GRE tunnel is taken over by the VSPA.

In the following example, if the tunnel is taken over by the VSPA, a 1600-byte cleartext packet will be fragmented by the VSPA, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated and IPsec-encrypted by the VSPA.

If the tunnel is not taken over by the VSPA, a 1600-byte cleartext packet will be fragmented by the RP, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated by the PFC and IPsec-encrypted by the VSPA.

```
interface Tunnel1
  ip mtu 1400
  ip vrf forwarding coke
  ip address 10.1.1.254 255.255.255.0
  tunnel source 172.1.1.1
  tunnel destination 100.1.1.1
  tunnel protection ipsec profile tp
  crypto engine slot 4/0 inside
!
interface GigabitEthernet6/1
  ! switch outside port
  ! mtu 1500 by default
  ip address 172.1.1.1 255.255.255.0
  crypto engine slot 4/0 outside
!
interface FastEthernet7/13
  ! switch inside port
  mtu 9216
  ip vrf forwarding coke
  ip address 13.1.1.2 255.255.255.0
!
```

Fragmentation of IPsec Packets Using VTI

The following are the relevant MTU settings for fragmentation of static virtual tunnel interface (sVTI) packets:

- The IP MTU of the VTI tunnel interface.
  - Prefragmentation by the VSPA will be based on this MTU.
  - Configuring this MTU is unnecessary because it is automatically adjusted to accommodate the IPsec overhead.

Note We recommend that the IP MTU of the VTI tunnel interface be left at its default value. If you change it, be sure that it does not exceed the MTU of the physical egress interface minus the IPsec overhead.

The fragmentation behavior using VTI is the same as the behavior shown in the "Fragmentation of GRE Packets in VRF Mode" section on page 5-7 for the case in which the tunnel is taken over by the VSPA.

Configuring IPsec Prefragmentation

IPsec prefragmentation can be configured globally or at the interface level. By default, IPsec prefragmentation is enabled globally. Enabling or disabling IPsec prefragmentation at the interface will override the global configuration.

IPsec Prefragmentation Configuration Guidelines

Note Tunnels support only IPsec prefragmentation; postfragmentation is not supported. The guidelines in this section apply only to an interface to which a crypto map is applied.

When configuring IPsec prefragmentation, follow these guidelines:

- To configure IPsec prefragmentation at the interface level, apply it on the interface to which the crypto map is applied.
- If an IPsec peer is experiencing high CPU utilization with large packet flows, verify that IPsec prefragmentation is enabled (the peer may be reassembling large packets).
- IPsec prefragmentation for IPsec VPNs operates in IPsec tunnel mode. It does not apply in IPsec transport mode.
- IPsec prefragmentation for IPsec VPNs functionality depends on the **crypto ipsec df-bit** configuration of the interface to which the crypto map is applied, and on the incoming packet “do not fragment” (DF) bit state. For general information about IPsec prefragmentation, see the following URL:

  http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftprefrg.html
- GRE+IP encapsulation adds 24 bytes to the packet size. When configuring for prefragmentation based on anticipated GRE overhead, use this value.
- IPsec encryption adds a number of bytes to the packet size depending on the configured IPsec transform set. When configuring for prefragmentation based on anticipated IPsec overhead, use the following table of worst-case IPsec overhead bytes for various IPsec transform sets:

IPsec Transform Set	IPsec Overhead, Maximum Bytes
esp-aes-(256 or 192 or 128) esp-sha-hmac or md5	73
esp-aes (256 or 192 or 128)	61
esp-3des, esp-des	45
esp-(des or 3des) esp-sha-hmac or md5	57
esp-null esp-sha-hmac or md5	45
ah-sha-hmac or md5	44

Configuring IPsec Prefragmentation Globally

IPsec prefragmentation is globally enabled by default. To enable or disable prefragmentation for IPsec VPNs at the global level, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto ipsec fragmentation before-encryption**	Enables prefragmentation for IPsec VPNs globally.
Step 2	Router(config)# **crypto ipsec fragmentation after-encryption**	Disables prefragmentation for IPsec VPNs globally.

Configuring IPsec Prefragmentation at the Interface

IPsec prefragmentation is globally enabled by default. To enable or disable prefragmentation for IPsec VPNs at the interface level, perform this task beginning in interface configuration mode for the interface to which the crypto map is attached:

	Command	Purpose
Step 1	Router(config-if)# **crypto ipsec fragmentation before-encryption**	Enables prefragmentation for IPsec VPNs on the interface.
Step 2	Router(config-if)# **crypto ipsec fragmentation after-encryption**	Disables prefragmentation for IPsec VPNs on the interface.

Note Enabling or disabling IPsec prefragmentation at the interface will override the global configuration.

Verifying the IPsec Prefragmentation Configuration

To verify that IPsec prefragmentation is enabled, consult the interface statistics on the encrypting switch and the decrypting switch. If fragmentation occurs on the encrypting switch, and no reassembly occurs on the decrypting switch, fragmentation is occurring before encryption, which means that the packets are not being reassembled before decryption and the feature is enabled.

To verify that the IPsec prefragmentation feature is enabled, enter the **show running-configuration** command on the encrypting switch. If the feature is enabled, no fragmentation feature will appear in the command output:

```
Router# show running-configuration

crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
!!! the postfragmentation feature appears here if IPsec prefragmentation is disabled
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

If IPsec prefragmentation has been disabled, the postfragmentation feature will appear in the command output:

```
Router# show running-configuration

crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
crypto ipsec fragmentation after-encryption
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

To display the configuration of the encrypting switch interface VLAN, enter the **show running-configuration interface** command. If the IPsec prefragmentation feature is enabled, a prefragmentation statement will appear in the command output:

```
Router# show running-configuration interface vlan2

interface Vlan2
 ip address 15.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 1/0
 crypto ipsec fragmentation before-encryption
```

If the IPsec prefragmentation feature has been disabled at the interface VLAN, a postfragmentation statement will appear in the command output:

```
Router# show running-configuration interface vlan2

interface Vlan2
 ip address 15.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 1/0
 crypto ipsec fragmentation after-encryption
end
```

Configuring MTU Settings

The Cisco IOS software supports several types of configurable maximum transmission unit (MTU) options at various levels of the protocol stack. You should ensure that all MTU values are consistent to avoid unnecessary fragmentation of packets.

MTU Settings Configuration Guidelines and Restrictions

When configuring MTU settings for a VSPA, follow these guidelines and note these restrictions:

- The MTU value used by the VSPA for fragmentation decisions is based on the IP MTU of the tunnel or of the crypto interface VLAN, not the egress interface. For information on the recommended MTU settings, see the "Fragmentation in Different Modes" section on page 5-3.
- If you have GRE tunneling configured, see the "Fragmentation in Different Modes" section on page 5-3 for information on the recommended MTU settings.

Note For additional information on fragmentation of packets, see the "Configuring IPsec Prefragmentation" section on page 5-9.

Changing the Physical Egress Interface MTU

You can configure either the Layer 3 MTU or the IP MTU of the physical egress interface. To change the MTU value on a physical egress interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Enters interface configuration mode for the interface.
Step 2	`Router(config-if)# mtu bytes`	Configures the maximum transmission unit (MTU) size for the interface. • *bytes*—The range is 1500 to 9216; the default is 1500.

[1] *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Changing the Tunnel Interface IP MTU

You can configure the IP MTU of the tunnel interface, but you cannot configure the Layer 3 MTU. To change the IP MTU value on a tunnel, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface tunnel_name`	Enters interface configuration mode for the tunnel.
Step 2	`Router(config-if)# ip mtu bytes`	Configures the IP MTU size for the tunnel. • *bytes*—The minimum is 68; the maximum and the default depend on the interface medium.

Changing the Interface VLAN MTU

You can configure the Layer 3 MTU of the interface VLAN. To change the MTU value on an interface VLAN, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface vlan_ID`	Enters interface configuration mode for the VLAN.
Step 2	`Router(config-if)# mtu bytes`	Configures the MTU size for the interface VLAN. • *bytes*—The range is 64 to 9216; the default is 1500.

Verifying the MTU Size

To verify the MTU size for an interface, enter the **show interface** command or the **show ip interface** command, as shown in the following examples:

To display the MTU value for a secure port, enter the **show interface** command:

```
Router# show interface g1/1

GigabitEthernet1/1 is up, line protocol is up (connected)
Hardware is C6k 1000Mb 802.3, address is 000a.8ad8.1c4a (bia 000a.8ad8.1c4a)
MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
reliability 255/255, txload 1/255, rxload 1/255
...
```

To display the MTU size for an interface VLAN, enter the **show interface** command:

```
Router# show interface vlan2
Vlan2 is up, line protocol is up
  Hardware is EtherSVI, address is 000e.39ad.e700 (bia 000e.39ad.e700)
  Internet address is 192.168.1.1/16
  MTU 1000 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
...
```

To display the IP MTU value for a GRE tunnel, enter the **show ip interface** command:

```
Router# show ip interface tunnel 2

Tunnel2 is up, line protocol is up
Internet address is 11.1.0.2/16
Broadcast address is 255.255.255.255
Address determined by non-volatile memory
MTU is 1450 bytes
...
```

Configuration Examples

The following sections provide examples of IPsec prefragmentation configurations:

- Crypto-Connect Mode IPsec Prefragmentation Configuration Example, page 5-13
- VRF Mode with GRE using Tunnel Protection IPsec Prefragmentation Configuration Example, page 5-15

Crypto-Connect Mode IPsec Prefragmentation Configuration Example

The following example shows an IPsec prefragmentation configuration using crypto-connect mode:

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
```

```
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ! mtu 1500 by default
  ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
  mtu 1000
  switchport
  switchport access vlan 502
  switchport mode access
!
interface GigabitEthernet4/0/1
  !VSPA inside port
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,2,1002-1005
  switchport mode trunk
  mtu 9216
  flowcontrol receive on
  flowcontrol send off
  spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
  !VSPA outside port
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,502,1002-1005
  switchport mode trunk
  mtu 9216
  flowcontrol receive on
  flowcontrol send off
  spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
  mtu 1000
  ip address 11.0.0.2 255.255.255.0
  crypto map testtag
  crypto engine slot 4/0
!
interface Vlan502
  !port vlan
  no ip address
  crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

VRF Mode with GRE using Tunnel Protection IPsec Prefragmentation Configuration Example

The following example shows an IPsec prefragmentation configuration using VRF mode with GRE and tunnel protection:

```
!
hostname router-1
!
ip vrf coke
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto keyring key1
 pre-shared-key address 100.1.1.1 key happy-eddie
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp profile prof1
 keyring key1
 match identity address 100.1.1.1 255.255.255.255
!
crypto ipsec transform-set TR esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile tp
 set transform-set TR
 set isakmp-profile prof1
!
!
crypto engine mode vrf
!
interface Tunnel1
  ip mtu 1400
  ip vrf forwarding coke
  ip address 10.1.1.254 255.255.255.0
  tunnel source 172.1.1.1
  tunnel destination 100.1.1.1
  tunnel protection ipsec profile tp
  crypto engine slot 4/0 inside
!
interface GigabitEthernet4/0/1
  !VSPA inside port
  flowcontrol receive on
  flowcontrol send off
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,1002-1005
  switchport mode trunk
  cdp enable
  spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
  !VSPA outside port
  no ip address
  flowcontrol receive on
  flowcontrol send off
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,1002-1005
  switchport mode trunk
  cdp enable
  spanning-tree portfast trunk
!
interface GigabitEthernet6/1
  ! mtu 1500 by default
  ip address 172.1.1.1 255.255.255.0
  crypto engine slot 4/0 outside
!
interface FastEthernet7/13
  ip vrf forwarding coke
  ip address 13.1.1.2 255.255.255.0
!
ip route 100.1.1.1 255.255.255.255 Tunnel1
end
```

CHAPTER 6

Configuring Quality of Service

This chapter provides information about configuring quality of service (QoS) features using the VSPA on the Catalyst 6500 Series switch. It includes the following sections:

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

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

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

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

Understanding QoS in the VSPA

Typical applications of quality of service (QoS) for VPN are the use of traffic policing to prevent a hub from overwhelming a lower-capacity spoke, and the prioritization over VPN of delay-sensitive traffic such as voice over IP (VoIP). In a system including the VSPA, QoS features for VPN traffic are provided by the VSPA module, its carrier card (SSC-600), and the platform (Catalyst 6500 Series switch).

- Module QoS—The VSPA provides traffic shaping, queuing, and bandwidth reservation services before encryption. Policies are attached to a crypto engine within the interface configuration.
- Carrier QoS—For each crypto engine, the SSC-600 provides a dual-priority queue for module traffic. Policies are attached to a crypto engine.
- Platform QoS—The Catalyst 6500 Series switch provides trust, data classification, and either remarking or policing of traffic on the tunnel interface. Policies are attached to an interface.

To activate the QoS capabilities of the module, carrier, and platform, you must enable QoS globally by entering the **mls qos** command.

When QoS is disabled globally, the system behavior is as follows:

- All QoS fields are left intact in packets.
- Packets flow through only one queue in the carrier card.

When QoS is enabled globally, the default system behavior is as follows:

- The default state of all ports and VLANs is the untrusted state, causing ports to clear the QoS fields in all traffic to zero unless a QoS policy is configured on the port.
- Packets flow through two queues in the carrier card. Packets with a CoS value of 5 will use the higher priority queue, while all other packets will use the lower priority queue.

Before configuring QoS for VPN, see the additional information provided in the following URLs:

Configuring QoS on the Catalyst 6500 Series switch:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/qos.html

Configuring QoS Features on a SIP:

http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/6500series/76cfgsip.html#Configuring_QoS_Features_on_a_SIP

Configuring QoS on the FlexWAN Modules:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexqos.html

QoS Policing on the Catalyst 6500 Series switch:

http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a00801c8c4b.shtml

QoS Output Scheduling on the Catalyst 6500 Series switch:

http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a008015bf98.shtml

QoS Troubleshooting:

http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a008074d6b1.shtml

Using the Module QoS Features of the VSPA

In VRF mode configurations using Virtual Tunnel Interface (VTI) or GRE with tunnel protection (TP), the VSPA can provide traffic shaping, queuing, and bandwidth reservation of outbound traffic before encryption, allowing you to prioritize traffic on a per-tunnel basis as well as to configure a shape rate for each tunnel. This section contains the following topics:

Classifying, Marking, and Policing Traffic

To apply the VSPA's QoS features, you must first ensure that the class of service (CoS) of packets is marked on ingress and that any necessary policing is performed before the packets are passed to the VSPA.

The Catalyst 6500 Series switch performs classification, marking, and policing of traffic to the VSPA. These functions are configured using the following commands:

- Use the **class-map** command to classify types of traffic.
- Use the **set** command to mark the CoS or DSCP bits for a traffic class.
- Use the **police** command to limit the rate of a traffic class.

For details on configuring classification, marking, and policing, see the "Using the Platform QoS Features of the Switch" section on page 6-8.

Setting Priority

For each tunnel, the VSPA provides one high-priority low-latency queue (LLQ) for latency-sensitive outbound traffic, such as VoIP. The high priority queue is served ahead of other queues in that tunnel. The **priority** policy-map class configuration command gives priority to a class of traffic belonging to a policy map, causing that traffic to be diverted to the high-priority queue. Only one priority level per tunnel is supported. When the **priority** command is used in a class map, no form of the **bandwidth** command is allowed in the same class map.

Shaping Traffic

The **shape average** policy-map class configuration command specifies a maximum data rate for a class of outbound traffic. While policing enforces a maximum rate by dropping or marking down excess packets, shaping queues the excess packets for sending at a later time. Packets exceeding the maximum rate will be delayed but will not be dropped unless excess traffic is sustained at rates higher than the configured shape rate for long periods of time, causing shape buffers to overflow.

When shaping is applied to a tunnel, all traffic in the tunnel must be included in the default class. Any additional classes must be defined in a child policy.

To configure traffic shaping in the VSPA, use the **shape average** *rate bc be* command, where the *rate* argument specifies the maximum average bit rate and the optional *be* argument is the allowed excess burst level. The optional *bc* argument (the committed burst size) is ignored, but if *be* is specified, then *bc* must be configured to a value of at least the number of bits transferred during 4 milliseconds of traffic at the shape *rate*. The **shape average** command can be configured only for the tunnel top-level policy. It cannot be used in a child policy.

Reserving Bandwidth

The **bandwidth** policy-map class configuration command reserves a minimum bandwidth for a class of traffic. You can configure the **bandwidth** command in a child policy to reserve either an absolute rate or a percentage of the tunnel shape rate. If the **priority** command is configured on another class map within the same policy map, only the **bandwidth remaining** form of the **bandwidth** command can be used, since the higher priority traffic overrules any bandwidth guarantees.

When you configure bandwidth reservation for a class, your settings are checked for capacity and oversubscription relative to the maximum shape rate. If a tunnel aggregate shaper is not configured, any configuration of bandwidth reservation will be rejected.

Setting the Queue Limit

The **queue-limit** policy-map class configuration command specifies the maximum number of packets the queue can hold for a class policy configured in a policy map. The VSPA supports only a packet-based queue limiting, and supports queue-limit configuration only on a class map.

Failover

If you deploy two VSPAs for intrachassis stateful failover using a blade failure group (BFG), the QoS configuration on the active VSPA is automatically reflected on the standby module. During a failover, packets in the queue are lost. The standby VSPA takes over, scheduling newly-received packets according to the QoS configuration. Interchassis failover is not supported.

For more information on failover using a blade failure group, see "Configuring Intrachassis IPsec Stateful Failover Using a Blade Failure Group" section on page 11-10.

Configuring Module QoS

Module QoS configuration in the VSPA uses the Cisco Modular QoS CLI (MQC) framework. You can define traffic classes, associate policies and actions to each traffic class, and attach these policies to interfaces by following these steps:

Step 1 Define traffic classes using **match** statements with the **class-map** command.

Step 2 Configure policies using the defined traffic classes with the **policy-map** command.

Step 3 Within the interface configuration, attach policies to a crypto engine with the **service-policy** command.

For the module QoS, attach the service policy to the tunnel interface in the config-crypto-engine configuration mode after entering the **crypto-engine** interface level command.

The VSPA supports a hierarchical policy using two service policy levels:

- A parent policy, supporting only a single default class, to apply a QoS mechanism to a traffic aggregate.
- A child policy to apply a QoS mechanism to a flow or subset of the aggregate.

Logical interfaces, such as subinterfaces and tunnel interfaces, require a hierarchical policy with the traffic-shaping feature at the parent level and queuing at lower levels. While the traffic-shaping feature regulates the output rate, queuing may introduce additional latency or cause packet drops when the ingress traffic rate surpasses the configured queuing capacity.

For each tunnel, the VSPA supports a child policy with up to 8 classes, including the default-class. Only one of the 8 traffic classes can be configured as a priority class on a tunnel interface. You can configure bandwidth reservation on any class that is not configured as the priority class. You cannot configure shaping on a traffic class (a child shaper); a single aggregate shaper can be configured in the parent policy.

Module QoS Configuration Guidelines and Restrictions

When configuring QoS settings for the VSPA, follow these guidelines and note these restrictions:

- To use the QoS features of the VSPA, you must enable QoS globally by entering the **mls qos** command.
- Because the VSPA performs QoS functions only on tunnel interfaces associated with the VSPA, configuring module QoS on a tunnel interface will always result in the tunnel being taken over.
- The VSPA performs QoS functions only on VTI or GRE/TP interfaces in VRF mode. The QoS functions are not supported with crypto connect mode or DMVPN.
- The QoS functions operate only on IPv4 traffic.
- QoS is supported for up to 2 K VTI tunnels or 1 K GRE/TP tunnels.
- The VSPA supports a maximum of 8 traffic classes per tunnel, including the default class.
  - We recommend that you configure one class as class-default.
  - One traffic class can be configured as priority, to be processed ahead of all other classes. This class is typically used for voice or other latency-sensitive traffic.
  - Each class can be configured separately for bandwidth reservation and a queue limit.
  - You cannot configure priority setting and bandwidth reservation within the same class map.
- When configuring bandwidth reservation, note the following guidelines:
  - Bandwidth reservation means a minimum bandwidth guarantee when 100 percent of the configured shape rate is utilized. If less than 100 percent is used, any class may use the available bandwidth above its configured reservation.
  - If no bandwidth is reserved for the default class, then 1 percent of the shape rate will be automatically reserved for the default class.
- The VSPA supports one aggregate shaper per tunnel, to be defined at the tunnel (parent) level. All traffic within the tunnel must be included in the shaper. If a shaper is defined, only the class-default class should be defined at the tunnel level, with the shaper applied to it. All other traffic classes must be defined in child policies.
- Any tunnel that uses module QoS functions must have a shaping policy.
- Because the VSPA relies on the ToS/CoS bits to classify and queue the packets properly, you should ensure that packets arriving at the VSPA have already been properly classified and marked.
- The dropping policy is Random Early Detection (RED), and the RED parameters are not configurable. You cannot configure fair queueing.
- Bandwidth is reserved per class for each tunnel independently. The minimum bandwidth guarantee on a class level will not propagate to the tunnel level. There is no bandwidth guarantee on a tunnel. You cannot configure an explicit minimum rate at the tunnel level.
- You should avoid any policy that causes the reordering or dropping of post-encrypted packets.
- The configuration of priority applies only within the tunnel in which it is configured, and does not affect other tunnels.
- Increasing the queue limit increases latency.

Configuring a Child and Parent Policy

To configure a child and parent policy, perform this task:

	Command	Purpose
Step 1	Router(config)# **policy-map** *child_policy_name*	Enters the policy map configuration for the specified child policy map.
Step 2	Router(config-pmap)# **class** [*child_policy_name* | **class-default**]	Enters the policy map class configuration for the default class map.
Step 3	Router(config-pmap-c)# **priority**	(Optional) Enables strict-priority (low latency queuing) on the class.
Step 4	Router(config-pmap-c)# **bandwidth** {*kbps* | **bandwidth percent** *percentage* | **bandwidth remaining percent** *percentage*}	(Optional) Enables minimum bandwidth reservation on a traffic class. • **bandwidth** *kbps* — Specifies the reserved bandwidth as an absolute value in kbps that cannot exceed the configured tunnel shape rate. • **bandwidth percent** *percentage* — Specifies the reserved bandwidth as a percentage of the configured tunnel shape rate. • **bandwidth remaining percent** *percentage* — Specifies the reserved bandwidth as a percentage of the remaining tunnel bandwidth up to the configured tunnel shape rate after all LLQ packets have been served.
Step 5	Router(config-pmap-c)# **queue-limit** *number_of_packets*	(Optional) Sets the maximum size (in packets) of the traffic queue for the class.
Step 6	Router(config-pmap-c)# **exit**	Exits the policy map class configuration.
Step 7	Router(config-pmap)# **exit**	Exits the policy map configuration.
Step 8	Router(config)# **policy-map** *parent_policy_name*	Enters the policy map configuration for the specified parent policy map.
Step 9	Router(config-pmap)# **class class-default**	Enters the policy map class configuration for the default class map.
Step 10	Router(config-pmap-c)# **shape average** *rate* [*bc be*]	Enables average rate traffic shaping. • *rate*—Specifies the committed information rate (CIR), in bits per second (bps). • *bc*—(Optional) Specifies the committed burst size, in bits. This field will be ignored, but must be set to a legal value if *be* is specified. • *be*—(Optional) Specifies the excess burst size, in bits.
Step 11	Router(config-pmap-c)# **service-policy** *child_policy_name*	(Optional) Attaches a child policy map with up to seven additional class maps. Including the class-default class map, there can be a total of up to eight class maps.

	Command	Purpose
Step 12	`Router(config-pmap-c)#` **`exit`**	Exits the policy map class configuration.
Step 13	`Router(config-pmap)#` **`exit`**	Exits the policy map configuration.

- The **bandwidth** and **bandwidth percent** commands cannot be configured in conjunction with the **priority** command. The **bandwidth remaining percent** command can be configured in conjunction with the **priority** command.
- By default, the queue limit is 1000 for all non-LLQ traffic classes; for LLQ classes, the default is the number of packets that can be transferred in 4 milliseconds at the configured shape rate.
- The shape *rate* can range from 128 Kbps to 1 Gbps. If a tunnel has a low shape rate, we recommend that you also configure a small excess burst size (*be*).
- The default excess burst size (*be*) is the number of bits transferred during 4 milliseconds of traffic at the shape *rate*. For example, for a 256000 bps shape rate, the default excess burst size will be 1024 bits.
- If you configure *be*, then you must configure *bc* (the committed burst size) to a value of at least the number of bits transferred during 4 milliseconds of traffic at the shape *rate*.

Note We recommend that you allow the system to determine settings for *bc* and *be*.

Configuring Shaping on a Tunnel

To configure shaping on a tunnel, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`mls qos`**	Enables QoS globally.
Step 2	`Router(config)#` **`interface`** *`tunnel_name`*	Selects the tunnel to which the policy map will be applied.
Step 3	`Router(config-if)#` **`crypto-engine`**	Enters QoS policy configuration mode for the VSPA.
Step 4	`Router(config-crypto-engine)#` **`service-policy output`** *`parent_policy_name`*	Applies the policy map to tunnel egress traffic.
Step 5	`Router(config-crypto-engine)#` **`end`**	Exits to the privileged EXEC mode.
Step 6	`Router#` **`show policy-map interface`** *`tunnel_name`*	Displays the statistics and configurations of the QoS policies attached to the tunnel interface.

For QoS configuration examples, see the "QoS Configuration Examples" section on page 6-11.

Using the Carrier QoS Features of the SSC-600

The SSC-600 implements a two-level, strict-priority QoS with two queues for each direction, inbound and outbound. Packets are dequeued in a two-to-one ratio, meaning that two packets are dequeued from the high-priority low-latency queue (LLQ) before one packet is dequeued from the low-priority queue. Packets are enqueued based on your priority-queue configuration settings. To take advantage of the

SSC-600's QoS capability, you must use standard QoS commands to ensure that the class of service (CoS) of packets is marked on ingress. You must configure the CoS map for the inside and outside ports and you must also enable QoS globally for the SSC-600 to acknowledge the CoS mapping.

Carrier QoS Configuration Guidelines and Restrictions

When configuring QoS settings for an SSC-600, follow these guidelines and note these restrictions:

- Packets are enqueued based on the **mls qos** command and the priority-queue configuration settings as follows:
  - When the **mls qos** command is not configured, all data packets are enqueued into the high-priority queue.
  - When the **mls qos** command is configured and no explicit priority-queue configuration is present on the VSPA Ethernet interfaces, only packets with a CoS value of 5 are enqueued into the high-priority queue; all other packets are enqueued into the low-priority queue.
  - When the **mls qos** command is configured and priority-queue configuration is present on the VSPA Ethernet interfaces, traffic is enqueued based on the priority-queue configuration.
- A maximum of three CoS map values can be sent to the high-priority queue. Because the CoS value of 5 is preconfigured as high-priority, you can choose only two other values for high-priority queueing.

Note Do not configure more than three CoS map values because any additional values will overwrite previously configured values. If you overwrite the CoS value of 5, the system will restore it, overwriting one of your other configured values. To restore an overwritten CoS map value, you must first delete the new value and then reconfigure the earlier value.

- When the **mls qos** command is configured, you must also configure the **mls qos trust** command on the VSPA Ethernet interfaces, as in the following example:

```
Interface GigabitEthernet4/0/1
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
!
Interface GigabitEthernet4/0/2
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
```

  In this example, the CoS values of 0, 1, and 5 are sent to the high-priority queue.

- In a blade failover group, both VSPAs must have matching carrier QoS configurations.
- If the **mls qos trust** command is not configured, the QoS fields in all traffic will be cleared to the default level. If the **mls qos trust** command is configured, the QoS fields will be preserved.

For a configuration example of module QoS, see the "Module QoS Configuration Example" section on page 6-13.

Using the Platform QoS Features of the Switch

The Catalyst 6500 Series switch allows data classification and either remarking or policing of packets on the tunnel interface.

Platform QoS configuration uses the Cisco Modular QoS CLI (MQC) framework. You can define traffic classes, associate policies and actions to each traffic class, and attach these policies to interfaces by following these steps:

Step 1 Define traffic classes using **match** statements with the **class-map** command.

Step 2 Configure policies using the defined traffic classes with the **policy-map** command.

Step 3 Attach defined policies to an interface with the **service-policy** command.

For more information on configuring QoS in the Catalyst 6500 Series switch, see the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/qos.html

Remarking of Packets

Remarking is specified by using a **set** command within a policy map. Platform QoS for VPN is capable of remarking the priority settings of original IP, GRE, and IPsec headers of tunnel traffic, depending on the tunnel type. For remarking of packets, the port trust settings must be as follows:

- If matching is based on ToS bits of the incoming packets, the LAN interface must be configured as a trusted port, using the **mls qos trust** command.
- Depending on the VPN mode and the desired behavior, the inside interface of the VSPA must be configured as VLAN-based, using the **mls qos vlan-based** command, or as a trusted port, using the **mls qos trust** command.

The following sections describe remarking behavior in different modes.

Remarking of GRE Packets in Crypto-Connect Mode

The outside interface of the VSPA must be configured as a trusted port, using the **mls qos trust** command.

The inside interface of the VSPA is configured as VLAN-based, using the **mls qos vlan-based** command. The remarking behavior will be as follows:

- Remarking is supported for both inbound and outbound traffic.
- If the GRE tunnel is taken over by the VSPA:
  - Apply the service policy to the tunnel interface. Any service policy on the crypto interface VLAN will be ignored.
  - Remarking is performed on the original IP header and copied to the GRE header and the IPsec header.
- If the GRE tunnel is not taken over by the VSPA:
  - Apply the service policy to the tunnel interface or to the crypto interface VLAN.
  - If the service policy is applied to the tunnel interface, remarking is performed on the original IP header and copied to the GRE header and the IPsec header.
  - If the service policy is applied to the crypto interface VLAN, remarking is performed on the GRE header and copied to the IPsec header. Remarking is not performed on the original IP header (inner header).
- Traffic matching the service policy will be remarked.

Remarking of GRE Packets with Tunnel Protection in VRF Mode

The outside interface of the VSPA must be configured as a trusted port, using the **mls qos trust** command. After egress from the VSPA, an encrypted packet cannot be remarked.

The inside interface of the VSPA is configured as VLAN-based, using the **mls qos vlan-based** command. The remarking behavior will be as follows:

- Remarking is supported for both inbound and outbound traffic.
- Apply the service policy to the tunnel interface.
- If the GRE tunnel is taken over by the VSPA, remarking is performed on the original IP header and copied to the GRE header and the IPsec header.
- If the GRE tunnel is not taken over by the VSPA, remarking is performed on the GRE header and copied to the IPsec header. Remarking is not performed on the original IP header (inner header).
- Traffic matching the service policy will be remarked.

Remarking of VTI Packets in VRF Mode

The outside interface of the VSPA must be configured as a trusted port, using the **mls qos trust** command. After egress from the VSPA, an encrypted packet cannot be remarked.

The inside interface of the VSPA is configured as VLAN-based, using the **mls qos vlan-based** command. The remarking behavior will be as follows:

- Apply the service policy to the tunnel interface.
- Remarking is supported for both inbound and outbound traffic.
- Remarking is performed on the original IP header and the IPsec header.
- Traffic matching the service policy will be remarked.

Policing of Packets

Policing enforces a maximum packet rate by dropping excess packets. Policing can limit the rate that packets are passed to and from the VSPA.

For more information on configuring policing in the Catalyst 6500 Series switch, see the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/qos.html

Platform QoS Guidelines and Restrictions

When configuring platform QoS for VPN, follow these guidelines and note these restrictions:

- To enable QoS, you must apply the **mls qos** command globally.
- Platform QoS policies can be applied to the tunnel interface in any of these situations:
  - GRE in crypto-connect mode, whether or not GRE is taken over by the VSPA
  - GRE with tunnel protection, whether or not GRE is taken over by the VSPA
  - Static VTI
  - mGRE

- The **match all, match not,** and **match cos** classification criteria are not supported for platform QoS on the tunnel interface.
- In the **police** command, the **exceed-action set** options are not supported for platform QoS on the tunnel interface, and cannot be configured for remarking or policing.
- Platform QoS policies do not apply to packets generated by the route processor or destined for the route processor.
- Platform QoS supports 1023 policers or remarkers.
- When applying a service policy during configuration, the policy does not take effect until after you exit the interface configuration mode.
- If you apply and then remove a service policy, some packets will be remarked to other priorities during the transition. To prevent this issue, enter the **mls acl tcam default-result permit** command in the global configuration.
- In some cases, upon removal of a service policy from a tunnel, the tunnel continues to remark outbound traffic. To prevent this issue, remove the IP access list first, then remove the service policy.
- In VRF mode, the outside interface is always untrusted. To provide QoS to inbound traffic, apply a service policy to the GRE or VTI tunnel interface, or to the interface VLAN.
- When configuring for blade-to-blade failover, you must enter identical QoS configurations on the inside interfaces of both VSPAs.

For a configuration example of platform QoS, see the "Platform QoS Configuration Example" section on page 6-11.

QoS Configuration Examples

This section provides examples of the following configurations:

- Platform QoS Configuration Example, page 6-11
- Carrier QoS Configuration Example, page 6-13
- Module QoS Configuration Example, page 6-13

Platform QoS Configuration Example

This example shows how to configure platform QoS with inbound and outbound service policies:

```
mls qos
!
! Define class maps
!
class-map match-any IPP1
 match ip precedence 1
class-map match-any IPP0
 match ip precedence 0
class-map match-any IPP3
 match ip precedence 3
class-map match-any IPP2
 match ip precedence 2
class-map match-any IPP5
 match ip precedence 5
class-map match-any IPP4
 match ip precedence 4
```

```
class-map match-any IPP7
 match ip precedence 7
class-map match-any IPP6
 match ip precedence 6
!
! Define policy maps
!
policy-map SET_3TO5
 class IPP3
  set precedence 5
!
policy-map SET_1TO5
 class IPP1
  set precedence 5
!
!
! LAN interface configuration
!
interface GigabitEthernet2/3
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 46,51,3501-4000
 switchport mode trunk
 mls qos trust ip-precedence
!
! inside interface configuration
!
interface GigabitEthernet3/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 mls qos vlan-based
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
! outside interface configuration
!
interface GigabitEthernet3/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
!
! tunnel interface configuration
!
interface Tunnel1
 ip vrf forwarding i1
 ip address 26.0.1.2 255.255.255.0
 ip access-group T1ACL_IN in
 ip access-group T1ACL_OUT out
 ip mtu 1400
 tunnel source 27.0.1.2
 tunnel destination 192.0.20.1
 tunnel mode ipsec ipv4
 tunnel vrf f1
 tunnel protection ipsec profile TUN_PROTECTION
 crypto engine slot 3/0 inside
 service-policy input SET_1TO5
 service-policy output SET_3TO5
```

Carrier QoS Configuration Example

The following example shows how to configure carrier QoS:

```
mls qos
!
Interface GigabitEthernet4/0/1
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
!
Interface GigabitEthernet4/0/2
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
```

Module QoS Configuration Example

The following example shows how to configure module QoS:

```
upgrade fpd auto
version 12.2
service timestamps debug datetime
service timestamps log datetime
no service password-encryption
service internal
service counters max age 10
!
hostname HUB2
!
boot-start-marker
boot system disk0:
boot-end-marker
!
logging buffered 1000000
!
no aaa new-model
clock timezone PST -8
ip subnet-zero
!
!
no ip domain-lookup
ip domain-name cisco.com
!
vtp domain same_domain
vtp mode off
mls qos
mls netflow interface
no mls flow ip
no mls flow ipv6
mls ip slb purge global
no mls acl tcam share-global
mls cef error action reset
mls mpls tunnel-recir
call admission limit 90
!
crypto pki trustpoint MSCA
 enrollment mode ra
 enrollment url http://43.0.111.111:80/certsrv/mscep/mscep.dll
 serial-number
 ip-address none
 subject-name cn=HUB2,ou=isbu,o=cisco
 revocation-check none
!
!
crypto pki certificate chain MSCA
 certificate 1C67C77C0000000004C4
 certificate ca 7C0299B7C394F789436EBEFCCEAED66D
crypto engine mode vrf
crypto engine gre vpnblade
!
!
!
!
!
fabric timer 15
!
power redundancy-mode combined
diagnostic bootup level minimal
diagnostic monitor syslog
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
no spanning-tree vlan 2-7
!
!
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
vlan internal allocation policy descending
vlan access-log ratelimit 2000
!
vlan 1
 tb-vlan1 1002
 tb-vlan2 1003
!
vlan 2-1001
!
vlan 1002
 tb-vlan1 1
 tb-vlan2 1003
!
vlan 1003
 tb-vlan1 1
 tb-vlan2 1002
 parent 1005
 backupcrf enable
!
vlan 1004
 bridge 1
 stp type ibm
!
vlan 1005
 bridge 1
!
class-map match-any class7
  match  dscp cs7
```

```
class-map match-any class6
  match  dscp cs6
class-map match-any class5
  match  dscp cs5
class-map match-any class4
  match  dscp cs4
class-map match-any class3
  match  dscp cs3
class-map match-any class2
  match  dscp cs2
class-map match-any class1
  match  dscp cs1
class-map match-any class567
  match  dscp cs5  cs6  cs7
class-map match-any class34
  match  dscp cs3  cs4
class-map match-any class12
  match  dscp cs1  cs2
!
!
policy-map Tunnel0ChildPolicy
  class class567
    priority
    queue-limit 100 packets
  class class34
    bandwidth remaining percent 40
  class class12
    bandwidth remaining percent 40
  class class-default
    bandwidth remaining percent 20
!
policy-map Tunnel0ParentPolicy
  class class-default
    shape average 1544000
   service-policy Tunnel0ChildPolicy
!
policy-map Tunnel1ChildPolicy
  class class7
    bandwidth percent 20
    queue-limit 100 packets
  class class6
    bandwidth percent 20
    queue-limit 100 packets
  class class5
    bandwidth percent 10
    queue-limit 100 packets
  class class4
    bandwidth percent 10
  class class3
    bandwidth percent 10
  class class2
    bandwidth percent 10
  class class1
    bandwidth percent 10
  class class-default
    bandwidth percent 10
!
policy-map Tunnel1ParentPolicy
  class class-default
    shape average 34000000 136000 0
   service-policy Tunnel1ChildPolicy
!
policy-map Tunnel2ChildPolicy
  class class7
```

```
    bandwidth 20000
  class class6
    bandwidth 20000
  class class5
    bandwidth 10000
  class class4
    bandwidth 10000
  class class3
    bandwidth 10000
  class class2
    bandwidth 10000
  class class1
    bandwidth 10000
  class class-default
    bandwidth 10000
!
policy-map Tunnel2ParentPolicy
  class class-default
    shape average 100000000
   service-policy Tunnel2ChildPolicy
!
policy-map Tunnel3ChildPolicy
  class class567
    bandwidth percent 30
  class class34
    bandwidth percent 30
  class class12
    bandwidth percent 20
  class class-default
    bandwidth percent 20
!
policy-map Tunnel3ParentPolicy
  class class-default
    shape average 1000000000
   service-policy Tunnel3ChildPolicy
!
policy-map Tunnel4ChildPolicy
  class class7
    priority
  class class6
    bandwidth remaining percent 20
  class class5
    bandwidth remaining percent 20
  class class4
    bandwidth remaining percent 20
  class class3
    bandwidth remaining percent 10
  class class2
    bandwidth remaining percent 10
  class class1
    bandwidth remaining percent 10
  class class-default
    bandwidth remaining percent 10
!
policy-map Tunnel4ParentPolicy
  class class-default
    shape average 256000
   service-policy Tunnel4ChildPolicy
!
policy-map Tunnel5ParentPolicy
  class class-default
    shape average 128000 512 0
!
!
```

```
!
!
crypto isakmp policy 10
 encr aes
 group 2
 lifetime 7200
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set MyTranSet esp-aes 256 esp-sha-hmac
no crypto ipsec nat-transparency udp-encaps
!
crypto ipsec profile MyIpsecProf
 set transform-set MyTranSet
!
!
buffers small permanent 1024
buffers small max-free 1500
buffers small min-free 500
buffers middle permanent 512
buffers middle max-free 3000
buffers middle min-free 100
buffers big permanent 1000
buffers big max-free 1000
buffers big min-free 300
!
!
interface Tunnel0
 bandwidth 10000000
 ip address 3.0.0.1 255.255.255.0
 ip hello-interval eigrp 10 60
 ip hold-time eigrp 10 180
 tunnel source Loopback0
 tunnel destination 5.0.0.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile MyIpsecProf
 crypto engine slot 4/0 inside
 crypto-engine
     service-policy output Tunnel0ParentPolicy
!
interface Tunnel1
 bandwidth 10000000
 ip address 3.0.1.1 255.255.255.0
 ip hello-interval eigrp 10 60
 ip hold-time eigrp 10 180
 tunnel source Loopback1
 tunnel destination 5.0.1.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile MyIpsecProf
 crypto engine slot 4/0 inside
 crypto-engine
     service-policy output Tunnel1ParentPolicy
!
interface Tunnel2
 bandwidth 10000000
 ip address 3.0.2.1 255.255.255.0
 ip hello-interval eigrp 10 60
 ip hold-time eigrp 10 180
 tunnel source Loopback2
 tunnel destination 5.0.2.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile MyIpsecProf
 crypto engine slot 4/0 inside
 crypto-engine
```

```
     service-policy output Tunnel2ParentPolicy
!
interface Tunnel3
 bandwidth 10000000
 ip address 3.0.3.1 255.255.255.0
 ip hello-interval eigrp 10 60
 ip hold-time eigrp 10 180
 tunnel source Loopback3
 tunnel destination 5.0.3.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile MyIpsecProf
 crypto engine slot 4/0 inside
 crypto-engine
     service-policy output Tunnel3ParentPolicy
!
interface Tunnel4
 bandwidth 10000000
 ip address 3.0.4.1 255.255.255.0
 ip hello-interval eigrp 10 60
 ip hold-time eigrp 10 180
 tunnel source Loopback4
 tunnel destination 5.0.4.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile MyIpsecProf
 crypto engine slot 4/0 inside
 crypto-engine
     service-policy output Tunnel4ParentPolicy
!
interface Tunnel5
 bandwidth 10000000
 ip address 3.0.5.1 255.255.255.0
 ip hello-interval eigrp 10 60
 ip hold-time eigrp 10 180
 tunnel source Loopback5
 tunnel destination 5.0.5.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile MyIpsecProf
 crypto engine slot 4/0 inside
 crypto-engine
     service-policy output Tunnel5ParentPolicy
!
interface Loopback0
 ip address 4.0.0.1 255.255.255.255
!
interface Loopback1
 ip address 4.0.1.1 255.255.255.255
!
interface Loopback2
 ip address 4.0.2.1 255.255.255.255
!
interface Loopback3
 ip address 4.0.3.1 255.255.255.255
!
interface Loopback4
 ip address 4.0.4.1 255.255.255.255
!
interface Loopback5
 ip address 4.0.5.1 255.255.255.255
!
interface TenGigabitEthernet2/1
 description EGRESS INTERFACE
 mtu 9216
 ip address 6.0.0.1 255.255.255.0
 load-interval 30
 shutdown
 mls qos trust dscp
 crypto engine outside
 hold-queue 4096 in
!
interface TenGigabitEthernet2/2
 no ip address
 shutdown
!
interface TenGigabitEthernet2/3
 description INGRESS INTERFACE
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2-7
 switchport mode trunk
 mtu 9216
 load-interval 30
 mls qos trust dscp
 hold-queue 4096 in
!
interface TenGigabitEthernet2/4
 description TO TESTCENTER PORT 2/2 (NOT IN USE)
 mtu 9216
 no ip address
 load-interval 30
 shutdown
!
interface TenGigabitEthernet2/5
 no ip address
 shutdown
!
interface TenGigabitEthernet2/6
 no ip address
 shutdown
!
interface TenGigabitEthernet2/7
 no ip address
 shutdown
!
interface TenGigabitEthernet2/8
 no ip address
 shutdown
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 wrr-queue cos-map 2 1 4
 priority-queue cos-map 1 5 6 7
 rcv-queue cos-map 1 3 4
 mls qos trust dscp
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 wrr-queue cos-map 2 1 4
 priority-queue cos-map 1 5 6 7
 rcv-queue cos-map 1 3 4
 mls qos trust dscp
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
interface GigabitEthernet5/1
 no ip address
 shutdown
!
interface GigabitEthernet5/2
 description LABNET
 ip address 44.0.111.118 255.0.0.0
 media-type rj45
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan2
 mtu 9216
 ip address 1.0.0.1 255.255.255.0
!
interface Vlan3
 mtu 9216
 ip address 1.0.1.1 255.255.255.0
!
interface Vlan4
 mtu 9216
 ip address 1.0.2.1 255.255.255.0
!
interface Vlan5
 mtu 9216
 ip address 1.0.3.1 255.255.255.0
!
interface Vlan6
 mtu 9216
 ip address 1.0.4.1 255.255.255.0
!
interface Vlan7
 mtu 9216
 ip address 1.0.5.1 255.255.255.0
!
router eigrp 10
 network 3.0.0.0
 no auto-summary
 distribute-list T0000 out Tunnel0
 distribute-list T0001 out Tunnel1
 distribute-list T0002 out Tunnel2
 distribute-list T0003 out Tunnel3
 distribute-list T0004 out Tunnel4
 distribute-list T0005 out Tunnel5
 timers active-time 10
 redistribute connected metric 900 100 255 1 1400
!
router ospf 10
 log-adjacency-changes
 summary-address 4.0.0.0 255.0.0.0
 redistribute connected metric 10 subnets
 network 6.0.0.0 0.0.0.255 area 0
 distribute-list 10 out
!
ip default-gateway 44.0.100.1
ip classless
ip route 43.0.0.0 255.0.0.0 44.0.100.1
ip route 223.255.254.53 255.255.255.255 44.0.100.1
!
!
no ip http server
no ip http secure-server
!
!
ip access-list standard T0000
 permit 1.0.0.0 0.0.0.255
ip access-list standard T0001
 permit 1.0.1.0 0.0.0.255
ip access-list standard T0002
 permit 1.0.2.0 0.0.0.255
ip access-list standard T0003
 permit 1.0.3.0 0.0.0.255
ip access-list standard T0004
 permit 1.0.4.0 0.0.0.255
ip access-list standard T0005
 permit 1.0.5.0 0.0.0.255
logging alarm informational
logging 43.0.111.111
access-list 10 permit 4.0.0.0 0.255.255.255
!
!
!
!
no cdp run
!
!
control-plane
!
!
dial-peer cor custom
!
!
!
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password cisco
 login
line vty 5 15
 login
!
exception core-file
mac-address-table aging-time 0
ntp clock-period 17219357
ntp update-calendar
ntp server 223.255.254.53
!
end
```

CHAPTER 7

Configuring IKE Features

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

Note For detailed information on Internet Key Exchange (IKE), see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

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

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

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

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

Overview of IKE

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

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

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

- Eliminates the need to manually specify all the IPsec security parameters in the crypto maps at both peers.
- Allows you to specify a lifetime for the IPsec security association (SA).
- Allows encryption keys to change during IPsec sessions.
- Allows IPsec to provide anti-replay services.
- Permits certification authority (CA) support for a manageable, scalable IPsec implementation.
- Allows dynamic authentication of peers.

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

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

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

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

Configuring Advanced Encryption Standard in an IKE Policy Map

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

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

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

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

Verifying the AES IKE Policy

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

```
Router# show crypto isakmp policy

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

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

For an AES configuration example, see the "Advanced Encryption Standard Configuration Example" section on page 7-22.

Configuring ISAKMP Keyrings

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

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

ISAKMP Keyrings Configuration Guidelines and Restrictions

When configuring ISAKMP keyrings, follow these guidelines and restrictions:

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

Limiting an ISAKMP Profile to a Local Termination Address or Interface

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

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

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

Limiting a Keyring to a Local Termination Address or Interface

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

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

For ISAKMP keyrings configuration examples, see the "ISAKMP Keyrings Configuration Examples" section on page 7-22.

Configuring Certificate to ISAKMP Profile Mapping

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

Certificate to ISAKMP Profile Mapping Configuration Guidelines and Restrictions

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

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

Mapping the Certificate to the ISAKMP Profile

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

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

Verifying the Certificate to ISAKMP Profile Mapping Configuration

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

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

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

Responder Configuration

```
crypto pki certificate map cert_map 10
! The above line is the certificate map definition.
 subject-name co ou = green
! The above line shows that the subject name must have "ou = green."
!
crypto isakmp profile certpro
! The above line shows that this is the ISAKMP profile that will match if the certificate
of the peer matches cert_map (shown on third line below).
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
```

Initiator Configuration

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

Command Output for show crypto pki certificates for the Initiator

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

Command Output for debug crypto isakmp for the Responder

```
Router# debug crypto isakmp

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

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

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

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

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

*Nov  6 19:31:25.046: ISAKMP:(0): processing SA payload. message ID = 0
*Nov  6 19:31:25.046: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.046: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.046: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.046: ISAKMP : Looking for xauth in profile prof2
*Nov  6 19:31:25.046: ISAKMP:(0):Checking ISAKMP transform 1 against priority 1 policy
*Nov  6 19:31:25.046: ISAKMP:      encryption 3DES-CBC
*Nov  6 19:31:25.046: ISAKMP:      hash MD5
*Nov  6 19:31:25.046: ISAKMP:      default group 1
*Nov  6 19:31:25.046: ISAKMP:      auth RSA sig
*Nov  6 19:31:25.050: ISAKMP:      life type in seconds
*Nov  6 19:31:25.050: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
*Nov  6 19:31:25.050: ISAKMP:(0):atts are acceptable. Next payload is 0
*Nov  6 19:31:25.050: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.050: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.050: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.050: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.050: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM2

*Nov  6 19:31:25.050: ISAKMP (0): constructing CERT_REQ for issuer
cn=mscavpn1,ou=isbu,o=cisco
```

```
*Nov  6 19:31:25.054: ISAKMP:(0): sending packet to 14.0.0.2 my_port 500 peer_port 500 (I) 
MM_SA_SETUP
*Nov  6 19:31:25.054: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.054: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM3

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

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

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

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

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

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

*Nov  6 19:31:25.122: ISAKMP:(83727): processing ID payload. message ID = 0
*Nov  6 19:31:25.122: ISAKMP (83727): ID payload
        next-payload : 6
        type         : 3
        USER FQDN    : a@vrf2.com
        protocol     : 17
        port         : 500
        length       : 18
*Nov  6 19:31:25.134: ISAKMP:(83727):: peer matches prof2 profile
*Nov  6 19:31:25.134: ISAKMP:(83727): processing CERT payload. message ID = 0
*Nov  6 19:31:25.134: ISAKMP:(83727): processing a CT_X509_SIGNATURE cert
*Nov  6 19:31:25.142: ISAKMP:(83727): peer's pubkey isn't cached
*Nov  6 19:31:25.158: %CRYPTO-6-IKMP_NO_ID_CERT_USER_FQDN_MATCH: ID of a@vrf2.com (type 3) 
and certificate user fqdn with empty
*Nov  6 19:31:25.158: ISAKMP (83727): adding peer's pubkey to cache
```

```
*Nov  6 19:31:25.158: ISAKMP:(83727): processing SIG payload. message ID = 0
*Nov  6 19:31:25.162: ISAKMP:(83727):SA authentication status:
        authenticated
*Nov  6 19:31:25.162: ISAKMP:(83727):SA has been authenticated with 14.0.0.2
*Nov  6 19:31:25.162: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.162: ISAKMP:(83727):Old State = IKE_R_MM5  New State = IKE_R_MM5

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

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

*Nov  6 19:31:25.238: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500
fvrf (R) QM_IDLE
*Nov  6 19:31:25.238: ISAKMP: set new node -134314170 to QM_IDLE
*Nov  6 19:31:25.242: ISAKMP:(83727): processing HASH payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing SA payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727):Checking IPSec proposal 1
*Nov  6 19:31:25.242: ISAKMP: transform 1, ESP_3DES
*Nov  6 19:31:25.242: ISAKMP:   attributes in transform:
*Nov  6 19:31:25.242: ISAKMP:      encaps is 1 (Tunnel)
*Nov  6 19:31:25.242: ISAKMP:      SA life type in seconds
*Nov  6 19:31:25.242: ISAKMP:      SA life duration (basic) of 3600
*Nov  6 19:31:25.242: ISAKMP:      SA life type in kilobytes
*Nov  6 19:31:25.242: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
*Nov  6 19:31:25.242: ISAKMP:      authenticator is HMAC-SHA
*Nov  6 19:31:25.242: ISAKMP:(83727):atts are acceptable.
*Nov  6 19:31:25.242: ISAKMP:(83727): processing NONCE payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing ID payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing ID payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727):QM Responder gets spi
*Nov  6 19:31:25.242: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_FROM_PEER,
IKE_QM_EXCH
*Nov  6 19:31:25.242: ISAKMP:(83727):Old State = IKE_QM_READY  New State =
IKE_QM_SPI_STARVE
*Nov  6 19:31:25.242: ISAKMP:(83727): Creating IPSec SAs
*Nov  6 19:31:25.246:         inbound SA from 14.0.0.2 to 15.0.0.2 (f/i)  1/714
        (proxy 12.0.0.2 to 13.0.0.2)
*Nov  6 19:31:25.246:         has spi 0x917AD879 and conn_id 0
*Nov  6 19:31:25.246:         lifetime of 3600 seconds
*Nov  6 19:31:25.246:         lifetime of 4608000 kilobytes
*Nov  6 19:31:25.246:         outbound SA from 15.0.0.2 to 14.0.0.2 (f/i) 1/714
        (proxy 13.0.0.2 to 12.0.0.2)
*Nov  6 19:31:25.246:         has spi  0xC54A5A05 and conn_id 0
*Nov  6 19:31:25.246:         lifetime of 3600 seconds
*Nov  6 19:31:25.246:         lifetime of 4608000 kilobytes
```

```
*Nov  6 19:31:25.246:  ISAKMP: Failed to find peer index node to update peer_info_list
*Nov  6 19:31:25.250: ISAKMP:(83727): sending packet to 14.0.0.2 my_port 500 peer_port 500
(R) QM_IDLE
*Nov  6 19:31:25.250: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_INTERNAL,
IKE_GOT_SPI
*Nov  6 19:31:25.250: ISAKMP:(83727):Old State = IKE_QM_SPI_STARVE  New State =
IKE_QM_R_QM2
*Nov  6 19:31:25.270: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500
fvrf (R) QM_IDLE
*Nov  6 19:31:25.274: ISAKMP:(83727):deleting node -134314170 error FALSE reason "QM done
(await)"
*Nov  6 19:31:25.274: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_FROM_PEER,
IKE_QM_EXCH
*Nov  6 19:31:25.274: ISAKMP:(83727):Old State = IKE_QM_R_QM2  New State =
IKE_QM_PHASE2_COMPLETE
*Nov  6 19:32:15.282: ISAKMP:(83727):purging node -134314170
```

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

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

IPv6 Crypto ISAKMP SA

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

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

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

IPv6 Crypto ISAKMP SA
```

Assigning the Group Name to the Peer

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

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

Verifying the Group Name to Peer Assignation Configuration

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

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

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

Initiator Configuration

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

Command Output for debug crypto isakmp for the Responder

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

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

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

For certificate to ISAKMP profile mapping configuration examples, see the “Certificate to ISAKMP Profile Mapping Configuration Examples” section on page 7-22.

Configuring an Encrypted Preshared Key

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

Encrypted Preshared Key Configuration Guidelines and Restrictions

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

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

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

- If you later unconfigure password encryption using the **no password encryption ae**s command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption** command exists, the type 6 passwords will be decrypted as and when required by the application.
- Because no one can “read” the password (configured using the **key config-key password-encryption** command), there is no way that the password can be retrieved from the switch. Existing management stations cannot “know” what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone, meaning that they cannot be loaded onto a switch. Before or after the configurations are loaded onto a switch, the password must be manually added (using the **key config-key password-encryption** command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.
- If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but the following alert message is printed:

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

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

Configuring an Encrypted Preshared Key

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

	Command	Purpose

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

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

Verifying the Encrypted Preshared Key Configuration

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

```
Router(config)# password logging

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

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

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

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

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

For an encrypted preshared key configuration example, see the "Encrypted Preshared Key Configuration Example" section on page 7-23.

Configuring Call Admission Control for IKE

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

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

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

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

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

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

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

Configuring the IKE Security Association Limit

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

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

Configuring a System Resource Limit

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

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

Clearing Call Admission Statistics

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

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

Verifying the Call Admission Control for IKE Configuration

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

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

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

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

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

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

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

For Call Admission Control for IKE configuration examples, see the "Call Admission Control for IKE Configuration Examples" section on page 7-23.

Configuring Dead Peer Detection

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

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

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

DPD is configured using the **crypto isakmp keepalive** command. DPD and Cisco IOS keepalives function on the basis of a timer. If the timer is set for 10 seconds, the switch will send a hello message every 10 seconds (unless, of course, the switch receives a hello message from the peer). The benefit of

Cisco IOS keepalives and periodic DPD is earlier detection of dead peers. However, Cisco IOS keepalives and periodic DPD rely on periodic messages that have to be sent with considerable frequency. The result of sending frequent messages is that the communicating peers must encrypt and decrypt more packets.

DPD and Cisco IOS keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the switch to detect a dead IKE peer, and when the switch detects the dead state, the switch deletes the IPsec and IKE SAs to the peer. If you configure multiple peers, the switch will switch over to the next listed peer for a stateless failover.

DPD Configuration Guidelines and Restrictions

When configuring DPD, follow these guidelines and restrictions:

- When the **crypto isakmp keepalive** command is configured, the Cisco IOS software negotiates the use of Cisco IOS keepalives or DPD, depending on which protocol the peer supports.
- If you do not configure the **periodic** option using the **crypto isakmp keepalive** command, the switch defaults to the **on-demand** approach.
- Before configuring periodic DPD, you should ensure that your IKE peer supports DPD. Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS software in all modes of operation—site-to-site, Easy VPN remote, and Easy VPN server.
- Using periodic DPD potentially allows the switch to detect an unresponsive IKE peer with better response time when compared to on-demand DPD. However, use of periodic DPD incurs extra overhead. When communicating to large numbers of IKE peers, you should consider using on-demand DPD instead.
- When the **crypto isakmp keepalive** command is configured, the Cisco IOS software negotiates the use of Cisco IOS keepalives or DPD, depending on which protocol the peer supports.
- To allow the switch to send DPD messages to the peer, enter the **crypto isakmp keepalive** command in global configuration mode as follows:

  ```
  Router(config)# crypto isakmp keepalive seconds [retries] [on-demand]
  ```

  seconds—Specifies the number of seconds between DPD messages; the range is from 10 to 3600 seconds.

  retries—(Optional) Specifies the number of seconds between DPD retries if the DPD message fails; the range is from 2 to 60 seconds. If unspecified, the default is 2 seconds.

  on-demand—(Optional) Specifies DPD retries are sent on demand. This is the default behavior.

Configuring a Dead Peer Detection Message

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

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

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

Verifying the DPD Configuration

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

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

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

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

For DPD configuration examples, see the "Dead Peer Detection Configuration Examples" section on page 7-24.

Configuring IPsec NAT Transparency

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

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

IPsec NAT Transparency Configuration Guidelines and Restrictions

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

- For non-DMVPN configurations, NAT transparency is supported in both tunnel and transport mode.
- For DMVPN configurations, NAT transparency is only supported in transport mode.

Configuring NAT Transparency

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

Disabling NAT Transparency

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

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

Configuring NAT Keepalives

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

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

The *seconds* value specifies the number of seconds between keepalive packets; the range is between 5 to 3,600 seconds.

For a NAT keepalive configuration example, see the "ISAKMP NAT Keepalive Configuration Example" section on page 7-24.

Verifying the NAT Configuration

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

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

```
Router# show crypto ipsec sa

interface:GigabitEthernet5/0/1
```

```
Crypto map tag:testtag, local addr. 10.2.80.161

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

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

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

inbound ah sas:

inbound pcp sas:

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

outbound ah sas:

outbound pcp sas:
```

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

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

Configuration Examples

This section provides examples of the following configurations:

- Advanced Encryption Standard Configuration Example, page 7-22
- ISAKMP Keyrings Configuration Examples, page 7-22
- Certificate to ISAKMP Profile Mapping Configuration Examples, page 7-22
- Encrypted Preshared Key Configuration Example, page 7-23
- Call Admission Control for IKE Configuration Examples, page 7-23
- Dead Peer Detection Configuration Examples, page 7-24

- ISAKMP NAT Keepalive Configuration Example, page 7-24

Advanced Encryption Standard Configuration Example

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

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

ISAKMP Keyrings Configuration Examples

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

- ISAKMP Profile Bound to a Local Interface Configuration Example, page 7-22
- ISAKMP Keyring Bound to a Local Interface Configuration Example, page 7-22
- ISAKMP Keyring Bound to a Local IP Address Configuration Example, page 7-22

ISAKMP Profile Bound to a Local Interface Configuration Example

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

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

ISAKMP Keyring Bound to a Local Interface Configuration Example

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

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

ISAKMP Keyring Bound to a Local IP Address Configuration Example

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

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

Certificate to ISAKMP Profile Mapping Configuration Examples

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

- Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Configuration Example, page 7-23

- Group Name Assigned to a Peer Associated with an ISAKMP Profile Configuration Example, page 7-23

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

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

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

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

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

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

client configuration group some_group
```

Encrypted Preshared Key Configuration Example

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

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

Call Admission Control for IKE Configuration Examples

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

- IKE Security Association Limit Configuration Example, page 7-24
- System Resource Limit Configuration Example, page 7-24

IKE Security Association Limit Configuration Example

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

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

System Resource Limit Configuration Example

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

```
Router(config)# call admission limit 5000
```

Dead Peer Detection Configuration Examples

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

- On-Demand DPD Configuration Example, page 7-24
- Periodic DPD Configuration Example, page 7-24

On-Demand DPD Configuration Example

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

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

Periodic DPD Configuration Example

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

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

ISAKMP NAT Keepalive Configuration Example

The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key 1234 address 56.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 56.0.0.1
 set transform-set t2
 match address 101
```

CHAPTER 8

Configuring Enhanced IPsec Features

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

- Overview of Enhanced IPsec Features, page 8-2
- Configuring Advanced Encryption Standard in a Transform Set, page 8-2
- Configuring Reverse Route Injection, page 8-3
- Configuring the IPsec Anti-Replay Window Size, page 8-5
- Configuring an IPsec Preferred Peer, page 8-8
- Configuring IPsec Security Association Idle Timers, page 8-11
- Configuring Distinguished Name-Based Crypto Maps, page 8-13
- Configuring Platform ACLs for Tunnel Interfaces, page 8-15
- Configuring Sequenced Crypto ACLs, page 8-16
- Understanding IPv6 IPsec Support in the VSPA, page 8-17
- Configuration Examples, page 8-19

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

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

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

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

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

Overview of Enhanced IPsec Features

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF). It provides security for transmission of sensitive information over unprotected networks, such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers.

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

Configuring Advanced Encryption Standard in a Transform Set

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

To configure the AES encryption algorithm within a transform set, perform this task beginning in global configuration mode:

Command	Purpose

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

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

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

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

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

Verifying the AES Transform Set

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

```
Router# show crypto ipsec transform-set

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

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

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

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

Configuring Reverse Route Injection

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

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

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

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

RRI Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring RRI:

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

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

Configuring RRI Under a Static Crypto Map

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

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

Configuring RRI Under a Dynamic Crypto Map

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

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

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

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

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

Configuring the IPsec Anti-Replay Window Size

Cisco IPsec authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. (Security association (SA) anti-replay is a security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks.) The decryptor checks off the sequence numbers that it has seen before. The encryptor assigns sequence numbers in an increasing order. The decryptor remembers the value (X) of the highest sequence number that it has already seen. N is the window size of the decryptor. Any packet with a sequence number less than X minus N is discarded. Currently, N is set at 64.

At times, the 64-packet window size is not sufficient. For example, Cisco quality of service (QoS) gives priority to high-priority packets, which could cause some low-priority packets to be discarded even though they are not replayed packets. The IPsec anti-replay window size feature allows you to expand the window size so that sequence number information can be kept for more than 64 packets.

Expanding the IPsec Anti-Replay Window Size Globally

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Disabling the IPsec Anti-Replay Checking

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

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

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

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

Configuring an IPsec Preferred Peer

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

This feature includes the following capabilities:

- Default peer configuration

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

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

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

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

- IPsec idle timer with default peer configuration

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

  IPsec SA idle timers increase the availability of resources by deleting SAs associated with idle peers. Because IPsec SA idle timers prevent the wasting of resources by idle peers, more resources are available to create new SAs when required. (If IPsec SA idle timers are not configured, only the global lifetimes for IPsec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.)

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

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

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

IPsec Preferred Peer Configuration Guidelines and Restrictions

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

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

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

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

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

Configuring a Default Peer

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

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

Configuring the IPsec Idle Timer with a Default Peer

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

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

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

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

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

Configuring IPsec Security Association Idle Timers

When a switch running Cisco IOS software creates an IPsec SA for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the switch could be prevented from creating new SAs with other peers. The IPsec security association idle timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. The idle timers can be configured either globally, on a per-crypto map basis, or through an ISAKMP profile. The benefits of this feature include the following:

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

IPsec Security Association Idle Timer Configuration Guidelines

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

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

Configuring the IPsec SA Idle Timer Globally

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

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

Configuring the IPsec SA Idle Timer per Crypto Map

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

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

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

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

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

Configuring Distinguished Name-Based Crypto Maps

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

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

Distinguished Name-Based Crypto Map Configuration Guidelines and Restrictions

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

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

Configuring a DN-Based Crypto Map

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

	Command	Purpose

Step 1	`Router(config)#` **`crypto isakmp policy`** *`priority`*
	`...`
	`Router(config-isakmp)#` **`exit`**
		Defines an ISAKMP policy and enters ISAKMP policy configuration mode.
		• *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
		Creates an ISAKMP policy at each peer.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

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

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

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

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

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

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

Configuring Platform ACLs for Tunnel Interfaces

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

Platform ACL on Tunnel Interfaces Configuration Guidelines and Restrictions

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

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

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

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

Configuring Sequenced Crypto ACLs

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

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

Configuring Deny Policy Enhancements for Crypto ACLs

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

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

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

Deny Policy Enhancements for ACLs Configuration Guidelines and Restrictions

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

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

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

Understanding IPv6 IPsec Support in the VSPA

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

This section contains the following topics:

- IPv6 IPsec Configuration Guidelines and Restrictions, page 8-17
- Tunnel Source Address Selection, page 8-18
- Configuring IPv6 IPsec, page 8-18

IPv6 IPsec Configuration Guidelines and Restrictions

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

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

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

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

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

Tunnel Source Address Selection

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

Configuring IPv6 IPsec

To configure IPsec for IPv6, perform this task:

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

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

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

Configuration Examples

This section provides examples of the following configurations:

- Advanced Encryption Standard Configuration Example, page 8-20
- Reverse Route Injection Configuration Examples, page 8-20
- IPsec Anti-Replay Window Size Configuration Examples, page 8-21
- IPsec Preferred Peer Configuration Examples, page 8-23
- IPsec Security Association Idle Timer Configuration Examples, page 8-24
- Distinguished Name-Based Crypto Maps Configuration Example, page 8-24
- Platform ACL Configuration Example, page 8-25
- Deny Policy Enhancements for ACLs Configuration Example, page 8-26
- IPv6 IPsec Configuration Example, page 8-26

Advanced Encryption Standard Configuration Example

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

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

Reverse Route Injection Configuration Examples

The following examples show how to configure RRI:

- RRI Under a Static Crypto Map Configuration Example, page 8-20
- RRI Under a Dynamic Crypto Map Configuration Example, page 8-20
- RRI with Existing ACLs Configuration Example, page 8-20
- RRI for Two Routes Configuration Example, page 8-21
- RRI via a User-Defined Hop Configuration Example, page 8-21

RRI Under a Static Crypto Map Configuration Example

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

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

RRI Under a Dynamic Crypto Map Configuration Example

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

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

RRI with Existing ACLs Configuration Example

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

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

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

RRI for Two Routes Configuration Example

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

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

RRI via a User-Defined Hop Configuration Example

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

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

IPsec Anti-Replay Window Size Configuration Examples

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

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

IPsec Anti-Replay Window Global Configuration Example

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

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN-Gateway1
!
boot-start-marker
boot-end-marker
!
clock timezone EST 0
no aaa new-model
ip subnet-zero
!
ip audit po max-events 100
no ftp-server write-enable
!
crypto isakmp policy 10
 authentication pre-share
 crypto isakmp key cisco123
 address 192.165.201.2
!
crypto ipsec security-association replay window-size 1024
!
crypto ipsec transform-set basic esp-des esp-md5-hmac
!
crypto map mymap 10 ipsec-isakmp
 set peer 192.165.201.2
 set transform-set basic
 match address 101
!
interface Ethernet0/0
```

```
 ip address 192.168.1.1 255.255.255.0
!
interface Serial1/0
 ip address 192.165.200.2 255.255.255.252
 serial restart-delay 0
 crypto map mymap
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.165.200.1
no ip http server
no ip http secure-server
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 172.16.2.0 0.0.0.255
!access-list 101 remark Crypto ACL
!
control-plane
!
line con 0
line aux 0
line vty 0 4
end
```

IPsec Anti-Replay Window per Crypto Map Configuration Example

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

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname dr_whoovie
!
enable secret 5 $1$KxKv$cbqKsZtQTLJLGPN.tErFZ1
enable password ww
!
ip subnet-zero
cns event-service server
crypto isakmp policy 1
 authentication pre-share
 crypto isakmp key cisco170
 address 172.150.150.2
 crypto isakmp key cisco180
 address 172.150.150.3
 crypto isakmp key cisco190
 address 172.150.150.4
crypto ipsec transform-set 170cisco esp-des esp-md5-hmac
crypto ipsec transform-set 180cisco esp-des esp-md5-hmac
crypto ipsec transform-set 190cisco esp-des esp-md5-hmac
crypto map ETH0 17 ipsec-isakmp
 set peer 172.150.150.2
 set security-association replay disable
 set transform-set 170cisco
 match address 170
crypto map ETH0 18 ipsec-isakmp
 set peer 150.150.150.3
 set transform-set 180cisco
 match address 180
crypto map ETH0 19 ipsec-isakmp
 set peer 150.150.150.4
 set transform-set 190cisco
 match address 190
!
interface Ethernet0
 ip address 172.150.150.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no mop enabled
 crypto map ETH0
!
interface Serial0
 ip address 172.160.160.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
ip classless
ip route 172.170.170.0 255.255.255.0 172.150.150.2
ip route 172.180.180.0 255.255.255.0 172.150.150.3
ip route 172.190.190.0 255.255.255.0 172.150.150.4
no ip http server
!
access-list 170 permit ip 172.160.160.0 0.0.0.255 172.170.170.0 0.0.0.255
access-list 180 permit ip 172.160.160.0 0.0.0.255 172.180.180.0 0.0.0.255
access-list 190 permit ip 172.160.160.0 0.0.0.255 172.190.190.0 0.0.0.255
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
end
```

IPsec Preferred Peer Configuration Examples

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

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

Default Peer Configuration Example

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

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

IPsec Idle Timer with Default Peer Configuration Example

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

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

IPsec Security Association Idle Timer Configuration Examples

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

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

IPsec SA Idle Timer Global Configuration Example

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

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

IPsec SA Idle Timer per Crypto Map Configuration Example

The following example configures the IPsec SA idle timer for the crypto map named test to drop SAs for inactive peers after 600 seconds:

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

Distinguished Name-Based Crypto Maps Configuration Example

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

```
! DN based crypto maps require you to configure an IKE policy at each peer.
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
 crypto isakmp key 1234567890 address 171.69.224.33
!
!The following is an IPsec crypto map (part of IPsec configuration). It can be used only
! by peers that have been authenticated by DN and if the certificate belongs to BigBiz.
crypto map map-to-bigbiz 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
 match address 124
 identity to-bigbiz
!
crypto identity to-bigbiz
dn ou=BigBiz
!
!
! This crypto map can be used only by peers that have been authenticated by hostname
!and if the certificate belongs to little.com.
crypto map map-to-little-com 10 ipsec-isakmp
 set peer 172.21.115.119
 set transform-set my-transformset
 match address 125
 identity to-little-com
!
crypto identity to-little-com
fqdn little.com
!
```

Platform ACL Configuration Example

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

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

Deny Policy Enhancements for ACLs Configuration Example

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

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

IPv6 IPsec Configuration Example

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

Hub Configuration

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

Spoke Configuration

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

CHAPTER 9

Configuring PKI

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

- Overview of PKI, page 9-2
- Understanding Multiple RSA Key Pairs, page 9-3
- Understanding Protected Private Key Storage, page 9-5
- Understanding the Trustpoint CA, page 9-8
- Understanding Query Mode Definition Per Trustpoint, page 9-12
- Understanding Direct HTTP Enroll with CA Servers (Reenroll Using Existing Certificates), page 9-15
- Understanding Manual Certificate Enrollment (TFTP and Cut-and-Paste), page 9-21
- Understanding Certificate Autoenrollment, page 9-26
- Understanding Key Rollover for Certificate Renewal, page 9-30
- Understanding PKI: Query Multiple Servers During Certificate Revocation Check, page 9-36
- Understanding the Online Certificate Status Protocol, page 9-37
- Understanding Certificate Security Attribute-Based Access Control, page 9-41
- Understanding PKI AAA Authorization Using the Entire Subject Name, page 9-45
- Understanding Source Interface Selection for Outgoing Traffic with Certificate Authority, page 9-47
- Understanding Persistent Self-Signed Certificates, page 9-49
- Configuration Examples, page 9-53

Note The procedures in this chapter assume you have some familiarity with PKI configuration concepts. For detailed information about PKI configuration concepts, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

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

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

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

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

Overview of PKI

Cisco IOS public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPsec), secure shell (SSH), and secure socket layer (SSL).

A PKI is composed of the following entities:

- Peers communicating on a secure network
- At least one certificate authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryption keys that are used for secure communications, and the signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol (LDAP) or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the secured communications is enrolled in the PKI, a process where the entity generates a Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Configuring PKI involves the following tasks:

- Deploying Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certificate authority (CA) to obtain a certificate and enroll in a PKI.
- Configuring authorization and revocation of certificates within a PKI. After a certificate is validated as a properly signed certificate, it is authorized using methods such as certificate maps, PKI-AAA, or a certificate-based access control list (ACL). The revocation status is checked by the issuing certificate authority (CA) to ensure that the certificate has not been revoked.

- Configuring certificate enrollment, which is the process of obtaining a certificate from a certificate authority (CA). Certificate enrollment occurs between the end host requesting the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA. Various methods are available for certificate enrollment.
- Storing public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates. These credentials can be stored in the default location on the router, which is NVRAM, or other locations.

Understanding Multiple RSA Key Pairs

The multiple RSA key pair support feature allows you to configure a Catalyst 6500 Series switch to have multiple Rivest, Shamir, and Adelman (RSA) key pairs. The Cisco IOS software can maintain a different key pair for each identity certificate.

Before this feature, Cisco IOS public key infrastructure (PKI) configurations allowed either one general-purpose key pair or a set of special-purpose key pairs (an encryption and a signing key pair). The scenarios in which the key pairs were deployed often required configurations that required the switch to enroll with multiple certificate servers because each server has an independent policy and may also have different requirements regarding general-purpose versus special-purpose certificates or key length. With this feature, a user can configure different key pairs for each certification authority (CA) with which the switch enrolls and can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus special-usage keys.

Multiple RSA Key Pairs Configuration Guidelines and Restrictions

When configuring multiple RSA key pair support, follow these guidelines and restrictions:

- We recommend that Secure Socket Layer (SSL) or other PKI clients do not attempt to enroll with the same CA multiple times.
- Internet Key Exchange (IKE) will not work for any identity that is configured to use a named key pair. If an IKE peer requests a certificate from a PKI trustpoint that is using multiple key support, the initial portion of the exchange will work, that is, the correct certificate will be sent in the certificate response; however, the named keypair will not be used and the IKE negotiation will fail.
- Whenever you regenerate a key pair, you must always reenroll the certificate identities with that key pair.

Configuring an RSA Key Pair

To configure an RSA key pair, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto key generate rsa [usage-keys | general-keys] [modulus modulus-size] [key-pair-label]`	Generates RSA key pairs. • **usage-keys**—(Optional) Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair. • **general-keys**—(Optional) Specifies that the general-purpose key pair should be generated. • *key-pair-label*—(Optional) Specifies the name of the key pair that the switch will use. (If this argument is enabled, you must specify either **usage-keys** or **general-keys**.) • **modulus** *modulus-size*—(Optional) Specifies the modulus for generating the RSA keys. The range is 384 to 2048 bits, and the modulus must be a multiple of 64. The default is 1024.
Step 2	`Router(config)# crypto pki trustpoint name`	Declares the CA that the switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 3	`Router(ca-trustpoint)# rsakeypair key-label [key-size [encryption-key-size]]`	Specifies which key pair to associate with the certificate. • *key-label*—The name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) The size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) The size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.)

Removing RSA Key Pair Settings

To delete a specified RSA key pair or all RSA key pairs that have been generated by your switch, enter the **crypto key zeroize rsa** command in global configuration mode as follows:

```
Router(config)# crypto key zeroize rsa [key-pair-label]
```

key-pair-label specifies the name of the key pair to be deleted. If the *key-pair-label* argument is used, you will delete only the specified RSA key pair. If no argument is used, you will delete all the RSA key pairs from your switch.

Verifying RSA Key Information

To verify RSA key information, use at least one of the privileged EXEC commands used in the examples.

To display your switch's RSA public keys, use the **show crypto key mypubkey rsa** command:

```
Router# show crypto key mypubkey rsa
% Key pair was generated at: 06:07:50 UTC Jan 13 1996
Key name: myswitch.example.com
 Usage: Encryption Key
 Key Data:
  00302017 4A7D385B 1234EF29 335FC973 2DD50A37 C4F4B0FD 9DADE748 429618D5
  18242BA3 2EDFBDD3 4296142A DDF7D3D8 08407685 2F2190A0 0B43F1BD 9A8A26DB
  07953829 791FCDE9 A98420F0 6A82045B 90288A26 DBC64468 7789F76E EE21
```

To display a list of all the RSA public keys stored on your switch (including the public keys of peers that have sent your switch their certificates during peer authentication for IPsec), or to display details of a particular RSA public key stored on your switch, use the **show crypto key pubkey-chain rsa** command:

```
Router# show crypto key pubkey-chain rsa
Codes: M - Manually Configured, C - Extracted from certificate
Code  Usage        IP-address     Name
M     Signature    10.0.0.1       myrouter.example.com
M     Encryption   10.0.0.1       myrouter.example.com
C     Signature    172.16.0.1     routerA.example.com
C     Encryption   172.16.0.1     routerA.example.com
C     General      192.168.10.3   routerB.domain1.com
```

For complete configuration information for Multiple RSA Key Pair Support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftmltkey.html

For an RSA key pair configuration example, see the "Multiple RSA Key Pairs Configuration Example" section on page 9-53.

Understanding Protected Private Key Storage

The protected private key storage feature allows a user to encrypt and lock the RSA private keys that are used on a Catalyst 6500 Series switch, which prevents unauthorized use of the private keys.

Protected Private Key Storage Configuration Guidelines and Restrictions

When configuring protected private key storage, follow these guidelines and restrictions:

- An encrypted key is not effective after the switch boots up until you manually unlock the key (using the **crypto key unlock rsa** command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP Security (IPsec), Secure Shell (SSH) and Secure Socket Layer (SSL); that is, management of the switch over a secure channel may not be possible until the necessary key pair is unlocked.
- If a passphrase is lost, you must regenerate the key, enroll with the CA server again, and obtain a new certificate. A lost passphrase cannot be recovered.
- If you want to change a passphrase, you must decrypt the key with the current passphrase using the **crypto key decrypt rsa** command and encrypt the key once more to specify the new passphrase.

Configuring Private Keys

To encrypt, decrypt, lock, and unlock private keys, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto key encrypt [write] rsa [name key-name] passphrase passphrase`	Encrypts the RSA keys. After this command is entered, the switch can continue to use the key; the key remains unlocked. • **write**—(Optional) Switch configuration is immediately written to NVRAM. If the **write** keyword is not specified, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the switch is reloaded. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be encrypted. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to encrypt the RSA key. To access the RSA key pair, the passphrase must be specified.
Step 2	`Router(config)# exit`	Exits global configuration mode.
Step 3	`Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is encrypted (protected) and unlocked.

	Command	Purpose
Step 4	`Router# crypto key lock rsa [name key-name] passphrase passphrase`	(Optional) Locks the encrypted private key on a running switch. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be locked. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to lock the RSA key. To access the RSA key pair, the passphrase must be specified. **Note** After the key is locked, it cannot be used to authenticate the switch to a peer device. This behavior disables any IPsec or SSL connections that use the locked key. Any existing IPsec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled.
Step 5	`Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is protected and locked. The output will also show failed connection attempts by applications such as IKE, SSH, and SSL.
Step 6	`Router# crypto key unlock rsa [name key-name] passphrase passphrase`	(Optional) Unlocks the private key. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be unlocked. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to unlock the RSA key. To access the RSA key pair, the passphrase must be specified. **Note** After this command is entered, you can continue to establish IKE tunnels.

	Command	Purpose
Step 7	`Router#` **`configure terminal`**	Enters global configuration mode.

Step 8	`Router(config)#` **`crypto key decrypt`** `[`**`write`**`]` **`rsa`** `[`**`name`**
	`key-name``]` **`passphrase`** *`passphrase`*
		(Optional) Deletes the encrypted key and leaves only the unencrypted key.
		• **write**—(Optional) Unencrypted key is immediately written to NVRAM. If the **write** keyword is not specified, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the switch is reloaded.
		• **name** *key-name*—(Optional) Name of the RSA key pair that is to be deleted. If a key name is not specified, the default key name, switchname.domainname, is used.
		• **passphrase** *passphrase*—Passphrase that is used to delete the RSA key. To access the RSA key pair, the passphrase must be specified.

Verifying the Protected and Locked Private Keys

To verify that the key is protected (encrypted) and locked, enter the **show crypto key mypubkey rsa** command:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki1-72a.cisco.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

For complete configuration information for protected private key storage, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_ppkey.html

For protected private key configuration examples, see the "Protected Private Key Storage Configuration Examples" section on page 9-53.

Understanding the Trustpoint CA

The **crypto pki trustpoint** command allows you to declare the certificate authority (CA) that your switch should use and to specify characteristics for the CA.

The **crypto pki trustpoint** command combines and replaces the functionality of the existing **crypto ca identity** command and the **crypto ca trusted-root** command. Although both of these existing commands allow you to declare the certification authority (CA) that your switch should use, only the **crypto ca identity** command supports enrollment (the requesting of a switch certificate from a CA).

Trustpoint CA Configuration Guidelines and Restrictions

When configuring a trustpoint CA, follow these guidelines and restrictions:

- After the trustpoint CA has been configured, you can obtain the certificate of the CA by using the **crypto pki authenticate** command or you can specify that certificates should not be stored locally but retrieved from a CA trustpoint by using the **crypto pki certificate query** command.
- Normally, certain certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. To save NVRAM space, you can use the **crypto pki certificate query** command to put the switch into query mode, preventing certificates from being stored locally; instead, they are retrieved from a specified CA trustpoint when needed. This will save NVRAM space but could result in a slight performance impact.

Configuring a Trustpoint CA

To declare the CA that your switch should use and specify characteristics for the trustpoint CA, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use. Enabling this command puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.

	Command	Purpose
Step 2	Router(ca-trustpoint)# **enrollment** [[**mode ra**] | [**retry period** *minutes*] | [**retry count** *number*] | [**url** *url*]]	Specifies enrollment parameters for your CA. • **mode ra**—(Optional) Specifies registration authority (RA) mode if your CA system provides a RA. RA mode is turned off until you enable the **mode ra** keyword. • *minutes*—(Optional) Specifies the wait period between certificate request retries. The default is 1 minute between retries. (Specify from 1 to 60 minutes.) • *number*—(Optional) Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. The default is 10 retries. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.
	Router(ca-trustpoint)# **root tftp** *server-hostname filename*	Obtains the CA via TFTP. • *server-hostname*—Name for the server that will store the trustpoint CA • *filename*—Name for the file that will store the trustpoint CA.
Step 3	Router(ca-trustpoint)# **enrollment http-proxy** *host-name port-num*	Obtains the CA via HTTP through the proxy server. • *host-name*—Name of the proxy server used to get the CA. • *port-num*—Port number used to access the CA. **Note** This command can be used in conjunction only with the **enrollment** command.
Step 4	Router(ca-trustpoint)# **primary** *name*	(Optional) Assigns a specified trustpoint as the primary trustpoint of the switch. • *name*—Name of the primary trustpoint of the switch.

	Command	Purpose
Step 5	`Router(ca-trustpoint)#` **`crl`** `{`**`query`** *`url`* `|` **`optional`**`}`	(Optional) Queries the certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked. • *url* —Lightweight Directory Access Protocol (LDAP) URL published by the certificate authority (CA) server is specified to query the CRL; for example, ldap://another_server. • **optional**—CRL verification is optional. **Note** If the **query** *url* option is not enabled, the switch will check the certificate distribution point (CDP) that is embedded in the certificate.
Step 6	`Router(ca-trustpoint)#` **`default`** *`command-name`*	(Optional) Sets the value of ca-trustpoint configuration mode to its default. • *command-nam*e—pki-trustpoint configuration subcommand. Default is off.
Step 7	`Router(ca-trustpoint)#` **`exit`**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 8	`Router(config)#` **`crypto pki authenticate`** *`name`*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 9	`Router(config)#` **`crypto pki trustpoint`** *`name`*	Reenters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 10	`Router(ca-trustpoint)#` **`crypto pki certificate query`**	(Optional) Turns on query mode per specified trustpoint, causing certificates not to be stored locally.

Verifying a Trustpoint CA

To verify information about your certificate, the certificate of the CA, and registration authority (RA) certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

CA Certificate

  Status: Available

  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F

  Key Usage: Not Set

RA Signature Certificate

  Status: Available

  Certificate Serial Number: 34BCF8A0
  Key Usage: Signature


RA KeyEncipher Certificate

  Status: Available

  Certificate Serial Number: 34BCF89F

  Key Usage: Encryption
```

To display the trustpoints that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:
Subject Name:
CN = ACSWireless Certificate Manager
O = cisco.com
C = US
Serial Number:01
Certificate configured.
CEP URL:http://ACSWireless
CRL query url:ldap://ACSWireless
```

For complete configuration information for the trustpoint CA, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/fttrust.html

For a trustpoint CA configuration example, see the "Trustpoint CA Configuration Example" section on page 9-54.

Understanding Query Mode Definition Per Trustpoint

Certificates contain public key information and are signed by certificate authority (CA) as proof of identity. Normally, all certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. The query mode definition per trustpoint feature allows you to define a query for a specific trustpoint so that the certificates associated with that specific trustpoint can be stored on a remote server.

This feature is especially useful for environments where multiple trustpoints are configured on a switch because it allows you more control over use of the trustpoint. Query mode can be activated on specific trustpoints rather than on all of the trustpoints on a switch.

Query Mode Definition Per Trustpoint Configuration Guidelines and Restrictions

When configuring query mode definition per trustpoint, follow these guidelines and restrictions:

- Normally, certain certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. To save NVRAM space, you can use the **query certificate** command to prevent certificates from being stored locally; instead, they are retrieved from a remote server, such as a CA or LDAP server, during startup. This will save NVRAM space but could result in a slight performance impact.

- Certificates associated with a specified trustpoint will not be written into NVRAM and the certificate query will be attempted during the next reload of the switch.
- When the global **crypto pki certificate query** command is used, the query certificate will be added to all trustpoints on the switch. When the **no crypto pki certificate query** command is used, any previous query certificate configuration will be removed from all trustpoints and any query in progress will be halted and the feature disabled.

Configuring Query Mode Definition Per Trustpoint CA

To configure a trustpoint CA and initiate query mode for the trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoin**t *name*	Declares the CA that your switch should use. Enabling this command puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	Router(ca-trustpoint)# **enrollment** [[**mode ra**] | [**retry period** *minutes*] | [**retry count** *number*] | [**url** *url*]]	Specifies enrollment parameters for your CA. • **mode ra**—(Optional) Specifies registration authority (RA) mode if your CA system provides a RA. RA mode is turned off until you enable the **mode ra** keyword. • *minutes*—(Optional) Specifies the wait period between certificate request retries. The default is 1 minute between retries. (Specify from 1 to 60 minutes.) • *number*—(Optional) Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. The default is 10 retries. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.
Step 3	Router(ca-trustpoint)# **enrollment http-proxy** *host-name port-num*	(Optional) Obtains the CA via HTTP through the proxy server. • *host-name*—Name of the proxy server used to get the CA. • *port-num*—Port number used to access the CA. **Note** This command can be used in conjunction only with the **enrollment** command.

	Command	Purpose
Step 4	`Router(ca-trustpoint)#` **`crl query`** *`url`*	(Optional) Specifies the URL for the CA server if the CA server supports query mode through LDAP. • *url* —LDAP URL published by the certificate authority (CA) server.
Step 5	`Router(ca-trustpoint)#` **`query certificate`**	Turns on query mode per specified trustpoint, causing certificates not to be stored locally and to be retrieved from a remote server.
Step 6	`Router(ca-trustpoint)#` **`exit`**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 7	`Router(config)#` **`crypto pki authenticate`** *`name`*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 8	`Router(config)#` **`crypto key generate rsa`**	(Optional) Generates RSA key pairs.
Step 9	`Router(config)#` **`crypto pki enroll`** *`trustpoint-name`*	(Optional) Obtains switch certificate. • *trustpoint-name*—Name of the CA. Enter the *name* value entered in Step 1.

Verifying Query Mode Definition Per Trustpoint CA

For query mode to operate correctly during the next reload, the certificates must be associated with the trustpoint. Use the **show crypto pki certificates** command to verify that each of the trustpoints has the needed certificates before storing the configuration and reloading the switch:

```
Router# show crypto pki certificates

Trustpoint yni:

  Issuing CA certificate pending:

    Subject Name:

     cn=nsca-r1 Cert Manager,ou=pki,o=cisco.com,c=US

    Fingerprint: C21514AC 12815946 09F635ED FBB6CF31

  Router certificate pending:

    Subject Name:

     hostname=trance.cisco.com,o=cisco.com

  Next query attempt:

    52 seconds
```

For complete configuration information for Query Mode Definition Per Trustpoint, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_qerym.html

For a query mode definition per trustpoint configuration example, see the "Query Mode Definition Per Trustpoint Configuration Example" section on page 9-54.

Understanding Direct HTTP Enroll with CA Servers (Reenroll Using Existing Certificates)

The direct HTTP enroll with CA servers feature allows users to bypass the registration authority (RA) when enrolling with a certification authority (CA) by configuring an enrollment profile. HTTP enrollment requests can be sent directly to the CA server.

The reenroll using existing certificates functionality allows a switch that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS certificate server so the enrollment request is automatically granted.

Direct HTTP Enroll with CA Servers Configuration Guidelines and Restrictions

When configuring direct HTTP enroll with CA servers, follow these guidelines and restrictions:

- The CA certificate and switch certificates must be returned in the privacy enhanced mail (PEM) format.
- If an enrollment profile is specified, an enrollment URL can not be specified in the trustpoint configuration.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.
- The newly created trustpoint can only be used one time (which occurs when the switch is enrolled with the Cisco IOS CA). After the initial enrollment is successfully completed, the credential information will be deleted from the enrollment profile.
- The Cisco IOS certificate server will automatically grant only the requests from clients who were already enrolled with the non-Cisco IOS CA. All other requests must be manually granted unless the server is set to be in auto grant mode (using the **grant automatic** command).
- To configure direct HTTP enroll with CA servers, you must perform the following steps:
  - Either configure a certificate enrollment profile for the client switch (see the "Configuring an Enrollment Profile for a Client Switch" section on page 9-16) or configure an enrollment profile for a client switch that is already enrolled with a third-party vendor (see the "Configuring an Enrollment Profile for a Client Switch Enrolled with a Third-Party Vendor CA" section on page 9-18).
  - Configure the CA certificate server to accept enrollment requests only from clients who are already enrolled with the third-party vendor CA trustpoint (see the "Configuring the CA to Accept Enrollment Requests from Clients of a Third-Party Vendor CA" section on page 9-19).

Configuring an Enrollment Profile for a Client Switch

To configure a certificate enrollment profile, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the trustpoint a given name and enters ca-trustpoint configuration mode. • *name*—Name of the CA trustpoint.
Step 2	Router(ca-trustpoint)# **enrollment profile** *label*	Specifies that an enrollment profile can be used for certificate authentication and enrollment. • *label*—Name for the enrollment profile.
Step 3	Router(ca-trustpoint)# **exit**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 4	Router(config)# **crypto pki profile enrollment** *label*	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.
Step 5	Router(ca-profile-enroll)# **authentication url** *url*	(Optional) Specifies the URL of the CA server to which to send certificate authentication requests. • *url*—URL of the CA server to which your switch should send authentication requests. If using HTTP, the URL should read "http://CA_name," where CA_name is the host Domain Name System (DNS) name or IP address of the CA. If using TFTP, the URL should read "tftp://certserver/file_specification." (If the URL does not include a file specification, the fully qualified domain name (FQDN) of the switch will be used.
	Router(ca-profile-enroll)# **authentication terminal**	(Optional) Specifies manual cut-and-paste certificate authentication.
Step 6	Router(ca-profile-enroll)# **authentication command** *http-command*	(Optional) Sends the HTTP request to the CA for authentication. • *http-command*—HTTP request to be sent to the CA server. This command should be entered after the **authentication url** command has been entered.
Step 7	Router(ca-profile-enroll)# **enrollment url** *url* or	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. • *url*—URL of the CA server.
	Router(ca-profile-enroll)# **enrollment terminal**	Specifies manual cut-and-paste certificate enrollment.

	Command	Purpose
Step 8	`Router(ca-profile-enroll)# **enrollment command** *http-command*`	(Optional) Sends the HTTP command to the CA for enrollment. • *http-command*—HTTP command to be sent to the CA server.
Step 9	`Router(ca-profile-enroll)# **parameter** *number* {**value** *value* | **prompt** *string*}`	(Optional) Specifies parameters for an enrollment profile. • *number*—Number identifying the parameter. Valid values range from 1 to 8. • **value** *value*—To be used if the parameter has a constant value. • **prompt** *string*—To be used if the parameter is supplied after the **crypto pki authenticate** command or the **crypto pki enroll** command has been entered. **Note** The value of the *string* argument does not have an effect on the value that is used by the switch. This command can be used multiple times to specify multiple values.
Step 10	`Router(ca-profile-enroll config)# **exit**`	Exits ca-profile-enroll configuration mode and enters global configuration mode.
Step 11	`Router(config)# **exit**`	Exits global configuration mode and enters Privileged EXEC mode.
Step 12	`Router# **show crypto pki certificates**`	(Optional) Verifies information about your certificate, the certificate of the CA, and RA certificates.
Step 13	`Router# **show crypto pki trustpoints**`	(Optional) Displays the trustpoints that are configured in the switch.

In configuring the direct HTTP enrollment profile, you can use the **parameter** command within an enrollment profile to provide predefined or console-input parameters to the **authentication command** command or the **enrollment command** command.

When you enter the **parameter** *number* command, a macroinstruction (macro) is created and named $P*number* (for example, $P1). If the **value** keyword is specified, the *value* argument is assigned to the macro. If the prompt keyword is specified, when the switch executes the **authentication command** command or the **enrollment command** command, the console will display the *string* argument as a prompt for user input. You can then enter a value to be assigned to the macro.

In addition to user-defined macros, three predefined macros are available:

- $REQ—The Certification Request Standard (PKCS #10) message to request certification of a key
- $FQDN—The FQDN of the switch
- $HOST—The hostname of the switch

This example shows how to use one predefined and three user-defined macros:

```
Router(config)# crypto ca profile enrollment E
Router(ca-profile-enroll)# authentication url http://ca-server.example.com
Router(ca-profile-enroll)# authentication command GET $P1
Router(ca-profile-enroll)# enrollment url
Router(ca-profile-enroll)# enrollment command ^C
POST action=getServerCert
&pkcs10Request=$REQ
&reference_number=$P2
&authcode=P3
&retrievedAs=rawDER
^C
Router(ca-profile-enroll)# parameter 1 value cacert.crt
Router(ca-profile-enroll)# parameter 2 prompt Enter the Reference Number:
Router(ca-profile-enroll)# parameter 3 prompt Enter the Auth Code:
```

Before the HTTP authentication and enrollment commands are posted by the switch to the CA, the console will prompt for any required user input, and macro values will be substituted for the macro names in the posted commands.

For an example of how to configure an enrollment profile for direct HTTP enrollment with a CA server, see the "Enrollment Profile for a Client Switch Configuration Example" section on page 9-54.

Configuring an Enrollment Profile for a Client Switch Enrolled with a Third-Party Vendor CA

When a client switch is already enrolled with a third-party vendor CA, but you want to reenroll that switch with a Cisco IOS certificate server, perform the following procedures.

Note Before beginning the configuration, you should have already performed the following tasks at the client switch:

- Defined a trustpoint that points to a third-party vendor CA.
- Authenticated and enrolled the client switch with the third-party vendor CA.

To configure a certificate enrollment profile for a client switch that is already enrolled with a third-party vendor CA so that the switch can reenroll with a Cisco IOS certificate server, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoint`** *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the Cisco IOS CA that is to be used.
Step 2	`Router(ca-trustpoint)#` **`enrollment profile`** *`label`*	Specifies that an enrollment profile is to be used for certificate reenrollment. • *label*—Name for the enrollment profile.
Step 3	`Router(ca-trustpoint)#` **`exit`**	Exists ca-trustpoint configuration mode and enters global configuration mode.

	Command	Purpose
Step 4	`Router(config)# crypto pki profile enrollment label`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command in Step 2.
Step 5	`Router(ca-profile-enroll)# enrollment url url`	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP. • *url*—The enrollment URL should point to the Cisco IOS CA.
Step 6	`Router(ca-profile-enroll)# enrollment credential label`	Specifies the non-Cisco IOS CA trustpoint that is to be enrolled with the Cisco IOS CA. • *label*—Name of the CA trustpoint of another vendor.
Step 7	`Router(ca-profile-enroll)# exit`	Exits ca-profile-enroll configuration mode and enters global configuration mode.
Step 8	`Router(config)# exit`	Exits global configuration mode and enters Privileged EXEC mode.
Step 9	`Router# show crypto pki certificates`	(Optional) Verifies information about your certificate, the certificate of the CA, and RA certificates
Step 10	`Router# show crypto pki trustpoints`	(Optional) Displays the trustpoints that are configured in the switch.

For an example of how to configure a certificate enrollment profile for a client switch that is already enrolled with a third-party vendor CA, see the "Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example" section on page 9-55.

Configuring the CA to Accept Enrollment Requests from Clients of a Third-Party Vendor CA

To configure the CA certificate server to accept enrollment requests only from clients who are already enrolled with the third-party vendor CA trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip http server`	Enables the HTTP server on your system.
Step 2	`Router(config)# crypto pki server cs-label`	Enables the certificate server and enters certificate server configuration mode. • *cs-label*—The *cs-label* argument must match the name that was specified by the **crypto pki trustpoint** command for the client switch.

	Command	Purpose
Step 3	Router(cs-server)# **database url** *root-url*	Specifies the location where all database entries for the certificate server will be stored. • *root-url*—Root URL. **Note** If this command is not specified, all database entries will be written to NVRAM.
Step 4	Router(cs-server)# **database level** {**minimal** | **names** | **complete**}	Controls what type of data is stored in the certificate enrollment database. • **minimal**—Enough information is stored only to continue issuing new certificates without conflict; the default value. • **names**—In addition to the information given in the minimal level, the serial number and subject name of each certificate. • **complete**—In addition to the information given in the minimal and names levels, each issued certificate is written to the database. **Note** The **complete** keyword produces a large amount of information; if it is specified, you should also specify an external TFTP server in which to store the data using the **database url** command.
Step 5	Router(cs-server)# **issuer-name** *DN-string*	Sets the CA issuer name to the specified DN-string. • *DN-string*—The default value is as follows: **issuer-name CN=*cs***-*label*.
Step 6	Router(cs-server)# **grant auto trustpoint** *label*	Enables the certificate server to automatically grant only the requests from clients that are already enrolled with the specified non-Cisco IOS CA trustpoint. • *label*—Name of the CA trustpoint of another vendor. **Note** The *label* argument should match the trustpoint that was specified for the client switch's enrollment profile (using the **enrollment credential** command).
Step 7	Router(cs-server)# **lifetime** {**ca-certificate** | **certificate**} *time*	(Optional) Specifies the lifetime, in days, of a CA certificate or a certificate. • *time*—Valid values range from 1 day to 1825 days. The default CA certificate lifetime is 3 years; the default certificate lifetime is 1 year. The maximum certificate lifetime is 1 month less than the lifetime of the CA certificate.

	Command	Purpose
Step 8	`Router(cs-server)# lifetime crl time`	(Optional) Defines the lifetime, in hours, of the Certificate Revocation List (CRL) that is used by the certificate server. • *time*—Maximum lifetime value is 336 hours (2 weeks). The default value is 168 hours (1 week).
Step 9	`Router(cs-server)# cdp-url url`	(Optional) Defines a Certificate Distribution Point (CDP) to be used in the certificates that are issued by the certificate server. • *url*—URL must be an HTTP URL.
Step 10	`Router(cs-server)# shutdown`	Disables a certificate server without removing the configuration. You should enter this command only after you have completely configured your certificate server.
Step 11	`Router(cs-server)# exit`	Exits certificate server configuration mode.
Step 12	`Router(config)# exit`	Exits global configuration mode.
Step 13	`Router# show crypto pki server`	(Optional) Displays the current state and configuration of the certificate server.

For complete configuration information for direct HTTP enroll with CA servers, including the "reenroll using existing certificates" functionality, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t4/feature/guide/gthttpca.html

For direct HTTP enroll with CA servers configuration examples, see the "Direct HTTP Enrollment with CA Servers Configuration Examples" section on page 9-54.

Understanding Manual Certificate Enrollment (TFTP and Cut-and-Paste)

The manual certificate enrollment (TFTP and cut-and-paste) feature allows users to generate a certificate request and accept certification authority (CA) certificates as well as the switch's certificates; these tasks are accomplished by a TFTP server or manual cut-and-paste operations. You might want to utilize TFTP or manual cut-and-paste enrollment in the following situations:

- The CA does not support Simple Certificate Enrollment Protocol (SCEP) (which is the most commonly used method for sending and receiving requests and certificates).
- A network connection between the switch and CA is not possible (which is how a switch running Cisco IOS software obtains its certificate).

Manual Certificate Enrollment (TFTP and Cut-and-Paste) Configuration Guidelines and Restrictions

When configuring manual certificate enrollment (TFTP and cut-and-paste), follow these guidelines and restrictions:

- You can switch between TFTP and cut-and-paste; for example, you can paste the CA certificate using the **enrollment terminal** command, then enter **no enrollment terminal** and **enrollment url tftp://certserver/file_specification** to switch to TFTP to send or receive requests and switch certificates. However, Cisco does not recommend switching URLs if SCEP is used; that is, if the enrollment URL is http://, do not change the enrollment URL between fetching the CA certificate and enrolling the certificate.

Configuring Manual Enrollment Using TFTP

Before configuring manual enrollment using TFTP, you must meet the following prerequisites:

- You must know the correct URL to use if you are configuring certificate enrollment using TFTP.
- The switch must be able to write a file to the TFTP server for the **crypto pki enroll** command.
- Some TFTP servers require that the file exist on the server before it may be written.
- Most TFTP servers require that the file be writable by anyone. This requirement may pose a risk because any switch or other device may write or overwrite the certificate request; thus, the switch will not be able to use the certificate once it is granted by the CA because the request was modified.

To declare the trustpoint CA that your switch should use and configure that trustpoint CA for manual enrollment using TFTP, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoint`** *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)#` **`enrollment`** `[`**`mode`**`] [`**`retry period`** *`minutes`*`] [`**`retry count`** *`number`*`]` **`url`** *`url`*	Specifies the enrollment parameters of your CA. • **mode**—Specifies registration authority (RA) mode if your CA system provides a RA. • *minutes*—Specifies the wait period between certificate request retries. The default is 1 minute between retries. • *number*—Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests. If you are using SCEP for enrollment, the URL must be in the form http://CA_name, where CA_name is the CA's host Domain Name System (DNS) name or IP address. If you are using TFTP for enrollment, the URL must be in the form tftp://certserver/file_specification.

	Command	Purpose
Step 3	`Router(ca-trustpoint)# crypto pki authenticate name`	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 4	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration.
Step 5	`Router(config)# crypto pki enroll name`	Obtains your switch's certificates from the CA. • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 6	`Router(config)# crypto pki import name certificate`	Imports a certificate using TFTP. • *name*—Name of the CA. Enter the *name* value entered in Step 1.

Configuring Certificate Enrollment Using Cut-and-Paste

To declare the trustpoint CA that your switch should use and configure that trustpoint CA for manual enrollment using cut-and-paste, perform this task:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment.
Step 3	`Router(ca-trustpoint)# crypto pki authenticate name`	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1.
Step 4	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration.

	Command	Purpose
Step 5	`Router(config)# `**`crypto pki enroll`** *`name`*	Obtains your switch's certificates from the CA. • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1.
Step 6	`Router(config)# `**`crypto pki import`** *`name`* **`certificate`**	Imports a certificate manually at the terminal. • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1. **Note** You must enter the **crypto pki import** command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the switch; the second time the command is entered, the other certificate is pasted into the switch. (It does not matter which certificate is pasted first.)

Verifying the Manual Certificate Enrollment Configuration

To verify information about your certificate, the certificate of the CA, and RA certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate
  Status:Available
  Certificate Serial Number:14DECE05000000000C48
  Certificate Usage:Encryption

  Issuer:
    CN = msca-root
    O = Cisco Systems
    C = U

  Subject:
    Name:Switch.cisco.com
    OID.1.2.840.113549.1.9.2 = Switch.cisco.com

     CRL Distribution Point:
    http://msca-root/CertEnroll/msca-root.crl

     Validity Date:
    start date:18:16:45 PDT Jun 7 2008
    end   date:18:26:45 PDT Jun 7 2009
    renew date:16:00:00 PST Dec 31 1969

     Associated Trustpoints:MS

   Certificate
    Status:Available
    Certificate Serial Number:14DEC2E9000000000C47
    Certificate Usage:Signature

     Issuer:
    CN = msca-root
    O = Cisco Systems
    C = US
     Subject:
    Name:Switch.cisco.com
    OID.1.2.840.113549.1.9.2 = Switch.cisco.com

     CRL Distribution Point:
        http://msca-root/CertEnroll/msca-root.crl

     Validity Date:
    start date:18:16:42 PDT Jun 7 2008
    end   date:18:26:42 PDT Jun 7 2009
    renew date:16:00:00 PST Dec 31 1969

     Associated Trustpoints:MS

   CA Certificate
    Status:Available
    Certificate Serial Number:3AC0A65E9547C2874AAF2468A942D5EE
    Certificate Usage:Signature

     Issuer:
    CN = msca-root
    O = Cisco Systems
    C = US

     Subject:
    CN = msca-root
    O = Cisco Systems
    C = US

     CRL Distribution Point:
        http://msca-root/CertEnroll/msca-root.crl

  Validity Date:
    start date:16:46:01 PST Feb 13 2008
    end   date:16:54:48 PST Feb 13 2013

  Associated Trustpoints:MS
```

To display the trustpoints that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:

    Subject Name:

    CN = ACSWireless Certificate Manager

     O = cisco.com

     C = US

          Serial Number:01

    Certificate configured.

    CEP URL:http://ACSWireless

    CRL query url:ldap://ACSWireless
```

For complete configuration information for manual certificate enrollment (TFTP and cut-and-paste), refer to this URL:

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

For manual certificate enrollment configuration examples, see the "Manual Certificate Enrollment Configuration Examples" section on page 9-56.

Understanding Certificate Autoenrollment

The certificate autoenrollment feature allows you to configure your switch to automatically request a certificate from the certification authority (CA) that is using the parameters in the configuration. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

Automatic enrollment will be performed on startup for any trustpoint CA that is configured and does not have a valid certificate. When the certificate expires that is issued by a trustpoint CA that has been configured for autoenrollment, a new certificate is requested. Although this feature does not provide seamless certificate renewal, it does provide unattended recovery from expiration.

Before the certificate autoenrollment feature, certificate enrollment required complicated, interactive commands that had to be executed on every switch. This feature allows you to preload all of the necessary information into the configuration and cause each switch to obtain certificates automatically when it is booted. Autoenrollment also checks for expired switch certificates.

Note Before submitting an automatic enrollment request, all necessary enrollment information must be configured.

To configure autoenrollment with a CA on startup, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoint`** *`name`*	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)#` **`enrollment url`** *`url`*	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name, where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	`Router(ca-trustpoint)#` **`subject-name`** `[`*`x.500-name`*`]`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—(Optional) If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	`Router(ca-trustpoint)#` **`ip-address`** `{`*`interface`* `|` **`none`**`}`	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	`Router(ca-trustpoint)#` **`serial-number`** `[`**`none`**`]`	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—(Optional) Specify this keyword if no serial number should be included.
Step 6	`Router(ca-trustpoint)#` **`auto-enroll`** `[`**`regenerate`**`]`	Enables autoenrollment. This command allows you to automatically request a switch certificate from the CA. By default, only the DNS name of the switch is included in the certificate. • **regenerate**—(Optional) Specify this keyword to generate a new key for the certificate even if a named key already exists.

	Command	Purpose
Step 7	`Router(ca-trustpoint)# password string`	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.
Step 8	`Router(ca-trustpoint)# rsakeypair key-label [key-size [encryption-key-size]]`	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) If this command is not enabled, the FQDN key pair is used.

Preloading Root CAs

After enabling automatic enrollment, you must authenticate the CA to establish a chain of trust. This can be done by implementing one of the following methods:

- Obtaining the CA Certificate, page 9-28
- Adding the Certificate of the CA, page 9-29

Obtaining the CA Certificate

To obtain the certificate of the CA, enter the **crypto pki authenticate** command in global configuration mode.

```
Router(config)# crypto pki authenticate name
```

name specifies the name of the CA.

Adding the Certificate of the CA

To add the certificate of the CA, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki certificate chain`** *`name`*	Enters certificate chain configuration mode, which allows you to add or delete specified certificates. • *name*—Name of the CA.
Step 2	`Router(config-cert-chain)#` **`certificate`** *`certificate-serial-number`*	Manually adds or deletes certificates. • *certificate-serial-number*—Serial number of the CA to add.

Verifying CA Information

To display information about your certificates, the certificates of the CA, and registration authority (RA) certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate

  Subject Name

    Name: myrouter.example.com
    IP Address: 10.0.0.1

  Status: Available

  Certificate Serial Number: 428125BDA34196003F6C78316CD8FA95

  Key Usage: Signature


Certificate

  Subject Name

    Name: myswitch.example.com
    IP Address: 10.0.0.1

  Status: Available

  Certificate Serial Number: AB352356AFCD0395E333CCFD7CD33897

  Key Usage: Encryption


CA Certificate

  Status: Available

  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F

  Key Usage: Not Set
```

To display the trustpoints configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:

    Subject Name:

    CN = ACSWireless Certificate Manager

     O = cisco.com

     C = US

    Serial Number:01

    Certificate configured.

    CEP URL:http://ACSWireless

    CRL query url:ldap://ACSWireless
```

For complete configuration information for Certificate Autoenrollment, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftautoen.html

For a certificate autoenrollment configuration example, see the "Certificate Autoenrollment Configuration Example" section on page 9-59.

Understanding Key Rollover for Certificate Renewal

Automatic certificate enrollment was introduced to allow the switch to automatically request a certificate from the certification authority (CA) server. By default, the automatic enrollment feature requests a new certificate when the old certificate expires. Connectivity can be lost while the request is being serviced because the existing certificate and key pairs are deleted immediately after the new key is generated. The new key does not have a certificate to match it until the process is complete, and incoming Internet Key Exchange (IKE) connections cannot be established until the new certificate is issued. The key rollover for certificate renewal feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available.

Key rollover can also be used with a manual certificate enrollment request. Using the same method as key rollover with certificate autoenrollment, a new key pair is created with a temporary name, and the old certificate and key pair are retained until a new certificate is received from the CA. When the new certificate is received, the old certificate and key pair are discarded and the new key pair is renamed with the name of the original key pair. Do not regenerate the keys manually; key rollover will occur when you enter the **crypto pki enroll** command.

Key Rollover for Certificate Renewal Configuration Guidelines and Restrictions

When configuring key rollover for certificate renewal, follow these guidelines and restrictions:

- Trustpoints configured to generate a new key pair using the **regenerate** command or the **regenerate** keyword of the **auto-enroll** command must not share key pairs with other trustpoints. To give each trustpoint its own key pair, use the **rsakeypair** command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatch.

Configuring Automatic Certificate Enrollment with Key Rollover

To configure key rollover with automatic certificate enrollment, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	Router(ca-trustpoint)# **enrollment url** *url*	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name, where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	Router(ca-trustpoint)# **subject-name** [*x.500-name*]	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—(Optional) If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	Router(ca-trustpoint)# **ip-address** {*interface* | **none**}	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	Router(ca-trustpoint)# **serial-number** [**none**]	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—(Optional) Specify this keyword if no serial number should be included.

	Command	Purpose
Step 6	`Router(ca-trustpoint)#` **auto-enroll** [*percent*] [**regenerate**]	Enables autoenrollment. This command allows you to automatically request a switch certificate from the CA. By default, only the DNS name of the switch is included in the certificate. • *percent*—Use the *percent* argument to specify that a new certificate will be requested after the percent lifetime of the current certificate is reached. • **regenerate**—Specify this keyword to generate a new key for the certificate even if a named key already exists. **Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: ! RSA key pair associated with trustpoint is exportable.
Step 7	`Router(ca-trustpoint)#` **password** *string*	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.
Step 8	`Router(ca-trustpoint)#` **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the e*ncryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 9	`Router(ca-trustpoint)#` **exit**	Exits ca-trustpoint configuration mode and returns to global configuration mode.

	Command	Purpose
Step 10	`Router(config)# crypto pki authenticate name`	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 11	`Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Manual Certificate Enrollment with Key Rollover

Note Do not regenerate the keys manually using the **crypto key generate** command; key rollover will occur when the **crypto pki enroll** command is entered.

To configure key rollover with manual certificate enrollment, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • name—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment url url`	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name,_ where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	`Router(ca-trustpoint)# subject-name [x.500-name]`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.

	Command	Purpose
Step 4	Router(ca-trustpoint)# **ip-address** {*interface* | **none**}	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	Router(ca-trustpoint)# **serial-number** [**none**]	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—Specify this keyword if no serial number should be included.
Step 6	Router(ca-trustpoint)# **regenerate**	Enables key rollover with certificate enrollment when the **crypto pki enroll** command is entered. **Note** This command generates a new key for the certificate even if a named key already exists. Do not use the **crypto key generate** command with the key rollover feature. **Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: ! RSA key pair associated with trustpoint is exportable.
Step 7	Router(ca-trustpoint)# **password** *string*	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.

	Command	Purpose
Step 8	Router(ca-trustpoint)# **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 9	Router(ca-trustpoint)# **exit**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 10	Router(config)# **crypto pki authenticate** *name*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 11	Router(config)# **crypto pki enroll** *name*	Requests certificates for all of your RSA key pairs. • *name*—Name of the CA. This command causes your switch to request as many certificates as there are RSA key pairs, so you need perform this command only once, even if you have special-usage RSA key pairs. When the **regenerate** configuration command is configured, this command will perform key rollover. **Note** This command requires you to create a challenge password that is not saved with the configuration. This password is required if your certificate needs to be revoked, so you must remember this password.
Step 12	Router(config)# **exit**	Exits global configuration mode.

For complete configuration information for key rollover for certificate renewal, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtkyroll.html

For key rollover configuration examples, see the "Key Rollover for Certificate Renewal Configuration Examples" section on page 9-59.

Understanding PKI: Query Multiple Servers During Certificate Revocation Check

Before an X.509 certificate presented by a peer is validated, the certificate revocation list (CRL) is checked to make sure that the certificate has not been revoked by the issuing certification authority (CA). The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL.

Previous versions of Cisco IOS software make only one attempt to retrieve the CRL, even when the certificate contains more than one CDP. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected.

The PKI:query multiple servers during certificate revocation check feature provides the ability for Cisco IOS software to make multiple attempts to retrieve the CRL by trying all of the available CDPs in a certificate. This allows operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP is also provided. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

To manually override the existing CDPs for a certificate with a URL or directory specification, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoin**t *name*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	Router(ca-trustpoint)# **match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string*	Manually overrides the existing CDP entries for a certificate with a URL or directory specification. • *certificate-map-label*—A user-specified label that must match the label argument specified in a previously defined **crypto pki certificate map** command. • **url**—Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • **directory**—Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • *string*—The URL or directory specification. Some applications may time out before all CDPs have been tried and will report an error message. This will not affect the switch, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.

For complete configuration information for the PKI: Query Multiple Servers During Certificate Revocation Check feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtcertrc.html

For a query multiple servers configuration example, see the "PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example" section on page 9-60.

Understanding the Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) feature allows users to enable OCSP instead of certificate revocation lists (CRLs) to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate.

OCSP Configuration Guidelines and Restrictions

When configuring OCSP, follow these guidelines and restrictions:

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server. If the OCSP server is unavailable, certificate verification will fail.

- The increased certificate size may cause a problem for low-end switches when certificates are stored on NVRAM. Before you add the Authority Info Access (AIA) extension to a certificate, make sure that the increased size will not cause deployment problems.
- An OCSP server usually operates in either push or poll mode. You can configure a CA server to push revocation information to an OCSP server or configure an OCSP server to periodically download (poll) a CRL from the CA server. To ensure that timely certificate revocation status is obtained, you should carefully consider the push and poll interval.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the switch will not accept the OCSP response. Refer to your OCSP manual for additional information.

Configuring OCSP

To configure your switch for OCSP to check certificate status, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the CA that your switch should use and puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	Router(ca-trustpoint)# **ocsp url** *url*	(Optional) Specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL will override the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate. • *url* —Specifies the HTTP URL to be used.
Step 3	Router(ca-trustpoint)# **revocation-check** *method1* [*method2*[*method3*]]	Checks the revocation status of a certificate. • *method1* [*method2*[*method3*]]—Specifies the method used by the switch to check the revocation status of the certificate. Available methods are as follows: – **crl**—Certificate checking is performed by a CRL. This is the default option. – **none**—Certificate checking is ignored. – **ocsp**—Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.

Verifying the OCSP Configuration

To display information about your certificate and the CA certificate, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate

  Status: Available
  Version: 3
  Certificate Serial Number: 18C1EE03000000004CBD
  Certificate Usage: General Purpose

  Issuer:
    cn=msca-root
    ou=pki msca-root
    o=cisco
    l=santa cruz2
    st=CA
    c=US
    ea=user@example.com

  Subject:
    Name: myrouter.example.com
    hostname=myrouter.example.com

  CRL Distribution Points:
    http://msca-root/CertEnroll/msca-root.crl

  Validity Date:
    start date: 19:50:40 GMT Oct 5 2004
    end   date: 20:00:40 GMT Oct 12 2004

    Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (360 bit)

    Signature Algorithm: SHA1 with RSA Encryption
    Fingerprint MD5: 2B5F53E6 E3E892E6 3A9D3706 01261F10
    Fingerprint SHA1: 315D127C 3AD34010 40CE7F3A 988BBDA5 CD528824

    X509v3 extensions:
    X509v3 Key Usage: A0000000
    Digital Signature
    Key Encipherment
    X509v3 Subject Key ID: D156E92F 46739CBA DFE66D2D 3559483E B41ECCF4
    X509v3 Authority Key ID: 37F3CC61 AF5E7C0B 434AB364 CF9FA0C1 B17C50D9
    Authority Info Access:

    Associated Trustpoints: msca-root
    Key Label: myrouter.example.com

   CA Certificate

    Status: Available
    Version: 3
    Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
    Certificate Usage: Signature

    Issuer:
    cn=msca-root
    ou=pki msca-root
    o=cisco
    l=santa cruz2
    st=CA
    c=US
    ea=user@example.com

    Subject:
    cn=msca-root
    ou=pki msca-root
    o=cisco
    l=santa cruz2
    st=CA
    c=US
    ea=user@example.com

    CRL Distribution Points:
    http://msca-root.example.com/CertEnroll/msca-root.crl

    Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017

    Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (512 bit)

    Signature Algorithm: SHA1 with RSA Encryption
    Fingerprint MD5: 84E470A2 38176CB1 AA0476B9 C0B4F478
    Fingerprint SHA1: 0F57170C 654A5D7D 10973553 EFB0F94F 2FAF9837

    X509v3 extensions:
    X509v3 Key Usage: C6000000
    Digital Signature
    Non Repudiation
    Key Cert Sign
    CRL Signature

  X509v3 Subject Key ID: 37F3CC61 AF5E7C0B 434AB364 CF9FA0C1 B17C50D9
  X509v3 Basic Constraints:
    CA: TRUE

  Authority Info Access:
  Associated Trustpoints: msca-root
```

To display the trustpoints and configured trustpoint subcommands that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:
    Subject Name:
    CN = ACSWireless Certificate Manager
    O = cisco.com
    C = US
    Serial Number:01
    Certificate configured.
    CEP URL:http://ACSWireless
    CRL query url:ldap://ACSWireless
```

For complete configuration information for OCSP, refer to this URL:

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

For OCSP configuration examples, see the "Online Certificate Status Protocol Configuration Examples" section on page 9-60.

Understanding Certificate Security Attribute-Based Access Control

Under the IPsec protocol, certificate authority (CA) interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. The certificate security attribute-based access control feature adds fields to the certificate to create a certificate-based ACL.

Certificate Security Attribute-Based Access Control Configuration Guidelines and Restrictions

When configuring certificate security attribute-based access control, follow these guidelines and restrictions:

- The certificate-based ACL specifies one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have. There are six logical tests for comparing the field with the value: equal, not equal, contains, does not contain, less than, and greater than or equal.
- If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL.
- The same field can be specified multiple times within the same ACL.
- More than one ACL can be specified. Each ACL will be processed in turn until a match is found or all of the ACLs have been processed.
- Memory is required to hold the ACLs as they are created and as they are loaded from the configuration file. The amount of memory depends on which fields within the certificate are being checked and how many ACLs have been defined. Certificate-based ACL support requires one or more compare operations when the fields in a certificate are being checked. Only the fields specified by the ACL are checked. The compare operations are a small part of certificate validation and will not have a noticeable effect on switch performance when validating certificates.

Configuring Certificate Security Attribute-Based Access Control

To configure Certificate Security Attribute-Based Access Control, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki certificate map label sequence-number`	Starts ca-certificate-map mode and defines certificate-based ACLs by assigning a label for the ACL that will also be referenced within the **crypto pki trustpoint** command. • *label*—An arbitrary string that identifies the ACL. • *sequence-number*—A sequence number that orders ACLs with the same label.
Step 2	`Router(ca-certificate-map)# field-name match-criteria match-value`	In ca-certificate-map mode, you specify one or more certificate fields together with their matching criteria and the value to match. • *field-name*—Specifies one of the following case-insensitive name strings or a date: – **subject-name** – **issuer-name** – **unstructured-subject-name** – **alt-subject-name** – **name** – **valid-start** – **expires-on** **Note** Date field format is *dd mm yyyy hh*:*mm*:*ss* or *mm dd yyyy hh*:*mm*:*ss*. • *match-criteria*—Specifies one of the following logical operators: – **eq**—Equal (valid for name and date fields) – **ne**—Not equal (valid for name and date fields) – **co**—Contains (valid only for name fields) – **nc**—Does not contain (valid only for name fields) – **lt** —Less than (valid only for date fields) – **ge** —Greater than or equal (valid only for date fields) • *match-value*—Specifies the name or date to test with the logical operator assigned by *match-criteria*. For example: `Router(ca-certificate-map)# subject-name co Cisco`
Step 3	`Router(ca-certificate-map)# exit`	Exits ca-certificate-map mode.

	Command	Purpose
Step 4	`Router(config)#` **`crypto pki trustpoint`** *`name`*	Starts ca-trustpoint configuration mode and creates a name for the CA. • *name*—Specifies a name for the CA.
Step 5	`Router(ca-trustpoint)#` **`match certificate`** *`certificate-map-label`*	Associates the certificate-based ACL defined with the **crypto pki certificate map** command to the trustpoint. • *certificate-map-label*—Specifies the label argument specified in the previously defined **crypto pki certificate map** command in Step 1.
Step 6	`Router(ca-trustpoint)#` **`exit`**	Exits ca-trustpoint configuration mode.

Verifying Certificate-Based ACLs

To verify the certificate-based ACL configuration, enter the **show crypto pki certificates** command. The following example shows the components of the certificates (CA and switch certificate) installed on the switch when the switch has both authenticated and enrolled with a trustpoint:

```
Router# show crypto pki certificates

CA Certificate
   Status: Available
   Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
   Certificate Usage: Signature

   Issuer:
    CN = new-user
    OU = pki new-user
    O = cisco
    L = santa cruz2
    ST = CA
    C = US
    EA = user@cysco.net

    Subject:
    CN = new-user
    OU = pki new-user
    O = cisco
    L = santa cruz2
    ST = CA
    C = US
    EA = user@cysco.net

    CRL Distribution Point:
    http://new-user.cysco.net/CertEnroll/new-user.crl

    Validity Date:
    start date: 14:19:29 PST Oct 31 2002
    end date: 14:27:27 PST Oct 31 2017

    Associated Trustpoints: MS


Certificate
  Status: Available
  Certificate Serial Number: 193E28D20000000009F7
  Certificate Usage: Signature

  Issuer:
     CN = new-user
     OU = pki new-user
     O = cisco
     L = santa cruz2
     ST = CA
    C = US
     EA = user@cysco.net

  Subject:
    Name: User1.Cysco.Net
    OID.1.2.840.113549.1.9.2 = User1.Cysco.Net

  CRL Distribution Point:
    http://new-user.cysco.net/CertEnroll/new-user.crl

  Validity Date:
    start date: 12:40:14 PST Feb 26 2003
    end   date: 12:50:14 PST Mar 5 2003
    renew date: 16:00:00 PST Dec 31 1969

  Associated Trustpoints: MS
```

For complete configuration information for Certificate Security Attribute-Based Access Control, refer to this URL:

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

For a certificate-based ACL example, see the "Certificate Security Attribute-Based Access Control Configuration Example" section on page 9-61.

Understanding PKI AAA Authorization Using the Entire Subject Name

When using public key infrastructure (PKI) and authentication, authorization, and accounting (AAA) functionality, users sometimes have attribute-value (AV) pairs that are different from those of every other user. As a result, a unique username is required for each user. The PKI AAA authorization using the entire subject name feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username.

PKI AAA Authorization Using the Entire Subject Name Configuration Guidelines and Restrictions

When configuring PKI AAA authorization using the entire subject name, follow these guidelines and restrictions:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). This feature will not work for the AAA server having such a character-set limitation.

- The **subject-name** command in the trustpoint configuration might not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the switch are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- Certificate authority (CA) servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring the AAA server with a full DN (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least-significant RDN first) is used.

Configuring PKI AAA Authorization Using the Entire Subject Name

To configure the entire certificate subject name for PKI authentication, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	`Router config)# aaa authorization network listname [method]`	Sets the parameters that restrict user access to a network. • *listname*—Character string used to name the list of authorization methods. • *method*—(Optional) Specifies an authorization method to be used for authorization. The *method* argument can be **group radius**, **group tacacs+**, or **group** *group-name*.
Step 3	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 4	`Router(ca-trustpoint)# enrollment url url`	Specifies the enrollment parameters of your CA. • *url*—The *url* argument is the URL of the CA to which your switch should send certificate requests.
Step 5	`Router(ca-trustpoint)# revocation-check method`	(Optional) Checks the revocation status of a certificate. • *method*—Method used by the switch to check the revocation status. Available methods are **ocsp**, **none**, and **crl**.
Step 6	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and enters global configuration mode.

	Command	Purpose
Step 7	`Router(config)#` **`authorization list`** `{`*`listname`*`}`	Specifies the AAA authorization list. - *listname*—Name of the list.
Step 8	`Router(config)#` **`authorization username subjectname all`**	Sets parameters for the different certificate fields that are used to build the AAA username. **all**—Specifies that the entire subject name of the certificate will be used as the authorization username.
Step 9	`Router(config)#` **`tacacs-server host`** *`hostname`* `[`**`key`** *`string`*`]` or	Specifies a TACACS+ host. - *hostname*—Name of the host. - **key** *string*—(Optional) Character string specifying authentication and encryption key.
	`Router(config)#` **`radius-server host`** *`hostname`* `[`**`key`** *`string`*`]`	Specifies a RADIUS host.

For complete configuration information for the PKI AAA authorization using the entire subject name feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t11/feature/guide/gt_dnall.html

For a PKI AAA Authorization Using the Entire Subject Name configuration example, see the "PKI AAA Authorization Using the Entire Subject Name Configuration Example" section on page 9-62.

Understanding Source Interface Selection for Outgoing Traffic with Certificate Authority

The source interface selection for outgoing traffic with certificate authority feature allows you to specify that the address of an interface be used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured.

Configuring Source Interface Selection for Outgoing Traffic with Certificate Authority

To configure the interface that you want to use as the source address for all outgoing TCP connections associated with a trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoin**t *name*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	Router(ca-trustpoint)# **enrollment url** *url*	Specifies the enrollment parameters of your CA. • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.
Step 3	Router(ca-trustpoint)# **source interface** *interface-address*	Specifies the interface to be used as the source address for all outgoing TCP connections associated with that trustpoint. • *interface-address*—Interface address.
Step 4	Router(config)# **interface** *type slot*/[subslot]/*port*	Configures an interface type and enters interface configuration mode. • *type*—Type of interface being configured. • *slot*/[subslot]/*port*—Number of the slot, subslot (optional), and port to be configured.
Step 5	Router(config-if)# **description** *string*	Adds a description to an interface configuration. • *string*—Descriptive string.
Step 6	Router(config-if)# **ip address** *ip-address mask*	Sets a primary or secondary IP address for an interface. • *ip-address*—IP address. • *mask*—Subnet mask.
Step 7	Router(config-if)# **interface** *type slot*/[*subslot*]/*port*	Configures an interface type. • *type*—Type of interface being configured. • *slot*/[*subslot*]/*port*—Number of the slot, subslot (optional), and port to be configured.
Step 8	Router(config-if)# **description** *string*	Adds a description to an interface configuration. • *string*—Descriptive string.

	Command	Purpose
Step 9	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for an interface. • *ip-address*—IP address. • *mask*—Subnet mask. • [*secondary*]—(Optional) Secondary address.
Step 10	`Router(config-if)# crypto map map-name`	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set.

For complete configuration information for source interface selection for outgoing traffic with certificate authority, refer to this URL:

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

For a source interface selection configuration example, see the "Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example" section on page 9-62.

Understanding Persistent Self-Signed Certificates

The persistent self-signed certificates feature saves a certificate generated by a Secure HTTP (HTTPS) server for the Secure Sockets Layer (SSL) handshake in a router's startup configuration.

Cisco IOS software has an HTTPS server that allows access to web-based management pages using a secure SSL connection. SSL requires the server to have an X.509 certificate that is sent to the client (web browser) during the SSL handshake to establish a secure connection between the server and the client.

The client expects the SSL server's certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a public key infrastructure (PKI) application programming interface (API). When the client receives this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate, so you are once again asked to accept it.

Requesting acceptance of the router's certificate each time that the router reloads can be annoying and may present an opportunity for an attacker to substitute an unauthorized certificate during the time that you are being asked to accept the certificate.

The persistent self-signed certificates feature overcomes all these limitations by saving a certificate in the router's startup configuration, resulting in the following benefits:

- Having a persistent self-signed certificate stored in the router's startup configuration (NVRAM) lessens the opportunity for an attacker to substitute an unauthorized certificate because the browser is able to compare the certificate offered by the router with the previously saved certificate and warn you if the certificate has changed.

- Having a persistent self-signed certificate stored in the router's startup configuration eliminates the user intervention that is necessary to accept the certificate every time that the router reloads.
- Because user intervention is no longer necessary to accept the certificate, the secure connection process is faster.

Persistent Self-Signed Certificates Configuration Guidelines and Restrictions

When configuring persistent self-signed certificates, follow these guidelines and restrictions:

- You must load an image that supports SSL.
- You can configure only one trustpoint for a persistent self-signed certificate.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Note This section is optional because if you enable the Secure HTTP (HTTPS) server, it generates a self-signed certificate automatically using default values. To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as it is enabled.

To configure a trustpoint and specify self-signed certificate parameters, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the certificate authority (CA) that your router should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	Router(ca-trustpoint)# **enrollment selfsigned**	Specifies self-signed enrollment.
Step 3	Router(ca-trustpoint)# **subject-name** [*x.500-name*]	(Optional) Specifies the requested subject name to be used in the certificate request. • *x.500-name*—(Optional) If the x.500-name argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.

	Command	Purpose
Step 4	`Router(ca-trustpoint)# `**`rsakeypair`** *`key-label`* `[`*`key-size`* `[`*`encryption-key-size`*`]]`	(Optional) Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 5	`Router(ca-trustpoint)# `**`crypto pki enroll`** *`trustpoint-name`*	Tells the router to generate the persistent self-signed certificate. • *trustpoint-name*—Name of the CA.
Step 6	`Router(ca-trustpoint)# `**`end`**	(Optional) Exits ca-trustpoint configuration mode.

Enabling the HTTPS Server

To enable the HTTPS server, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`ip http secure-server`**	Enables the secure HTTP web server. **Note** A key pair (modulus 1024) and a certificate are generated.
Step 2	`Router(config)# `**`end`**	Exits global configuration mode.

Note You must enter a **write memory** command to save the configuration. This command also saves the self-signed certificate and the HTTPS server in enabled mode.

Verifying the Persistent Self-Signed Certificate Configuration

To verify that a self-signed certificate and a trustpoint have been created, use the **show crypto pki certificates**, **show crypto mypubkey rsa***,* and the **show crypto pki trustpoints** commands.

The **show crypto pki certificates** command displays information about your certificate, the CA certificate, and any registration authority certificates:

```
Router# show crypto pki certificates

Router Self-Signed Certificate
   Status: Available
   Certificate Serial Number: 01
   Certificate Usage: General Purpose
   Issuer:
    cn=IOS-Self-Signed-Certificate-3326000105
   Subject:
    Name: IOS-Self-Signed-Certificate-3326000105
    cn=IOS-Self-Signed-Certificate-3326000105
   Validity Date:
    start date: 19:14:14 GMT Dec 21 2004
    end   date: 00:00:00 GMT Jan 1 2020
   Associated Trustpoints: TP-self-signed-3326000105
```

Note The number 3326000105 above is the router's serial number and varies depending on the router's actual serial number.

The **show crypto mypubkey rsa** command displays information about the key pair corresponding to the self-signed certificate:

```
Router# show crypto mypubkey rsa

% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
  Usage: General Purpose Key
  Key is not exportable.
  Key Data:
    30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88F70
    6BC78B6D 67D6CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
    BFD1C2B7 E64A3804 9BBD7326 207BD456 19BAB78B D075E78E 00D2560C B09289AE
    6DECB8B0 6672FB3A 5CDAEE92 9D4C4F71 F3BCB269 214F6293 4BA8FABF 9486BCFC
    2B941BCA 550999A7 2EFE12A5 6B7B669A 2D88AB77 39B38E0E AA23CB8C B7020301 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
  Usage: Encryption Key
  Key is not exportable.
  Key Data:
    307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5680E 89777B42
    463E5783 FE96EA9E F446DC7B 70499AF3 EA266651 56EE29F4 5B003D93 2FC9F81D
    8A46E12F 3FBAC2F3 046ED9DD C5F27C20 1BBA6B9B 08F16E45 C34D6337 F863D605
    34E30F0E B4921BC5 DAC9EBBA 50C54AA0 BF551BDD 88453F50 61020301 0001
```

Note The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated once any key pair is created on the router and SSH starts up.

The **show crypto pki trustpoints** command displays the trustpoints that are configured in the router:

```
Router# show crypto pki trustpoints

Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.cisco.com
        Serial Number: 01
    Persistent self-signed certificate trust point
```

For complete configuration information for persistent self-signed certificates, refer to this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a008040adf0.html

For persistent self-signed certificates configuration examples, see the "Persistent Self-Signed Certificates Configuration Examples" section on page 9-63.

Configuration Examples

This section provides examples of the following configurations:

- Multiple RSA Key Pairs Configuration Example, page 9-53
- Protected Private Key Storage Configuration Examples, page 9-53
- Trustpoint CA Configuration Example, page 9-54
- Query Mode Definition Per Trustpoint Configuration Example, page 9-54
- Direct HTTP Enrollment with CA Servers Configuration Examples, page 9-54
- Manual Certificate Enrollment Configuration Examples, page 9-56
- Certificate Autoenrollment Configuration Example, page 9-59
- Key Rollover for Certificate Renewal Configuration Examples, page 9-59
- PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example, page 9-60
- Online Certificate Status Protocol Configuration Examples, page 9-60
- Certificate Security Attribute-Based Access Control Configuration Example, page 9-61
- PKI AAA Authorization Using the Entire Subject Name Configuration Example, page 9-62
- Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example, page 9-62
- Persistent Self-Signed Certificates Configuration Examples, page 9-63

Multiple RSA Key Pairs Configuration Example

The following example is a sample trustpoint configuration that specifies the RSA key pair exampleCAkeys:

```
Router(config)# crypto key generate rsa general-keys label exampleCAkeys
Router(config)# crypto pki trustpoint exampleCAkeys
Router(ca-trustpoint)# enrollment url http://exampleCAkeys/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# rsakeypair exampleCAkeys 1024 1024
```

Protected Private Key Storage Configuration Examples

This section contains the following configuration examples:

- Encrypted Key Configuration Example, page 9-54
- Locked Key Configuration Example, page 9-54

Encrypted Key Configuration Example

The following example shows how to encrypt the pki1-72a.cisco.com RSA key:

```
Router(config)# crypto key encrypt rsa name pki1-72a.cisco.com passphrase cisco1234
Router(config)# exit
```

Locked Key Configuration Example

The following example shows how to lock the pki1-72a.cisco.com key:

```
Router# crypto key lock rsa name pki1-72a.cisco.com passphrase cisco1234
```

Trustpoint CA Configuration Example

The following example shows how to declare the CA named kahului and specify characteristics for the trustpoint CA:

```
Router(config)# crypto pki trustpoint kahului
Router(ca-trustpoint)# enrollment url http://kahului
Router(ca-trustpoint)# crl query ldap://kahului
```

Query Mode Definition Per Trustpoint Configuration Example

The following configuration example shows a trustpoint CA that uses query mode:

```
Router(config)# crypto pki trustpoint trustpoint1
Router(ca-trustpoint)# enrollment url http://ca-server1
Router(ca-trustpoint)# crl query http://ca-server1
Router(ca-trustpoint)# query certificate
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustpoint1
Router(config)# crypto key generate rsa
Router(config)# crypto pki enroll trustpoint1
```

Direct HTTP Enrollment with CA Servers Configuration Examples

This section provides the following configuration examples:

- Enrollment Profile for a Client Switch Configuration Example, page 9-54
- Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example, page 9-55
- Certificate Server Automatically Accepting Enrollment Requests Only from the Client Switch Configuration Example, page 9-55

Enrollment Profile for a Client Switch Configuration Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

```
Router(config)# crypto pki trustpoint Entrust
Router(ca-trustpoint)# enrollment profile E
Router(ca-trustpoint)# exit
Router(config)# crypto pki profile enrollment E
Router(ca-profile-enroll)# authentication url  http://entrust:81
Router(ca-profile-enroll)# authentication command  GET /certs/cacert.der
Router(ca-profile-enroll)# enrollment url  http://entrust:81/cda-cgi/clientcgi.exe
Router(ca-profile-enroll)# enrollment command  POST reference_number=$P2&authcode=$P1
&retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc
Router(ca-profile-enroll)# parameter 2 value 5001

Router(config)# crypto ca profile enrollment E
Router(ca-profile-enroll)# authentication url http://ca-server.example.com
Router(ca-profile-enroll)# authentication command GET $P1
Router(ca-profile-enroll)# enrollment url
Router(ca-profile-enroll)# enrollment command ^C
POST action=getServerCert
&pkcs10Request=$REQ
&reference_number=$P2
&authcode=P3
&retrievedAs=rawDER
^C
Router(ca-profile-enroll)# parameter 1 value cacert.crt
Router(ca-profile-enroll)# parameter 2 prompt Enter the Reference Number:
Router(ca-profile-enroll)# parameter 3 prompt Enter the Auth Code:
```

Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example

The following example shows how to configure the following tasks on the client switch:

- Define the msca-root trustpoint that points to the third-party vendor CA and enroll and authenticate the client with the third-party vendor CA.
- Define cs trustpoint for the Cisco IOS CA.
- Define enrollment profile “cs1,” which points to Cisco IOS CA, and mention (via the enrollment credential command) that msca-root is being initially enrolled with the Cisco IOS CA.

```
! Define trustpoint "msca-root" for non-Cisco IOS CA.
Router(config)# crypto pki trustpoint msca-root
Router(ca-trustpoint)# enrollment mode ra
Router(ca-trustpoint)# enrollment url http://msca-root:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# ip-address FastEthernet2/0
Router(ca-trustpoint)# revocation-check crl

! Configure trustpoint "cs" for Cisco IOS CA.
Router(config)# crypto pki trustpoint cs
Router(ca-trustpoint)# enrollment profile cs1
Router(ca-trustpoint)# revocation-check crl

! Define enrollment profile "cs1."
Router(config)# crypto pki profile enrollment cs1
Router(ca-profile-enroll)# enrollment url  http://cs:80
Router(ca-profile-enroll)# enrollment credential  msca-root
```

Certificate Server Automatically Accepting Enrollment Requests Only from the Client Switch Configuration Example

The following example shows how to configure the certificate server, and enter the **grant auto trustpoint** command to instruct the certificate server to accept enrollment requests only from clients who are already enrolled with msca-root trustpoint:

```
Router(config)# crypto pki server cs
Router(cs-server)# database level minimum
Router(cs-server)# database url nvram:
Router(cs-server)# issuer-name CN=cs
Router(cs-server)# grant auto trustpoint msca-root

Router(config)# crypto pki trustpoint cs
Router(ca-trustpoint)# revocation-check crl
Router(ca-trustpoint)# rsakeypair cs

Router(ca-trustpoint)# crypto pki trustpoint msca-root
Router(ca-trustpoint)# enrollment mode ra
Router(ca-trustpoint)# enrollment url http://msca-root:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# revocation-check crl
```

Manual Certificate Enrollment Configuration Examples

This section provides the following manual certificate enrollment configuration examples:

- Manual Certificate Enrollment Using TFTP Configuration Example, page 9-56
- Manual Certificate Enrollment Using Cut-and-Paste Configuration Example, page 9-56

Manual Certificate Enrollment Using TFTP Configuration Example

The following example shows the configuration of manual certificate enrollment using TFTP:

```
Router(config)# crypto pki trustpoint MS
Router(ca-trustpoint)# enrollment url tftp://CA-Server/TFTPfiles/switch1
Router(ca-trustpoint)# crypto pki authenticate MS
Router(ca-trustpoint)# exit
Router(config)# crypto pki enroll MS
Router(config)# crypto pki import MS certificate
```

Manual Certificate Enrollment Using Cut-and-Paste Configuration Example

The following example shows how to configure manual cut-and-paste certificate enrollment. In this example, the name of the trustpoint CA is MS, and the **crypto pki import** command is entered twice because usage keys (signature and encryption keys) are used.

```
Router(config)# crypto pki trustpoint MS
Router(ca-trustpoint)# enrollment terminal
Router(ca-trustpoint)# crypto pki authenticate MS


Enter the base 64 encoded CA certificate.

End with a blank line or the word "quit" on a line by itself


-----BEGIN CERTIFICATE-----

MIICNDCCAd6gAwIBAgIQOsCmXpVHwodKryRoqULV7jANBgkqhkiG9w0BAQUFADA5
MQswCQYDVQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJ
bXNjYS1yb290MB4XDTAyMDIxNDAwNDYwMVoXDTA3MDIxNDAwNTQ0OFowOTELMAkG
A1UEBhMCVVMxFjAUBgNVBAoTDUNpc2NvIFN5c3RlbXMxEjAQBgNVBAMTCW1zY2Et
cm9vdDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCix8nIGFg+wvy3BjFbVi25wYoG
K2N0HWWHpqxFuFhqyBnIC0OshIn9CtrdN3JvUNHr0NIKocEwNKUGYmPwWGTfAgMB
AAGjgcEwgb4wCwYDVR0PBAQDAgHGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYE
FKIacsl6dKAfuNDVQymlSp7esf8jMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9t
c2NhLXJvb3QvQ2VydEVucm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8v
```

```
XFxtc2NhLXJvb3RcQ2VydEVucm9sbFxtc2NhLXJvb3QuY3JsMBAGCSsGAQQBgjcV
AQQDAgEAMA0GCSqGSIb3DQEBBQUAA0EAeuZkZMX9qkoLHfETYTpVWjZPQbBmwNRA
oJDSdYdtL3BcI/uLL5q7EmODyGfLyMGxuhQYx5r/40aSQgLCqBq+yg==

-----END CERTIFICATE-----


Certificate has the following attributes:

Fingerprint:D6C12961 CD78808A 4E02193C 0790082A

% Do you accept this certificate? [yes/no]:y

Trustpoint CA certificate accepted.

% Certificate successfully imported


Router(config)#

Router(config)# crypto pki enroll MS

% Start certificate enrollment..



% The subject name in the certificate will be:Router.cisco.com

% Include the router serial number in the subject name? [yes/no]:n

% Include an IP address in the subject name? [no]:n

Display Certificate Request to terminal? [yes/no]:y

Signature key certificate request -

Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAxdhXFDiWAn/hIZs9zfOtssKA
daoWYu0ms9Fe/Pew01dh14vXdxgacstOs2Pr5wk6jLOPxpvxOJPWyQM6ipLmyVxv
ojhyLTrVohrh6Dnqcvk+G/5ohss9o9RxvONwx042pQchFnx9EkMuZC7evwRxJEqR
mBHXBZ8GmP3jYQsjS8MCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgeAMA0GCSqGSIb3DQEBBAUAA4GBAMT6WtyFw95POY7UtF+YIYHiVRUf4SCq
hRIAGrljUePLo9iTqyPU1Pnt8JnIZ5P5BHU3MfgP8sqodaWub6mubkzaohJ1qD06
O87fnLCNid5Tov5jKogFHIki2EGGZxBosUw9lJlenQdNdDPbJc5LIWdfDvciA6jO
Nl8rOtKnt8Q+

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]:

Encryption key certificate request -

Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwG60QojpDbzbKnyj8FyTiOcv
THkDP7XD4vLT1XaJ409z0gSIoGnIcdFtXhVlBWtpq3/O9zYFXr1tH+BMCRQi3Lts
0IpxYa3D9iFPqev7SPXpsAIsY8a6FMq7TiwLObqiQjLKL4cbuV0Frjl0Yuv5A/Z+
kqMOm7c+pWNWFdLe9lsCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgUgMA0GCSqGSIb3DQEBBAUAA4GBACF7feURj/fJMojPBlR6fa9BrlMJx+2F
H91YM/CIiz2n4mHTeWTWKhLoT8wUfa9NGOk7yi+nF/F7035twLfq6n2bSCTW4aem
8jLMMaeFxwkrV/ceQKrucmNC1uVx+fBy9rhnKx8j60XE25tnp1U08r6om/pBQABU
```

```
eNPFhozcaQ/2

---End - This line not part of the certificate request---


Redisplay enrollment request? [yes/no]:

n

Router(config)#crypto pki import MS certificate


Enter the base 64 encoded certificate.

End with a blank line or the word "quit" on a line by itself


MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0MloXDTAzMDYwODAxMjY0MlowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMXYVxQ4lgJ/4SGbPc3zrbLCgHWqFmLtJrPRXvz3sNNXYdeL13cYGnLL
TrNj6+cJOoyzj8ab8TiT1skDOoqS5slcb6I4ci061aIa4eg56nL5Phv+aIbLPaPU
cbzjcMdONqUHIRZ8fRJDLmQu3r8EcSRKkZgR1wWfBpj942ELI0vDAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFL8Quz8dyz4EGIeKx9A8UMNHLE4s
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAJo2
r6sHPGBdTQX2EDoJpR/A2UHXxRYqVSHkFKZw0z31r5JzUM0oPNUETV7mnZlYNVRZ
CSEX/G8boi3WOjz9wZo=

% Router Certificate successfully imported


Router(config)#

Router(config)# crypto pki import MS certificate


Enter the base 64 encoded certificate.

End with a blank line or the word "quit" on a line by itself


MIIDajCCAxSgAwIBAgIKFN7OBQAAAAAMSDANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0NVoXDTAzMDYwODAxMjY0NVowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMButEKI6Q282yp8o/Bck4jnL0x5Az+1w+Ly09V2ieNPc9IEiKBpyHHR
bV4VZQVraat/zvc2BV69bR/gTAkUIty7bNCKcWGtw/YhT6nr+0j16bACLGPGuhTK
u04sCzm6okIyyi+HG7ldBa45dGLr+QP2fpKjDpu3PqVjVhXS3vZbAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCBSAwHQYDVR0OBBYEFPDO29oRdlEUSgBMg6jZR+YFRWlj
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
```

```
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAHaU
hyCwLirUghNxCmLzXRG7C3W1j0kSX7a4fX9OxKR/Z2SoMjdMNPPyApuh8SoT2zBP
ZKjZU2WjcZG/nZF4W5k=

% Router Certificate successfully imported
```

Certificate Autoenrollment Configuration Example

The following example shows how to configure the switch to autoenroll with a CA on start-up:

```
Router(config)# crypto pki trustpoint frog
Router(ca-trustpoint)# enrollment url http://frog.phoobin.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet-0
Router(ca-trustpoint)# auto-enroll regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsa-key frog 2048
!
Router(config)# crypto pki certificate chain frog
Router(config-cert-chain)# certificate ca 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F70D0101 04050030
79310B30 09060355 04061302 5553310B 30090603 55040813 02434131 15301306
0355040A 130C4369 73636F20 53797374 656D3120 301E0603 55040B13 17737562
6F726420 746F206B 6168756C 75692049 50495355 31243022 06035504 03131B79
6E692D75 31302043 65727469 66696361 7465204D 616E6167 6572301E 170D3030
30373134 32303536 32355A17 0D303130 37313430 31323834 335A3032 310E300C
06035504 0A130543 6973636F 3120301E 06092A86 4886F70D 01090216 11706B69
2D343562 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100B3 0512A201 3B4243E1 378A9703 8AC5E3CE F77AF987 B5A422C4
15E947F6 70997393 70CF34D6 63A86B9C 4347A81A 0551FC02 ABA62360 01EF7DD2
6C136AEB 3C6C3902 03010001 A381F630 81F3300B 0603551D 0F040403 02052030
1C060355 1D110415 30138211 706B692D 3435622E 63697363 6F2E636F 6D301D06
03551D0E 04160414 247D9558 169B9A21 23D289CC 2DDA2A9A 4F77C616 301F0603
551D2304 18301680 14BD742C E892E819 1D551D91 683F6DB2 D8847A6C 73308185
0603551D 1F047E30 7C307AA0 3CA03AA4 38303631 0E300C06 0355040A 13054369
73636F31 24302206 03550403 131B796E 692D7531 30204365 72746966 69636174
65204D61 6E616765 72A23AA4 38303631 0E300C06 0355040A 13054369 73636F31
24302206 03550403 131B796E 692D7531 30204365 72746966 69636174 65204D61
6E616765 72300D06 092A8648 86F70D01 01040500 03410015 BC7CECF9 696697DF
E887007F 7A8DA24F 1ED5A785 C5C60452 47860061 0C18093D 08958A77 5737246B
0A25550A 25910E27 8B8B428E 32F8D948 3DD1784F 954C70
quit
```

Key Rollover for Certificate Renewal Configuration Examples

This section contains the following examples:

- Certificate Autoenrollment with Key Rollover Configuration Example, page 9-60
- Manual Certificate Enrollment with Key Rollover Configuration Example, page 9-60

Certificate Autoenrollment with Key Rollover Configuration Example

The following example shows how to configure the switch to autoenroll with the CA named trustme1 on startup. In this example, the **regenerate** keyword is specified, so a new key will be generated for the certificate. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```
Router(config)# crypto pki trustpoint trustme1
Router(ca-trustpoint)# enrollment url http://trustme1.company.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet0
Router(ca-trustpoint)# serial-number none
Router(ca-trustpoint)# auto-enroll 90 regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsakeypair trustme1 2048
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustme1
```

Manual Certificate Enrollment with Key Rollover Configuration Example

The following example shows how to configure key rollover to regenerate new keys with a manual certificate enrollment from the CA named trustme2.

```
Router(config)# crypto pki trustpoint trustme2
Router(ca-trustpoint)# enrollment url http://trustme2.company.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet0
Router(ca-trustpoint)# serial-number none
Router(ca-trustpoint)# regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsakeypair trustme2 2048
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustme2
Router(config)# crypto pki enroll trustme2
Router(config)# exit
```

PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example

The following example uses the **match certificate override cdp** command to override the CDPs for the certificate map named Group1 defined in a **crypto pki certificate map** command:

```
Router(config)# crypto pki certificate map Group1 10
Router(ca-certificate-map)# subject-name co ou=WAN
Router(ca-certificate-map)# subject-name co o=Cisco
Router(config)# crypto pki trustpoint pki
Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com
```

Online Certificate Status Protocol Configuration Examples

This section provides the following configuration examples:

- OCSP Server Configuration Example, page 9-61
- CRL Then OCSP Server Configuration Example, page 9-61

- Specific OCSP Server Configuration Example, page 9-61

OCSP Server Configuration Example

The following example shows how to configure the switch to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

CRL Then OCSP Server Configuration Example

The following example shows how to configure the switch to download the CRL from the certificate distribution point (CDP); if the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Specific OCSP Server Configuration Example

The following example shows how to configure your switch to use the OCSP server at the HTTP URL http://myocspserver:81. If the server is down, revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

Certificate Security Attribute-Based Access Control Configuration Example

The following example shows how to configure a certificate-based ACL:

```
Router(config)# crypto pki certificate map Group 10
Router(ca-certificate-map)# subject-name co Cisco
Router(config-cert-map)# exit
Router(config)# crypto pki trustpoint Access
Router(ca-trustpoint)# match certificate Group
Router(ca-trustpoint)# exit
```

PKI AAA Authorization Using the Entire Subject Name Configuration Example

The following example shows that the entire subject name of the certificate is to be used for PKI AAA authorization:

```
Router(config)# aaa new-model
Router(config)# aaa authorization network tac-o group tacacs+

Router(config)# crypto pki trustpoint test
Router(ca-trustpoint)# enrollment url http://caserver:80
Router(ca-trustpoint)# revocation-check crl
Router(ca-trustpoint)# exit
Router(ca-trustpoint)# authorization list tac-o
Router(ca-trustpoint)# authorization username subjectname all

Router(config)# tacacs-server host 20.2.2.2 key a_secret_key
```

Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example

In the following example, the switch is located in a branch office. The switch uses IP Security (IPsec) to communicate with the main office. Ethernet 1 is the outside interface that connects to the Internet Service Provider (ISP). Ethernet 0 is the interface connected to the LAN of the branch office. To access the CA server located in the main office, the switch must send its IP datagrams out interface Ethernet 1 (address 10.2.2.205) using the IPsec tunnel. Address 10.2.2.205 is assigned by the ISP. Address 10.2.2.205 is not a part of the branch office or main office.

The CA cannot access any address outside the company because of a firewall. The CA sees a message coming from 10.2.2.205 and cannot respond (that is, the CA does not know that the switch is located in a branch office at address 10.1.1.1, which it is able to reach).

Adding the **source interface** command tells the switch to use address 10.1.1.1 as the source address of the IP datagram that it sends to the CA. The CA is able to respond to 10.1.1.1.

This example is configured using the **source interface** command and the interface addresses as described above.

```
Router(config)# crypto pki trustpoint ms-ca
Router(ca-trustpoint)# enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# source interface ethernet0

Router(config)# interface ethernet 0
Router(config-if)# description inside interface
Router(config-if)# ip address 10.1.1.1 255.255.255.0

Router(config)# interface ethernet 1
Router(config-if)# description outside interface
Router(config-if)# ip address 10.2.2.205 255.255.255.0
Router(config-if)# crypto map main-office
```

Persistent Self-Signed Certificates Configuration Examples

The following examples show how to configure a persistent self-signed certificate:

- Trustpoint and Self-Signed Certificate Configuration Example, page 9-63
- Enabling the HTTPS Server Configuration Example, page 9-63

Trustpoint and Self-Signed Certificate Configuration Example

The following example shows how to configure a trustpoint and a self-signed certificate. In this example, a trustpoint named local is declared, its enrollment is requested, and a self-signed certificate with an IP address is generated.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki trustpoint local
Router(ca-trustpoint)# enrollment selfsigned
Router(ca-trustpoint)# end
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Note A router can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server Configuration Example

In the following example, the HTTPS server is enabled and a default trustpoint is generated because one was not previously configured:

```
Router(config)# ip http secure-server

% Generating 1024 bit RSA keys ...[OK]

*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified. Issue "write memory"
to save new certificate

Router(config)#
```

Note You must save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following router reloads.

The following message also appears:

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled

Router(config)#
```

Note Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your access control lists (ACLs) to permit or deny SSH access to the router.

CHAPTER 10

Configuring Advanced VPNs

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

Note The procedures in this chapter assume you have familiarity with security configuration concepts, such as VLANs, ISAKMP policies, preshared keys, transform sets, access control lists, and crypto maps. For more information about these and other security configuration concepts, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

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

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

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

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

Overview of Advanced VPNs

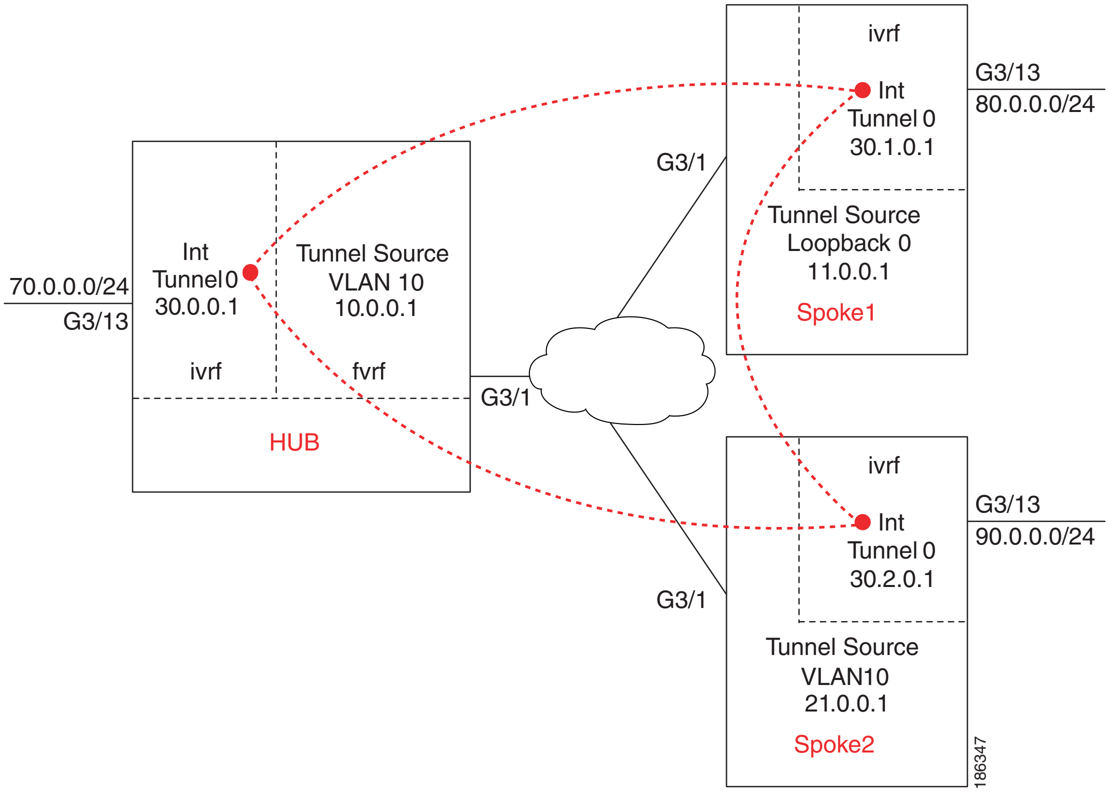
Configuring IP Security (IPsec) Virtual Private Networks (VPNs) in large, complicated networks can be quite complex. This chapter introduces Dynamic Multipoint VPN (DMVPN) and Easy VPN, two features that ease IPsec configuration in advanced environments.

Configuring DMVPN

The DMVPN feature allows users to better scale large and small IPsec VPNs by combining generic routing encapsulation (GRE) tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP).

Figure 10-1 shows an example of a DMVPN configuration with a hub and two spokes.

Figure 10-1 DMVPN Configuration Example



DMVPN Configuration Guidelines and Restrictions

When configuring DMVPN, follow these guidelines and restrictions:

- A tunnel key should not be configured. If a tunnel key is configured, neither the PFC3 or the VSPA will take over the tunnel and the tunnel will be CEF switched.
- GRE tunnels in different Virtual Routing and Forwarding (VRF) instances cannot share the same tunnel source.

- In non-VRF mode, multipoint GRE tunnels should not share the same tunnel source.
- Multicast streaming is not supported across DMVPN on a Catalyst 6500 Series switch. Only multicast packets from a control plane such as routing protocols are supported.
- In a VRF-Aware DMVPN configuration, the **mls mpls tunnel-recir** command must be configured globally on the PE/hub if the CE/DMVPN spokes need to talk to other CEs across the MPLS cloud.
- For the NAT-transparency aware enhancement to work with DMVPN, you must use IPsec transport mode on the transform set. Also, even though NAT-transparency (IKE and IPsec) can support two peers (IKE and IPsec) being translated to the same IP address (using the User Datagram Protocol [UDP] ports to differentiate them [this would be Peer Address Translation]), this functionality is not supported for DMVPN. All DMVPN spokes must have a unique IP address after they have been NAT translated. They can have the same IP address before they are NAT translated.
- If you use the dynamic creation for spoke-to-spoke tunnels benefit of this feature, you must use IKE certificates or wildcard preshared keys for Internet Security Association and Key Management Protocol (ISAKMP) authentication.

Note We recommend that you do not use wildcard preshared keys because access to the entire VPN is compromised if one spoke switch is compromised.

- GRE tunnel keepalive (that is, the **keepalive** command under the GRE interface) is not supported on multipoint GRE tunnels.
- FVRF is not supported on a multipoint GRE (mGRE) tunnel configured on a DMVPN spoke. FVRF is supported on an mGRE tunnel configured on a DMVPN hub.

To enable mGRE and IPsec tunneling for hub and spoke switches, configure your mGRE tunnel for IPsec encryption using the following procedures:

- DMVPN Prerequisites, page 10-3
- Configuring an IPsec Profile, page 10-4
- Configuring the Hub for DMVPN in VRF Mode, page 10-5
- Configuring the Hub for DMVPN in Crypto-Connect Mode, page 10-7
- Configuring the Spoke for DMVPN in VRF Mode, page 10-8
- Configuring the Spoke for DMVPN in Crypto-Connect Mode, page 10-10
- Verifying the DMVPN Configuration, page 10-12
- DMVPN Configuration Examples, page 10-17

For complete configuration information for DMVPN support, refer to this URL:

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

DMVPN Prerequisites

Before configuring an IPsec profile, you must define a transform set by using the **crypto ipsec transform-set** command.

Configuring an IPsec Profile

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

To configure an IPsec profile, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto ipsec profile name`	Defines the IPsec parameters that are to be used for IPsec encryption between "spoke and hub" and "spoke and spoke" switches. This command enters crypto map configuration mode. • *name*—Name of the IPsec profile.
Step 2	`Router(config-crypto-map)# set transform-set transform-set-name`	Specifies which transform sets can be used with the IPsec profile. • *transform-set-name*—Name of the transform set.
Step 3	`Router(config-crypto-map)# set identity`	(Optional) Specifies identity restrictions to be used with the IPsec profile.
Step 4	`Router(config-crypto-map)# set security association lifetime {seconds seconds | kilobytes kilobytes}`	(Optional) Overrides the global lifetime value for the IPsec profile. • *seconds*— Number of seconds a security association will live before expiring. • *kilobytes*— Volume of traffic (in kilobytes) that can pass between IPsec peers using a given security association before that security association expires.
Step 5	`Router(config-crypto-map)# set pfs [group1 | group2]`	(Optional) Specifies that IP Security should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPsec profile. If this command is not specified, the default (group1) will be enabled. • **group1**—(Optional) Specifies that IPsec should use the 768-bit Diffie-Hellman (DH) prime modulus group when performing the new DH exchange. • **group2**—(Optional) Specifies the 1024-bit DH prime modulus group.

Configuring the Hub for DMVPN in VRF Mode

In VPN routing and forwarding instance (VRF) mode, to configure the hub switch for mGRE and IPsec integration (that is, to associate the tunnel with the IPsec profile configured in the previous procedure), perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface tunnel tunnel-number`	Configures a tunnel interface and enters interface configuration mode. • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	`Router(config-if)# ip vrf forwarding inside-vrf-name`	(Optional) Associates a VRF with an interface or subinterface. This step is required only when configuring an inside VRF. • *inside-vrf-name*—Name assigned to the VRF.
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.
Step 4	`Router(config-if)# ip mtu bytes`	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.
Step 5	`Router(config-if)# ip nhrp authentication string`	Configures the authentication string for an interface using the Next Hop Resolution Protocol (NHRP). • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 6	`Router(config-if)# ip nhrp map multicast dynamic`	Allows NHRP to automatically add spoke switches to the multicast NHRP mappings.
Step 7	`Router(config-if)# ip nhrp network-id number`	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 8	`Router(config-if)# tunnel source {ip-address | type number}`	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number (for example, VLAN 2).

	Command	Purpose
Step 9	`Router(config-if)#` **`tunnel mode gre multipoint`**	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 10	`Router(config-if)#` **`tunnel vrf`** *`front-door-vrf-name`*	(Optional) Associates a VRF instance with a specific tunnel destination, interface, or subinterface. This step is required only when configuring a front door VRF (FVRF). • *front-door-vrf-name*—Name assigned to the VRF. This may or may not be the same as the *inside-vrf-name.*
Step 11	`Router(config-if)#` **`tunnel protection ipsec profile`** *`name`*	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.
Step 12	`Router(config-if)#` **`crypto engine slot`** *`slot/subslot`* **`inside`**	Assigns the specified crypto engine to the inside interface. • *slot/subslot*—The slot where the VSPA is located.
Step 13	`Router(config-if)#` **`interface`** *`type slot/subslot/port`*	Configures the DMVPN physical egress interface.
Step 14	`Router(config-if)#` **`ip vrf forwarding`** *`front-door-vrf-name`*	(Optional) Associates a VRF with an interface or subinterface. This step is required only when configuring a front door VRF (FVRF). • *front-door-vrf-name*—Name assigned to the VRF. This is the same name used in Step 10.
Step 15	`Router(config-if)#` **`ip address`** *`address mask`*	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 16	`Router(config-if)#` **`crypto engine outside`**	Enables the crypto engine on the interface.

Configuring the Hub for DMVPN in Crypto-Connect Mode

In crypto-connect mode, to configure the hub switch for mGRE and IPsec integration (that is, to associate the tunnel with the IPsec profile configured in the previous procedure), perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`interface tunnel`** *`tunnel-number`*	Configures a tunnel interface and enters interface configuration mode • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	`Router(config-if)# `**`ip address`** *`ip-address mask`* `[`*`secondary`*`]`	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.
Step 3	`Router(config-if)# `**`ip mtu`** *`bytes`*	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.
Step 4	`Router(config-if)# `**`ip nhrp authentication`** *`string`*	Configures the authentication string for an interface using the Next Hop Resolution Protocol (NHRP). • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 5	`Router(config-if)# `**`ip nhrp map multicast dynamic`**	Allows NHRP to automatically add spoke switches to the multicast NHRP mappings.
Step 6	`Router(config-if)# `**`ip nhrp network-id`** *`number`*	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 7	`Router(config-if)# `**`tunnel source`** `{`*`ip-address`* `|` *`type number`*`}`	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number (for example, VLAN 2).
Step 8	`Router(config-if)# `**`tunnel mode gre multipoint`**	Sets the encapsulation mode to mGRE for the tunnel interface.

	Command	Purpose
Step 9	`Router(config-if)# tunnel protection ipsec profile name`	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.
Step 10	`Router(config-if)# crypto engine slot slot/subslot`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the VSPA is located.
Step 11	`Router(config)# interface vlan ifvlan`	Configures the DMVPN inside VLAN.
Step 12	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 7. • *mask*—Subnet mask.
Step 13	`Router(config-if)# crypto engine slot slot/subslot`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot and subslot where the VSPA is located.
Step 14	`Router(config-if)# interface type slot/subslot/port`	Configures the DMVPN physical egress interface.
Step 15	`Router(config-if)# no ip address`	Assigns no IP address to the interface.
Step 16	`Router(config-if)# crypto connect vlan ifvlan`	Connects the outside access port VLAN to the inside (crypto) interface VLAN and enters crypto-connect mode. • *ifvlan*—DMVPN inside VLAN identifier.

Configuring the Spoke for DMVPN in VRF Mode

In VRF mode, to configure spoke switches for mGRE and IPsec integration, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface tunnel tunnel-number`	Configures a tunnel interface and enters interface configuration mode. • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	`Router(config-if)# ip vrf forwarding inside-vrf-name`	(Optional) Associates a VRF with an interface or subinterface. This step is required only when configuring an inside VRF. • *inside-vrf-name*—Name assigned to the VRF.

	Command	Purpose
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.
Step 4	`Router(config-if)# ip mtu bytes`	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.
Step 5	`Router(config-if)# ip nhrp authentication string`	Configures the authentication string for an interface using NHRP. • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 6	`Router(config-if)# ip nhrp map hub-tunnel-ip-address hub-physical-ip-address`	Statically configures the IP-to-NonBroadcast MultiAccess (NBMA) address mapping of IP destinations connected to an NBMA network. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 7	`Router(config-if)# ip nhrp map multicast hub-physical-ip-address`	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub switch. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 8	`Router(config-if)# ip nhrp nhs hub-tunnel-ip-address`	Configures the hub switch as the NHRP next-hop server. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub.
Step 9	`Router(config-if)# ip nhrp network-id number`	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 10	`Router(config-if)# tunnel source {ip-address | type number}`	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number; for example, VLAN 2.

	Command	Purpose
Step 11	`Router(config-if)# tunnel mode gre multipoint`	Sets the encapsulation mode to mGRE for the tunnel interface. Use this command if data traffic can use dynamic spoke-to-spoke traffic.
Step 12	`Router(config-if)# tunnel protection ipsec profile name`	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.
Step 13	`Router(config-if)# crypto engine slot slot/subslot inside`	Assigns the specified crypto engine to the inside interface. • *slot/subslot*—The slot where the VSPA is located.
Step 14	`Router(config-if)# interface type slot/subslot/port`	Configures the DMVPN physical egress interface.
Step 15	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 16	`Router(config-if)# crypto engine outside`	Enables the crypto engine on the interface.

Configuring the Spoke for DMVPN in Crypto-Connect Mode

In crypto-connect mode, to configure spoke switches for mGRE and IPsec integration, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface tunnel tunnel-number`	Configures a tunnel interface and enters interface configuration mode • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.
Step 3	`Router(config-if)# ip mtu bytes`	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.

	Command	Purpose
Step 4	Router(config-if)# **ip nhrp authentication** *string*	Configures the authentication string for an interface using NHRP. • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 5	Router(config-if)# **ip nhrp map** *hub-tunnel-ip-address hub-physical-ip-address*	Statically configures the IP-to-NonBroadcast MultiAccess (NBMA) address mapping of IP destinations connected to an NBMA network. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 6	Router(config-if)# **ip nhrp map multicast** *hub-physical-ip-address*	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub switch. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 7	Router(config-if)# **ip nhrp nhs** *hub-tunnel-ip-address*	Configures the hub switch as the NHRP next-hop server. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub.
Step 8	Router(config-if)# **ip nhrp network-id** *number*	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 9	Router(config-if)# **tunnel source** {*ip-address* | *type number*}	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number; for example, VLAN 2.
Step 10	Router(config-if)# **tunnel mode gre multipoint**	Sets the encapsulation mode to mGRE for the tunnel interface. Use this command if data traffic can use dynamic spoke-to-spoke traffic.
Step 11	Router(config-if)# **tunnel protection ipsec profile** *name*	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.
Step 12	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the VSPA is located.

	Command	Purpose
Step 13	`Router(config)# interface vlan ifvlan`	Configures the DMVPN inside VLAN.
Step 14	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 7. • *mask*—Subnet mask.
Step 15	`Router(config-if)# crypto engine slot slot/subslot`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the VSPA is located.
Step 16	`Router(config-if)# interface type slot/subslot/port`	Configures the DMVPN physical egress interface.
Step 17	`Router(config-if)# no ip address`	Assigns no IP address to the interface.
Step 18	`Router(config-if)# crypto connect vlan ifvlan`	Connects the outside access port VLAN to the inside (crypto) interface VLAN and enters crypto-connect mode. • *ifvlan*—DMVPN inside VLAN identifier.

Verifying the DMVPN Configuration

To verify that your DMVPN configuration is working, use the **show crypto isakmp sa**, **show crypto map**, and **show ip nhrp** commands.

The **show crypto isakmp sa** command displays all current IKE security associations (SAs) at a peer.

The following sample output is displayed after IKE negotiations have successfully completed between a hub and two spokes and between the two spokes, as shown in Figure 10-1 on page 10-2:

```
HUB# show crypto isakmp sa
dst             src             state          conn-id slot status
10.0.0.1        11.0.0.1        QM_IDLE          68001 ACTIVE
10.0.0.1        21.0.0.1        QM_IDLE          68002 ACTIVE


SPOKE1# show crypto isakmp sa
dst             src             state          conn-id slot status
11.0.0.1        21.0.0.1        QM_IDLE          68002 ACTIVE
21.0.0.1        11.0.0.1        QM_IDLE          68003 ACTIVE
10.0.0.1        11.0.0.1        QM_IDLE          68001 ACTIVE

SPOKE2# show crypto isakmp sa
dst             src             state          conn-id slot status
10.0.0.1        21.0.0.1        QM_IDLE          68001 ACTIVE
11.0.0.1        21.0.0.1        QM_IDLE          68003 ACTIVE
21.0.0.1        11.0.0.1        QM_IDLE          68002 ACTIVE
```

The **show crypto map** command displays the crypto map configuration.

The following sample output is displayed after a crypto map has been configured:

```
HUB# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: VPN-PROF
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 11.0.0.1
        Extended IP access list
            access-list  permit gre host 10.0.0.1 host 11.0.0.1
        Current peer: 11.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65538 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 21.0.0.1
        Extended IP access list
            access-list  permit gre host 10.0.0.1 host 21.0.0.1
        Current peer: 21.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
 using crypto engine WS-IPSEC-3[4/0]



SPOKE1# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: VPN-PROF
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 10.0.0.1
        Extended IP access list
            access-list  permit gre host 11.0.0.1 host 10.0.0.1
        Current peer: 10.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65538 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 21.0.0.1
        Extended IP access list
            access-list  permit gre host 11.0.0.1 host 21.0.0.1
        Current peer: 21.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
 using crypto engine WS-IPSEC-3[4/0]



SPOKE2# **show crypto map**
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: VPN-PROF
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 10.0.0.1
        Extended IP access list
            access-list  permit gre host 21.0.0.1 host 10.0.0.1
        Current peer: 10.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65538 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 11.0.0.1
        Extended IP access list
            access-list  permit gre host 21.0.0.1 host 11.0.0.1
        Current peer: 11.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
 using crypto engine WS-IPSEC-3[4/0]
```

The **show ip nhrp** command displays the NHRP cache.

The following sample output shows that NHRP registration occurred. Note that NHRP between the hub and a spoke is static, while NHRP between spokes is dynamic:

```
Router# **show ip nhrp**
HUB# **show ip nhrp**
30.1.0.1/32 via 30.1.0.1, Tunnel0 created 00:18:13, expire 01:41:46
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 11.0.0.1
30.2.0.1/32 via 30.2.0.1, Tunnel0 created 00:11:55, expire 01:48:04
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 21.0.0.1


SPOKE1# **show ip nhrp**
30.0.0.1/32 via 30.0.0.1, Tunnel0 created 00:23:39, never expire
  Type: static, Flags: authoritative used
  NBMA address: 10.0.0.1
30.2.0.1/32 via 30.2.0.1, Tunnel0 created 00:04:27, expire 01:47:59
  Type: dynamic, Flags: router
  NBMA address: 21.0.0.1


SPOKE2# show ip nhrp
30.0.0.1/32 via 30.0.0.1, Tunnel0 created 00:12:02, never expire
  Type: static, Flags: authoritative used
  NBMA address: 10.0.0.1
30.1.0.1/32 via 30.1.0.1, Tunnel0 created 00:04:29, expire 01:41:40
  Type: dynamic, Flags: router
  NBMA address: 11.0.0.1
```

For DMVPN configuration examples, see the "DMVPN Configuration Examples" section on page 10-17.

Configuring the Easy VPN Server

The Easy VPN server provides server support for the Cisco VPN Client Release 4.x and later software clients and Cisco VPN hardware clients. The feature allows a remote end user to communicate using IP Security (IPsec) with any Cisco IOS Virtual Private Network (VPN) gateway. Centrally managed IPsec policies are pushed to the client by the server, minimizing configuration by the end user.

The Easy VPN server features include:

- RADIUS support for user profiles
- User-based policy control
- Session monitoring for VPN group access
- **backup-gateway** command
- **pfs** command

Easy VPN Server Configuration Guidelines and Restrictions

When configuring the Easy VPN server, follow these guidelines and restrictions:

- The following IPsec protocol options and attributes currently are not supported by Cisco VPN clients, so these options and attributes should not be configured on the switch for these clients:
  - Authentication with public key encryption
  - Digital Signature Standard (DSS)
  - Diffie-Hellman (DH) groups (1)
  - IPsec Protocol Identifier (IPSEC_AH)
  - IPsec Protocol Mode (Transport mode)
  - Manual keys
  - Perfect Forward Secrecy (PFS)
- Enhanced Easy VPN, which uses Dynamic Virtual Tunnel Interfaces (DVTI) instead of dynamic crypto maps, is not supported.

For complete configuration information about the Easy VPN Server feature and the enhancements, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftunity.html

Configuring the Easy VPN Remote

The Easy VPN remote feature allows Cisco routers and security appliances to establish a site-to-site VPN connection to a Cisco Easy VPN Server without complex remote-side configuration. Centrally managed IPsec policies are pushed to the client by the server, minimizing configuration by the end user.

Easy VPN Remote Configuration Guidelines

Follow these guidelines when configuring Easy VPN for the VSPA:

Caution You must clear all other crypto configurations from your running configuration on the Cisco IOS-based Easy VPN client that you are using to connect to the VSPA. If an ISAKMP policy is configured, it takes precedence over the preinstalled Easy VPN ISAKMP policies and the connection will fail. Other clients such as the VPN3000 and PIX systems running Easy VPN will prevent you from configuring Easy VPN unless all crypto configurations are removed. For complete configuration information for Easy VPN client support, refer to this URL:

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

For an Easy VPN server configuration example, see the "Easy VPN Server (Router Side) Configuration Example" section on page 10-21.

Configuring Easy VPN Remote RSA Signature Storage

The Easy VPN remote RSA signature support feature provides for the support of Rivest, Shamir, and Adelman (RSA) signatures on Easy VPN remote devices. The support is provided through RSA certificates that can be stored on or off the remote device.

Easy VPN Remote RSA Signature Support Configuration Guidelines and Restrictions

When configuring Easy VPN remote RSA signature support, follow these guidelines and restrictions:

- You must have a Cisco Virtual Private Network (VPN) remote device and be familiar with configuring the device.
- You must have a certificate authority (CA) available to your network before you configure this interoperability feature. The CA must support the public key infrastructure (PKI) protocol of Cisco Systems, which is the Simple Certificate Enrollment Protocol (SCEP) (formerly called Certificate Enrollment Protocol [CEP]).
- This feature should be configured only when you also configure both IPsec and Internet Key Exchange (IKE) in your network.
- The Cisco IOS software does not support CA server public keys greater than 2048 bits.

Configuring Easy VPN Remote RSA Signature Support

The RSA signatures for an Easy VPN remote device are configured the same way that you would configure RSA signatures for any other Cisco device.

For information about configuring RSA signatures, refer to the *Cisco IOS Security Configuration Guide*.

To enable the RSA signatures, when you are configuring the Easy VPN remote and assigning the configuration to the outgoing interface, you must omit the **group** command. The content of the first Organizational Unit (OU) field will be used as the group.

For information about configuring Cisco Easy VPN remote devices, refer to the feature document, *Easy VPN Remote RSA Signature Support*, at the following location:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtevcrsa.html

Configuration Examples

This section provides examples of the following configurations:

- DMVPN Configuration Examples, page 10-17
- Easy VPN Server (Router Side) Configuration Example, page 10-21

DMVPN Configuration Examples

The following sections provide examples of DMVPN configuration:

- DMVPN Hub with VRF Mode Configuration Example, page 10-18
- DMVPN Spoke with VRF Mode Configuration Example, page 10-19
- DMVPN Spoke with Crypto-Connect Mode Configuration Example, page 10-20

The DMVPN examples are based on the implementation shown in Figure 10-1 on page 10-2, using the following configuration parameters:

- The hub switch (HUB) is configured in VRF mode with inside VRF (IVRF) and front-door VRF (FVRF)
- One spoke switch (SPOKE1) is configured in VRF mode with IVRF but no FVRF
- One spoke switch (SPOKE2) is configured in crypto connect mode
- EIGRP is configured to distribute routes over the tunnels
- In all switches, interface gi3/1 is the interface to the provider network
- In all switches, interface gi3/13 is the interface to the private LAN

Note The tunnel source can be the same as the physical egress port. If the tunnel source is not the physical egress port, make sure that traffic to and from the tunnel source passes through the physical egress port.

DMVPN Hub with VRF Mode Configuration Example

The following is a configuration example of the VSPA serving as a DMVPN hub using VRF mode with inside VRF and front-door VRF (FVRF):

```
hostname HUB
!
ip vrf fvrf
 rd 1000:1
!
ip vrf ivrf
 rd 1:1
!
crypto engine mode vrf
!
crypto keyring RING1 vrf fvrf
  pre-shared-key address 0.0.0.0 0.0.0.0 key abcdef
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
!
crypto ipsec transform-set ts esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile VPN-PROF
 set transform-set ts
!
!
interface Tunnel0
! EIGRP uses the configured bandwidth to allocate bandwidth for its routing update 
mechanism
 bandwidth 1000000
 ip vrf forwarding ivrf
 ip address 30.0.0.1 255.0.0.0
 ip nhrp authentication cisco123
 ip nhrp map multicast dynamic
 ip nhrp network-id 1000
! For a large number of tunnels, the following two commands are recommended
! EIGRP timers are adjusted to match the default timers for a WAN interface
 ip hello-interval eigrp 200 60
 ip hold-time eigrp 200 180
! The following two EIGRP commands are necessary to allow spoke-to-spoke communication
 no ip next-hop-self eigrp 200
 no ip split-horizon eigrp 200
 tunnel source Vlan10
 tunnel mode gre multipoint
 tunnel vrf fvrf
 tunnel protection ipsec profile VPN-PROF
 crypto engine slot 4/0 inside
!
interface Vlan10
 ip vrf forwarding fvrf
 ip address 10.0.0.1 255.255.255.0
 crypto engine outside
!
interface GigabitEthernet3/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10
 switchport mode trunk
```

```
interface GigabitEthernet3/13
 description Local LAN interface
 ip vrf forwarding ivrf
 ip address 70.0.0.1 255.255.255.0

router eigrp 10
 no auto-summary
 !
 address-family ipv4 vrf ivrf
 redistribute connected
 network 30.0.0.0
 network 70.0.0.0
 no auto-summary
 autonomous-system 200
 exit-address-family
!
! In this example, tunnel destination reachability is provided by static routes
! A routing protocol could also be used
ip route vrf fvrf 11.0.0.0 255.0.0.0 10.0.0.2
ip route vrf fvrf 21.0.0.0 255.0.0.0 10.0.0.2

end
```

DMVPN Spoke with VRF Mode Configuration Example

The following is a configuration example of the VSPA serving as a DMVPN spoke using VRF mode with inside VRF but no front-door VRF:

```
hostname SPOKE1
!
ip vrf ivrf
 rd 1:1
!
crypto engine mode vrf
!
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key abcdef address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 60
!
!
crypto ipsec transform-set ts esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile VPN-PROF
 set transform-set ts
!
interface Tunnel0
 bandwidth 100000
 ip vrf forwarding ivrf
 ip address 30.1.0.1 255.0.0.0
 ip nhrp authentication cisco123
 ip nhrp map 30.0.0.1 10.0.0.1
 ip nhrp map multicast 10.0.0.1
 ip nhrp network-id 1000
 ip nhrp nhs 30.0.0.1
 ip hello-interval eigrp 200 60
 ip hold-time eigrp 200 180
 tunnel source Loopback0
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN-PROF
 crypto engine slot 4/0 inside
!
interface Loopback0
 ip address 11.0.0.1 255.255.255.0
!

interface GigabitEthernet3/1
 ip address 11.255.255.1 255.255.255.0
 crypto engine outside
!
interface GigabitEthernet3/13
 ip vrf forwarding ivrf
 ip address 80.0.0.1 255.255.255.0

router eigrp 10
 no auto-summary
 !
 address-family ipv4 vrf ivrf
 autonomous-system 200
 network 30.0.0.0
 network 70.0.0.0
 no auto-summary
 redistribute connected
 exit-address-family

ip route 10.0.0.0 255.0.0.0 11.255.255.2
ip route 21.0.0.0 255.0.0.0 11.255.255.2

end
```

DMVPN Spoke with Crypto-Connect Mode Configuration Example

The following is a configuration example of the VSPA serving as a DMVPN spoke using crypto-connect mode:

```
hostname SPOKE2
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key abcdef address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 60
!
!
crypto ipsec transform-set ts esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile VPN-PROF
 set transform-set ts
!
interface Tunnel0
 bandwidth 1000000
 ip address 30.2.0.1 255.0.0.0
 ip nhrp authentication cisco123
 ip nhrp map 30.0.0.1 10.0.0.1
 ip nhrp map multicast 10.0.0.1
 ip nhrp network-id 1000
 ip nhrp nhs 30.0.0.1
 ip hello-interval eigrp 200 60
 ip hold-time eigrp 200 180
 tunnel source Vlan10
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN-PROF
 crypto engine slot 4/0 inside
!
interface Vlan10
 ip address 21.0.0.1 255.255.255.0
 no mop enabled
 crypto engine slot 4/0 inside
!
interface GigabitEthernet3/1
 no ip address
 crypto connect vlan 10
!
interface GigabitEthernet3/13
 ip address 90.0.0.1 255.255.255.0
!
router eigrp 200
 redistribute connected
 network 30.0.0.0
 network 90.0.0.0
 no auto-summary

ip route 10.0.0.0 255.0.0.0 21.0.0.2
ip route 11.0.0.0 255.0.0.0 21.0.0.2

end
```

Easy VPN Server (Router Side) Configuration Example

The following is an example of an Easy VPN server router-side configuration:

```
!
version 12.2
!
hostname sanjose
!
logging snmp-authfail
logging buffered 1000000 debugging
aaa new-model
aaa authentication login authen local
aaa authorization network author local
!
username unity password 0 uc
ip subnet-zero
no ip source-route
!
mpls ldp logging neighbor-changes
mls flow ip destination
mls flow ipx destination
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 10 2
```

```
!
crypto isakmp client configuration group group1
 key 12345
 domain cisco.com
 pool pool1
!
crypto isakmp client configuration group default
 key 12345
 domain cisco.com
 pool pool2
!
crypto ipsec transform-set myset3 esp-3des esp-md5-hmac
!
crypto dynamic-map test_dyn 1
 set transform-set myset3
 reverse-route
!
! Static client mapping
crypto map testtag client authentication list authen
crypto map testtag isakmp authorization list author
crypto map testtag client configuration address respond
crypto map testtag 10 ipsec-isakmp
 set peer 10.5.1.4
 set security-association lifetime seconds 900
 set transform-set myset3
 match address 109
!
! Dynamic client mapping
crypto map test_dyn client authentication list authen
crypto map test_dyn isakmp authorization list author
crypto map test_dyn client configuration address respond
crypto map test_dyn 1 ipsec-isakmp dynamic test_dyn
!
!
no spanning-tree vlan 513
!
redundancy
  main-cpu
  auto-sync running-config
  auto-sync standard
!
interface GigabitEthernet2/1
 no ip address
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,513,1002-1005
 switchport mode trunk
!
interface GigabitEthernet2/2
 no ip address
 shutdown
!
interface GigabitEthernet6/1/1
 no ip address
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,513,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet6/1/2
 no ip address
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 cdp enable
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan2
 no ip address
 crypto connect vlan 513
!
interface Vlan513
 ip address 10.5.1.1 255.255.0.0
 crypto map test_dyn
 crypto engine slot 6/1 inside
!
ip local pool pool1 22.0.0.2
ip local pool pool2 23.0.0.3
ip classless
ip pim bidir-enable
!
access-list 109 permit ip host 10.5.1.1 host 22.0.0.2
arp 127.0.0.12 0000.2100.0000 ARPA
!
snmp-server enable traps tty
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
!
line con 0
line vty 0 4
 password lab
 transport input lat pad mop telnet rlogin udptn nasi
!
end
```

CHAPTER 11

Configuring Duplicate Hardware and IPsec Failover

This chapter provides information about configuring duplicate hardware and IPsec failover using the VSPA on the Catalyst 6500 Series switch. It includes the following sections:

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

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

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

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

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

Overview of Duplicate Hardware Configurations and IPsec Failover

For critical VPN communications, you can deploy redundant VPN hardware and configure your system for failover in case of hardware failure. The following topics provide information about configuring for IPsec failover using the VSPA:

- Configuring Multiple VSPAs in a Chassis, page 11-2
- Understanding Stateless Failover Using HSRP, page 11-3
- IPsec Stateless Failover Configuration Guidelines and Restrictions, page 11-3

Configuring Multiple VSPAs in a Chassis

You can deploy up to ten VSPAs in a single chassis, with the restriction that no more than one VSPA can be used to perform IPsec services for any given interface VLAN.

Multiple VSPAs in a Chassis Configuration Guidelines

When configuring multiple VSPAs in a chassis, follow these guidelines:

- Using the **no switchport** command followed by the **switchport** command readds all VLANs to a trunk port (this situation occurs when you are first switching to a routed port and then back to a switch port). For detailed information on configuring trunk ports, see "Configuring a Trunk Port" section on page 3-14 of Chapter 3, "Configuring VPNs in Crypto-Connect Mode."
- As with single VSPA deployments, you must properly configure each VSPA's inside and outside port. You can add an interface VLAN only to the inside port of one VSPA. Do not add the same interface VLAN to the inside port of more than one VSPA.

  Assigning interface VLANs to the inside ports of the VSPAs allows you to decide which VSPA can be used to provide IPsec services for a particular interface VLAN.

  Note You do not need to explicitly add interface VLANs to the inside trunk ports of the VSPAs. Entering the **crypto engine slot** command achieves the same results.

  Note There is no support for using more than one VSPA to do IPsec processing for a single interface VLAN.

- SA-based load balancing is not supported.
- If you assign the same crypto map to multiple interfaces, then you must use the **crypto map local address** command, and all interfaces must be assigned to the same crypto engine.

For a configuration example of multiple VSPAs in a chassis, see the "Multiple VSPAs in a Chassis Configuration Example" section on page 11-12.

Understanding Stateless Failover Using HSRP

The IPsec failover (VPN high availability) feature allows you to employ a secondary (standby) switch that automatically takes over the primary (active) switch's tasks in the event of an active switch failure. IPsec failover, stateless or stateful, is designed to work in conjunction with the Hot Standby Routing Protocol (HSRP) and Reverse Route Injection (RRI).

HSRP is used between the active and standby switch in either stateless or stateful mode, tracking the state of switch interfaces and providing a failover mechanism between primary and secondary devices. An HSRP group shares a single virtual IP address as its crypto peer address so that the remote crypto peer requires no reconfiguration after a failover. The configured HSRP timers determine the time that it takes for the standby switch to take over.

RRI uses information derived from the negotiated IPsec SAs to create static routes to the networks identified in those SAs. During an HSRP and IPsec failover, RRI allows dynamic routing information updates.

In an IPsec stateless failover, the HSRP group's virtual IP address transfers over to the standby switch, but no IPsec or ISAKMP SA state information is transferred to the standby switch. The remote crypto peer detects the failure using Dead Peer Detection (DPD) or a keepalive mechanism. The remote crypto peer then communicates with the standby switch at the HSRP group address to renegotiate the dropped ISAKMP SAs and IPsec SAs before traffic transmission can resume.

When used together, HSRP and RRI provide a reliable network design for VPNs and reduce configuration complexity on remote peers.

For complete HSRP configuration information, refer to this URL:

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp_ps6922_TSD_Products_Configuration_Guide_Chapter.html

IPsec Stateless Failover Configuration Guidelines and Restrictions

When configuring IPsec stateless failover, follow these guidelines and restrictions:

- When configuring IPsec stateless failover with the VSPA, all VSPA configuration rules apply. You must apply crypto maps to interface VLANs.
- The recommended HSRP timer values are one second for hello timers and three seconds for hold timers. These values should prevent an undesirable failover that is caused by temporary network congestion or transient high CPU loads.

  These timer values can be adjusted upward if you are running high loads or have a large number of HSRP groups. Temporary failures and load-related system stability can be positively affected by raising the timer values as needed. The hello timer value should be approximately a third of the hold timer value.
- The **standby preempt** command is required, and should be configured with no **priority** or **delay** options.
- To allow dynamic routing information updates during the HSRP and IPsec failover, enable the Reverse Route Injection (RRI) feature using the **reverse-route** command.
- To verify that all processes are running properly after enabling HSRP, use the **show standby** command.

- The following features are not supported with IPsec stateless failover:
  - The **standby use-bia** command—Always use a virtual HSRP MAC address for the switch's MAC address.
  - DMVPN or tunnel protection.
  - Secured WAN ports (for example, IPsec over FlexWAN or SIP module port adapters)— This restriction is due to limitations of HSRP.

Configuring IPsec Stateless Failover

Note IPsec stateful failover is supported only within a chassis using a blade failure group, as described in "Configuring Intrachassis IPsec Stateful Failover Using a Blade Failure Group" section on page 11-10. Inter-chassis stateful failover is not supported.

The following sections describe how to configure IPsec stateless failover in crypto-connect and VRF modes:

- Configuring IPsec Stateless Failover Using HSRP with Crypto-Connect Mode, page 11-4
- Configuring IPsec Stateless Failover with VRF Mode, page 11-10

Configuring IPsec Stateless Failover Using HSRP with Crypto-Connect Mode

To configure IP stateless failover using HSRP, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp policy** *priority* ... Router(config-isakmp) # **exit**	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest. For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.
Step 2	Router(config)# **crypto isakmp key** *keystring* **address** *peer-address*	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer-address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 3	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 4	Router(config)# **access-list** *access-list-number* {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • **ip** *source*—Address of the host from which the packet is being sent. • *source-wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination-wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.
Step 5	Router(config)# **crypto dynamic-map** *dynamic-map-name seq-number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a dynamic crypto map template and enters the crypto map configuration mode. • *dynamic-map-name*—Name that identifies the dynamic crypto map template. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 6	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp dynamic** *dynamic-map-name*	Creates a crypto map entry and binds it to the dynamic crypto map template. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. • *dynamic-map-name*—Name that identifies the dynamic crypto map template.
Step 7	Router(config-if)# **interface gigabitethernet** *slot*/*subslot*/*port*	Enters interface configuration mode for the LAN-side Gigabit Ethernet interface.
Step 8	Router(config-if)# **ip address** *address mask*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 9	Router(config-if)# **standby** [*group-number*] **ip** *ip-address*	Enables the HSRP. • *group-number*—(Optional) Group number on the interface for which HSRP is being activated. The default is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2. • *ip-address*—IP address of the standby switch interface.
Step 10	Router(config-if)# **standby** [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*	Configures the time between hello packets and the hold time before other switches declare the active switch to be down. • *group-number*—(Optional) Group number to which the timers apply. • **msec**—(Optional) Interval in milliseconds. Millisecond timers allow for faster failover. • *hellotime*—Hello interval (in seconds). This is an integer from 1 to 254. The default is 3 seconds. If the **msec** option is specified, *hellotime* is in milliseconds. This is an integer from 15 to 999. • *holdtime*—Time (in seconds) before the active or standby switch is declared to be down. This is an integer from x to 255. The default is 10 seconds. If the **msec** option is specified, *holdtime* is in milliseconds. This is an integer from y to 3000.

	Command	Purpose
Step 11	Router(config-if)# **standby** [*group-number*] [**priority** *priority*] **preempt** [**delay** [**minimum** | **sync**] *seconds*]	Sets the standby priority used in choosing the active switch. • *group-number*—(Optional) Group number to which the priority applies. • *priority*—(Optional) The priority value range is from 1 to 255, where 1 denotes the lowest priority and 255 denotes the highest priority. Specify that, if the local switch has priority over the current active switch, the local switch should attempt to take its place as the active switch. • **delay**—(Optional) Specifies a preemption delay, after which the Hot Standby switch preempts and becomes the active switch. • **minimum**—(Optional) Specifies the minimum delay period in seconds. • **sync**—(Optional) Specifies the maximum synchronization period for IP redundancy clients in seconds. • *seconds*—(Optional) Causes the local switch to postpone taking over the active role for a minimum number of seconds since that switch was last restarted. The range is from 0 to 3600 seconds (1 hour). The default is 0 seconds (no delay).
Step 12	Router(config-if)# **standby** [*group-number*] **track** *type number* [*interface-priority*]	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the device's Hot Standby priority is lowered. • *group-number*—(Optional) Group number on the interface for which HSRP is being activated. • *type*—Interface type (combined with interface number) that will be tracked. • *number*—Interface number (combined with interface type) that will be tracked. • *interface-priority*—(Optional) Amount by which the Hot Standby priority for the switch is decremented (or incremented) when the interface goes down (or comes back up). Range is from 0 to 255. Default is 10.
Step 13	Router(config-if)# **standby** [*group-number*] *name*	Configures the standby group name for the interface. • *group-number*—(Optional) Group number to which the name is being applied. • *name*—Name of the HSRP standby group.
Step 14	Router(config-if)# **interface vlan** *vlan_ID*	Enters interface configuration mode for the specified crypto interface VLAN.

	Command	Purpose
Step 15	`Router(config-if)# ip address address mask`	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 16	`Router(config-if)# standby [group-number] ip ip-address`	Enables the HSRP. • *group-number*—(Optional) Group number on the interface for which HSRP is being activated. The default is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2. • *ip-address*—Virtual IP address of the HSRP standby group.
Step 17	`Router(config-if)# standby [group-number] timers [msec] hellotime [msec] holdtime`	Configures the time between hello packets and the hold time before other switches declare the active switch to be down. • *group-number*—(Optional) Group number to which the timers apply. • **msec**—(Optional) Interval in milliseconds. Millisecond timers allow for faster failover. • *hellotime*—Hello interval (in seconds). This is an integer from 1 to 254. The default is 3 seconds. If the **msec** option is specified, *hellotime* is in milliseconds. This is an integer from 15 to 999. • *holdtime*—Time (in seconds) before the active or standby switch is declared to be down. This is an integer from x to 255. The default is 10 seconds. If the **msec** option is specified, *holdtime* is in milliseconds. This is an integer from y to 3000.

	Command	Purpose
Step 18	Router(config-if)# **standby** [*group-number*] [**priority** *priority*] **preempt** [**delay** [**minimum** | **sync**] *seconds*]	Sets the standby priority used in choosing the active switch. • *group-number*—(Optional) Group number to which the priority applies. • *priority*—(Optional) The priority value range is from 1 to 255, where 1 denotes the lowest priority and 255 denotes the highest priority. Specify that, if the local switch has priority over the current active switch, the local switch should attempt to take its place as the active switch. • **delay**—(Optional) Specifies a preemption delay, after which the hot standby switch preempts and becomes the active switch. • **minimum**—(Optional) Specifies the minimum delay period in seconds. • **sync**—(Optional) Specifies the maximum synchronization period for IP redundancy clients in seconds. • *seconds*—(Optional) Causes the local switch to postpone taking over the active role for a minimum number of seconds since that switch was last restarted. The range is from 0 to 3600 seconds (1 hour). The default is 0 seconds (no delay).
Step 19	Router(config-if)# **standby** [*group-number*] **track** *type number* [*interface-priority*]	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the device's hot standby priority is lowered. • *group-number*—(Optional) Group number on the interface for which HSRP is being activated. • *type*—Interface type (combined with interface number) that will be tracked. • *number*—Interface number (combined with interface type) that will be tracked. • *interface-priority*—(Optional) Amount by which the Hot Standby priority for the switch is decremented (or incremented) when the interface goes down (or comes back up). Range is from 0 to 255. Default is 10.
Step 20	Router(config-if)# **standby** [*group-number*] *name*	Configures the standby group name for the interface. • *group-number*—(Optional) Group number to which the name is being applied. • *name*—Name of the HSRP standby group.

	Command	Purpose
Step 21	`Router(config-if)# `**`crypto map`** *`map-name`* **`redundancy`** *`name`*	Defines a backup IPsec peer. Both routers in the standby group are defined by the redundancy standby name and share the same virtual IP address. • *map-name*—Name of the crypto map set. • *name*—Name of the HSRP standby group.
Step 22	`Router(config-if)# `**`crypto engine slot`** *`slot`*	Assigns the crypto engine to the inside (crypto) interface VLAN. • *slot*—The slot where the VSPA is located.
Step 23	`Router(config-if)# `**`interface gigabitethernet`** *`slot/subslot/port`*	Enters interface configuration mode for the outside Gigabit Ethernet interface.
Step 24	`Router(config-if)# `**`crypto connect vlan`** *`vlan_ID`*	Connects the outside access port to the inside (crypto) interface VLAN and enters crypto-connect mode. • *vlan_ID*—Interface VLAN identifier.

For examples of IPsec stateless failover configurations using HSRP, see the "IPsec Stateless Failover Using HSRP with Crypto-Connect Mode Configuration Examples" section on page 11-15.

Configuring IPsec Stateless Failover with VRF Mode

Chassis-to-chassis failover with VRF mode is configured differently from non-VRF (crypto-connect) mode. In VRF mode, the HSRP configuration goes on the physical interface, but the crypto map is added to the interface VLAN. In non-VRF mode, both the HSRP configuration and the crypto map are on the same interface. RRI dynamically inserts and removes routes from the active and standby switch VRF routing tables.

For a configuration example of VRF mode with stateless failover, see the "IPsec Stateless Failover Using HSRP with VRF Mode Configuration Example" section on page 11-17.

Configuring Intrachassis IPsec Stateful Failover Using a Blade Failure Group

This section describes how to configure IPsec stateful failover within a chassis using a blade failure group (BFG).

Note IPsec stateful failover is only supported within a chassis using a blade failure group. Inter-chassis failover is not supported.

When one or more pairs of VSPAs are installed in a chassis, each pair can be configured as a blade failure group (BFG). The two modules do not need to reside within the same SSC. Within the BFG, each VSPA serves as a backup for the other VSPA. A BFG is an active/active configuration.

When a VSPA is joining a BFG or booting to come online, all of its IPsec and IKE data structures are synchronized with its peer. For each IPsec tunnel or IKE SA, and based on the per-interface crypto engine assignment, only one VSPA can be designated as active. For IKE SAs, an active VSPA is the one that is accelerating cryptographic computations. For IPsec tunnels, the active VSPA is the one that the traffic is passing through. For each IKE SA or IPsec tunnel, there is an active VSPA and its backup. For example, in a system that supports 1000 tunnels with two VSPAs, 500 of the tunnels may be active on one VSPA and the remaining 500 may be active on the second VSPA. Both VSPAs then replicate data to each other so that either one can take over in the event of a failure. Each VSPA can have only one partner for all of the IKE and IPsec SAs that it protects.

IPsec Stateful Failover Using a BFG Configuration Guidelines

When configuring IPsec stateful failover using a BFG, follow these guidelines:

- You can install or remove one of the VSPAs comprising a BFG without disrupting any of the tunnels on the other VSPA.

Configuring a BFG for IPsec Stateful Failover

To configure IPsec stateful failover using a BFG, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# redundancy`	Enters redundancy configuration mode.
Step 2	`Router(config-red)# linecard-group group-number feature-card`	Identifies the line card group ID for a Blade Failure Group and enters redundancy line card configuration mode. • *group-number*—Specifies a group ID for the BFG.
Step 3	`Router(config-r-lc)# subslot slot/subslot`	Adds the first module to the group. • *slot*—Specifies the chassis slot number where the carrier card is installed. • *subslot*—Specifies the secondary slot number on a carrier card where a module is installed.
Step 4	`Router(config-r-lc)# subslot slot/subslot`	Adds the second module to the group.

For an IPsec stateful failover using a BFG configuration example, see the "IPsec Stateful Failover Using a Blade Failure Group Configuration Example" section on page 11-19.

Verifying the IPsec Stateful Failover Using a BFG Configuration

To verify the IPsec stateful failover using a BFG configuration, use the **show redundancy linecard group** and **show crypto ace redundancy** commands.

To display the components of a Blade Failure Group, enter the **show redundancy linecard group** command:

```
Router# show redundancy linecard-group 1

Line Card Redundancy Group:1 Mode:feature-card
Class:load-sharing
Cards:
Slot:3 Sublot:0
Slot:5 Sublot:0
```

To display information about a Blade Failure Group, enter the **show crypto ace redundancy** command:

```
Router# show crypto ace redundancy

--------------------------------------
LC Redundancy Group ID             :1
Pending Configuration Transactions:0
Current State                      :OPERATIONAL
Number of blades in the group      :2
Slots
--------------------------------------
Slot:3 subslot:0
Slot state:0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 22 times
Initialization Timer not running
Slot:5 subslot:0
Slot state:0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 24 times
Initialization Timer not running
```

Configuration Examples

This section provides examples of the following configurations:

- Multiple VSPAs in a Chassis Configuration Example, page 11-12
- IPsec Stateless Failover Using HSRP with Crypto-Connect Mode Configuration Examples, page 11-15
- IPsec Stateless Failover Using HSRP with VRF Mode Configuration Example, page 11-17
- IPsec Stateful Failover Using a Blade Failure Group Configuration Example, page 11-19

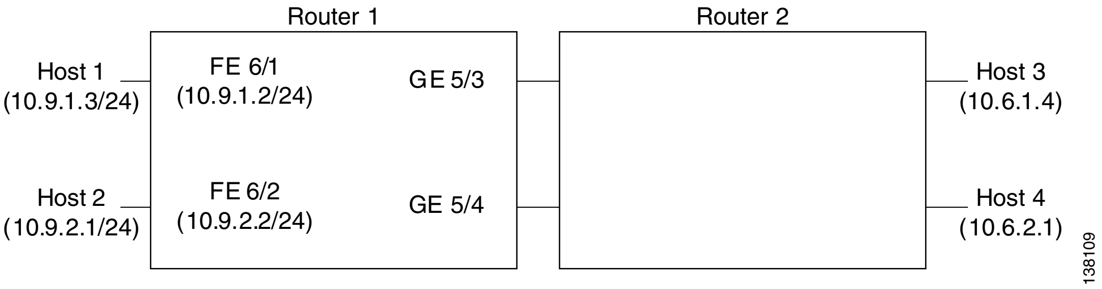
Multiple VSPAs in a Chassis Configuration Example

This section provides an example of a configuration using multiple VSPAs in a chassis as shown in Figure 11-1. Note the following in these examples:

- A VSPA is in slot 2, subslot 0 and slot 3, subslot 0 of router 1.
- In the configuration example, three exclamation points (!!!) precede descriptive comments.

Note In the following figure, the router with the VSPA could be a Cisco 7600 Series router or a Catalyst 6500 Series switch.

Figure 11-1 Multiple VSPAs in a Chassis Configuration Example



```
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key mykey address 10.8.1.1
crypto isakmp key mykey address 10.13.1.1
!
crypto ipsec transform-set xform1 ah-md5-hmac esp-des esp-sha-hmac
crypto ipsec transform-set xform2 esp-3des esp-sha-hmac
!
!!! crypto map applied to VLAN 12, which is
!!! assigned to "inside" port of VSPA in slot 3
crypto map cmap2 10 ipsec-isakmp
 set peer 10.8.1.1
 set transform-set xform1
 match address 102
!
!!! crypto map applied to VLAN 20, which is
!!! assigned to "inside" port of VSPA in slot 2/0
crypto map cmap3 10 ipsec-isakmp
 set peer 10.13.1.1
 set transform-set xform2
 match address 103
!
!!! "port" VLAN, crypto connected to VLAN 12 by VSPA on slot 3/0
interface Vlan11
 no ip address
 crypto connect vlan 12
!
!!! "interface" VLAN, assigned to VSPA on slot 3/0
interface Vlan12
 ip address 10.8.1.2 255.255.0.0
 crypto map cmap2
 crypto engine slot 3/0
!
!!! "port" VLAN, crypto connected to VLAN 20 by VSPA on slot 2/0
interface Vlan19
 no ip address
 crypto connect vlan 20
!
!!! "interface" VLAN, assigned to VSPA on slot 2/0
interface Vlan20
```

```
 ip address 10.13.1.2 255.255.0.0
 crypto map cmap3
 crypto engine slot 2/0
!
!!! connected to Host 1
interface FastEthernet6/1
 ip address 10.9.1.2 255.255.255.0
!
!!! connected to Host 2
interface FastEthernet6/2
 ip address 10.9.2.2 255.255.255.0
!
!!! connected to Router 2
interface GigabitEthernet5/3
 switchport
 switchport mode access
 switchport access vlan 11
!
!!! connected to Router 2
interface GigabitEthernet5/4
 switchport
 switchport mode access
 switchport access vlan 19
!
interface GigabitEthernet2/0/1
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 12,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet2/0/2
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 11,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet3/0/1
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 20,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet3/0/2
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 19,1002-1005
 switchport mode trunk
 cdp enable
!
ip classless
!
!!! packets from Host 1 to Host 3 are routed from FastEthernet6/1
!!! to VLAN 12, encrypted with crypto map cmap2
!!! using VSPA in slot 3/0, and forwarded to peer 10.8.1.1
!!! through GigabitEthernet5/3
ip route 10.6.1.4 255.255.255.255 10.8.1.1
!
!!! packets from Host 2 to Host 4 are routed from FastEthernet6/2
!!! to VLAN 20, encrypted with crypto map cmap3
!!! using VSPA in slot 2/0, and forwarded to peer 10.13.1.1
!!! through GigabitEthernet5/4
ip route 10.6.2.1 255.255.255.255 10.13.1.1
!
!!! ACL matching traffic between Host 1 and Host 3
access-list 102 permit ip host 10.9.1.3 host 10.6.1.4
!
!!! ACL matching traffic between Host 2 and Host 4
access-list 103 permit ip host 10.9.2.1 host 10.6.2.1
```

IPsec Stateless Failover Using HSRP with Crypto-Connect Mode Configuration Examples

This section provides the following configuration examples of IPsec stateless failover using HSRP:

- IPsec Stateless Failover for the Active Chassis Configuration Example, page 11-15
- IPsec Stateless Failover for the Remote Switch Configuration Example, page 11-16

IPsec Stateless Failover for the Active Chassis Configuration Example

The following example shows the configuration for an active chassis that is configured for IPsec stateless failover using HSRP:

```
hostname router-1
!
vlan 2-1001
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 1234567890 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set PYTHON esp-3des
!
crypto dynamic-map dynamap_1 20
 set transform-set PYTHON
 reverse-route
!
!
crypto map MONTY 1 ipsec-isakmp dynamic dynamap_1
!
interface GigabitEthernet1/3
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet1/4
 ip address 50.0.0.3 255.0.0.0
!
interface GigabitEthernet4/0/1
```

```
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 502
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 172.1.1.3 255.255.255.0
 standby ip 172.1.1.100
 standby preempt
 standby name KNIGHTSOFNI
 standby track GigabitEthernet1/3
 standby track GigabitEthernet1/4
 no mop enabled
 crypto map MONTY redundancy KNIGHTSOFNI
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
ip route 10.0.0.0 255.0.0.0 172.1.1.4
ip route 20.0.0.0 255.0.0.0 172.1.1.4
ip route 50.0.0.0 255.0.0.0 50.0.0.13
ip route 50.0.1.1 255.255.255.255 50.0.0.13
ip route 50.0.2.1 255.255.255.255 50.0.0.13
ip route 50.0.3.1 255.255.255.255 50.0.0.13
ip route 50.0.4.1 255.255.255.255 50.0.0.13
ip route 50.0.5.1 255.255.255.255 50.0.0.13
```

IPsec Stateless Failover for the Remote Switch Configuration Example

The following example shows the configuration for a remote switch that is configured for IPsec stateless failover using HSRP:

```
hostname router-remote
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 12345 address 172.1.1.100
!
!
crypto ipsec transform-set ha_transform esp-3des
!
crypto map test_1 local-address Vlan2
crypto map test_1 10 ipsec-isakmp
 set peer 172.1.1.100
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 set pfs group2
 match address test_1
!
interface GigabitEthernet1/1
 ip address 10.0.0.2 255.255.255.0
!
interface GigabitEthernet1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1-2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 20.0.1.1 255.255.255.0
 crypto map test_1
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
ip route 10.0.0.0 255.0.0.0 10.0.0.13
ip route 50.0.1.0 255.255.255.0 20.0.1.2
ip route 172.1.1.0 255.255.255.0 20.0.1.2
!
ip access-list extended test_1
 permit ip host 10.0.1.1 host 50.0.1.1
```

IPsec Stateless Failover Using HSRP with VRF Mode Configuration Example

The following example shows a VRF mode configuration with HSRP chassis-to-chassis stateless failover with crypto maps:

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
vlan 2,3
!
crypto keyring key1
  pre-shared-key address 14.0.1.1 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp keepalive 10
crypto isakmp profile ivrf
   vrf ivrf
   keyring key1
   match identity address 14.0.1.1 255.255.255.255
!
crypto ipsec transform-set ts esp-3des esp-sha-hmac
!
crypto map map_vrf_1 local-address Vlan3
crypto map map_vrf_1 10 ipsec-isakmp
 set peer 14.0.1.1
 set transform-set ts
 set isakmp-profile ivrf
 match address acl_1
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 13.254.254.1 255.255.255.0
!
interface GigabitEthernet1/1.1
 encapsulation dot1Q 2000
 ip vrf forwarding ivrf
 ip address 13.254.254.1 255.0.0.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 3
 switchport mode access
!

interface GigabitEthernet4/0/1
 !VSPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !VSPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan3
 ip address 15.0.0.2 255.255.255.0
 standby delay minimum 0 reload 0
 standby 1 ip 15.0.0.100
 standby 1 timers msec 100 1
 standby 1 priority 105
 standby 1 preempt
 standby 1 name std-hsrp
 standby 1 track GigabitEthernet1/2
 crypto engine outside
!
interface Vlan2
 ip vrf forwarding ivrf
 ip address 15.0.0.252 255.255.255.0
 crypto map map_vrf_1 redundancy std-hsrp
 crypto engine slot 4/0 inside

!
ip classless
ip route 12.0.0.0 255.0.0.0 15.0.0.1
ip route 13.0.0.0 255.0.0.0 13.254.254.2
ip route 14.0.0.0 255.0.0.0 15.0.0.1
ip route 223.255.254.0 255.255.255.0 17.1.0.1
ip route vrf ivrf 12.0.0.1 255.255.255.255 15.0.0.1
!
ip access-list extended acl_1
 permit ip host 13.0.0.1 host 12.0.0.1
!
!
arp vrf ivrf 13.0.0.1 0000.0000.2222 ARPA
```

IPsec Stateful Failover Using a Blade Failure Group Configuration Example

The following example shows how to configure IPsec stateful failover using a blade failure group (BFG):

```
Router(config)# redundancy
Router(config-red)# line-card-group 1 feature-card
Router(config-r-lc)# subslot 3/1
Router(config-r-lc)# subslot 5/1
```

CHAPTER 12

Configuring Monitoring and Accounting

This chapter provides information about configuring monitoring and accounting using the VSPA on the Catalyst 6500 Series switch. It includes the following sections:

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, see the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

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

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

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

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

Overview of Monitoring and Accounting for the VSPA

This chapter describes some IPsec features that can be used to monitor and manage the IPsec VPN. These features include:

- The IPsec VPN monitoring feature, which provides VPN session monitoring enhancements that will allow you to troubleshoot the VPN and monitor the end-user interface.
- The IPsec VPN accounting feature, which enables session accounting records to be generated by indicating when the session starts and when it stops.

- The IPsec and IKE MIB support for Cisco VRF-aware IPsec feature, which provides manageability of VPN routing and forwarding- (VRF-) aware IPsec using MIBs.

Monitoring and Managing IPsec VPN Sessions

The IPsec VPN monitoring feature provides VPN session monitoring enhancements that will allow you to troubleshoot the Virtual Private Network (VPN) and monitor the end-user interface. A crypto session is a set of IPsec connections (flows) between two crypto endpoints. If the two crypto endpoints use IKE as the keying protocol, they are IKE peers to each other. Typically, a crypto session consists of one IKE security association (for control traffic) and at least two IPsec security associations (for data traffic, one per each direction). There may be duplicated IKE security associations (SAs) and IPsec SAs or duplicated IKE SAs or IPsec SAs for the same session in the duration of rekeying or because of simultaneous setup requests from both sides.

Session monitoring enhancements include the following:

- Ability to specify an Internet Key Exchange (IKE) peer description in the configuration file
- Summary listing of crypto session status
- Syslog notification for crypto session up or down status
- Ability to clear both IKE and IP Security (IPsec) security associations (SAs) using one command-line interface (CLI)

Adding the Description of an IKE Peer

To add the description of an IKE peer to an IPsec VPN session, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto isakmp peer`** `{`**`ip-address`** *`ip-address`*`}`	Enables an IPsec peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and enters ISAKMP peer configuration mode. • *ip-address*—IP address of the peer.
Step 2	`Router(config-isakmp-peer)#` **`description`** *`description`*	Adds a description for an IKE peer. • *description*—Description identifying the peer.

Verifying Peer Descriptions

To verify peer descriptions, enter the **show crypto isakmp peer** command:

```
Router# show crypto isakmp peer

Peer: 10.2.2.9 Port: 500
Description: connection from site A
flags: PEER_POLICY
```

When the peer at address 10.2.2.9 connects and the session comes up, the syslog status will be shown as follows:

```
%CRYPTO-5-SESSION_STATUS: Crypto tunnel is UP. Peer 10.2.2.9:500 Description: connection
from site A Id: ezvpn
```

Getting a Summary Listing of Crypto Session Status

You can get a list of all the active VPN sessions by entering the **show crypto session** command. The listing will include the following:

- Interface
- IKE peer description, if available
- IKE SAs that are associated with the peer by which the IPsec SAs are created
- IPsec SAs serving the flows of a session

Multiple IKE or IPsec SAs may be established for the same peer, in which case IKE peer descriptions will be repeated with different values for the IKE SAs that are associated with the peer and for the IPsec SAs that are serving the flows of the session.

You can also use the **show crypto session detail** variant of this command to obtain more detailed information about the sessions.

The following is sample output for the **show crypto session** command without the **detail** keyword:

```
Router# show crypto session

Crypto session current status

Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.0.0.2/500
IKE SA: local 172.0.0.1/500 remote 172.0.0.2/500 Active
IPSEC FLOW: permit ip 10.10.10.0/255.255.255.0 10.30.30.0/255.255.255.0
Active SAs: 2, origin: crypto map
```

The following is sample output using the **show crypto session** command with the **detail** keyword:

```
Router# show crypto session detail

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.1.1.3 port 500 fvrf: (none) ivrf: (none)
Desc: this is my peer at 10.1.1.3:500 Green
Phase1_id: 10.1.1.3
IKE SA: local 10.1.1.4/500 remote 10.1.1.3/500 Active
Capabilities:(none) connid:3 lifetime:22:03:24
IPSEC FLOW: permit 47 host 10.1.1.4 host 10.1.1.3
Active SAs: 0, origin: crypto map
Inbound: #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
IPSEC FLOW: permit ip host 10.1.1.4 host 10.1.1.3
Active SAs: 4, origin: crypto map
Inbound: #pkts dec'ed 4 drop 0 life (KB/Sec) 4605665/2949
Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4605665/2949
```

Syslog Notification for Crypto Session Up or Down Status

The syslog notification for crypto session up or down status function provides syslog notification every time the crypto session comes up or goes down. To enable syslog logging of the session status, enter the **crypto logging session** and **crypto logging ezvpn** commands in configuration mode.

The following is a sample syslog notification showing that a crypto session is up:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is UP. Peer 10.6.6.1:500 fvrf=name10 ivrf=name20
Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

The following is a sample syslog notification showing that a crypto session is down:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is DOWN. Peer 10.6.6.1:500 fvrf=name10
ivrf=name20 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

Clearing a Crypto Session

In previous Cisco IOS software releases, there was no single command to clear both IKE and IPsec security associations (SAs). Instead, you entered the **clear crypto isakmp** command to clear IKE and the **clear crypto ipsec** command to clear IPsec. The **clear crypto session** command allows you to clear both IKE and IPsec with a single command. To clear a specific crypto session or a subset of all the sessions (for example, a single tunnel to one remote site), you must provide session-specific parameters, such as a local or remote IP address, a local or remote port, a front-door VPN routing and forwarding (FVRF) name, or an inside VRF (IVRF) name. Typically, the remote IP address will be used to specify a single tunnel to be deleted.

If a local IP address is provided as a parameter when you enter the **clear crypto session** command, all the sessions (and their IKE SAs and IPsec SAs) that share the IP address as a local crypto endpoint (IKE local address) will be cleared. If you do not provide a parameter when you enter the **clear crypto session** command, all IPsec SAs and IKE SAs in the switch will be deleted.

To clear a crypto session, enter the **clear crypto session** command in privileged EXEC mode from the switch command line. No configuration statements are required in the configuration file to use this command:

```
Router# clear crypto session
```

For complete configuration information for IPsec VPN Monitoring, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t4/feature/guide/gt_ipsvm.html

For IPsec VPN monitoring configuration examples, see the "IPsec VPN Monitoring Configuration Example" section on page 12-10.

Configuring IPsec VPN Accounting

The IPsec VPN accounting feature enables session accounting records to be generated by indicating when the session starts and when it stops.

A VPN session is defined as an Internet Key Exchange (IKE) security association (SA) and the one or more SA pairs that are created by the IKE SA. The session starts when the first IP Security (IPsec) pair is created and stops when all IPsec SAs are deleted. If IPsec accounting is configured, after IKE phases are complete, an accounting start record is generated for the session. New accounting records are not generated during a rekeying.

Session-identifying information and session-usage information is passed to the Remote Authentication Dial-In User Service (RADIUS) server by standard RADIUS attributes and vendor-specific attributes (VSAs).

To enable IPsec VPN accounting, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **aaa new-model**	Enables periodic interim accounting records to be sent to the accounting server.
Step 2	Router(config)# **aaa authentication login** *list-name* **group radius**	Sets authentication, authorization, and accounting (AAA) authentication at login using RADIUS servers. • *list-name*—Character string used to name the list of authentication methods activated when a user logs in. • **group radius**—Uses the list of all RADIUS servers for authentication.
Step 3	Router(config)# **aaa authorization network** *list-name* **group radius**	Runs authorization for all network-related service requests, including Serial Line Internet Protocol (SLIP), PPP, PPP Network Control Programs (NCPs), and AppleTalk Remote Access (ARA). • *list-name*—Character string used to name the list of authorization methods activated when a user logs in. • **group radius**—Uses the list of all RADIUS servers for authentication.
Step 4	Router(config)# **aaa accounting network** *list-name* **start-stop** [**broadcast**] **group radius**	Enables AAA accounting of network-related requested services for billing or security purposes when you use RADIUS. • *list-name*—Character string used to name the list of the accounting methods. • **start-stop**—Sends a start accounting notice at the beginning of a process and a stop accounting notice at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether the start accounting notice was received by the accounting server. • **broadcast**—(Optional) Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group. • **group radius**—Uses the list of all RADIUS servers for authentication as defined by the **aaa group server radius** command.

	Command	Purpose
Step 5	`Router(config)# aaa accounting update periodic minutes`	(Optional) Sends accounting updates to the accounting server while a session is up. • *minutes* — Specifies the interval (in number of minutes) at which accounting records are to be sent to the accounting server.
Step 6	`Router(config)# aaa session-id common`	Specifies whether the same session ID will be used for each AAA accounting service type within a call or whether a different session ID will be assigned to each accounting service type. • **common**—Ensures that all session identification (ID) information that is sent out for a given call will be made identical. The default behavior is common.
Step 7	`Router(config)# crypto isakmp profile profile-name`	Audits IP security (IPsec) user sessions and enters isakmp-profile configuration mode. • *profile-name*—Name of the user profile. To associate a user profile with the RADIUS server, the user profile name must be identified.
Step 8	`Router(conf-isa-prof)# vrf ivrf`	Associates the on-demand address pool with a Virtual Private Network (VPN) routing and forwarding (VRF) instance name. • *ivrf*—VRF to which the IPsec tunnel will be mapped.
Step 9	`Router(conf-isa-prof)# match identity group group-name`	Matches an identity from a peer in an ISAKMP profile. • *group-name*—A unity group that matches identification (ID) type ID_KEY_ID. If unity and main mode Rivest, Shamir, and Adelman (RSA) signatures are used, the *group-name* argument matches the Organizational Unit (OU) field of the Distinguished Name (DN).
Step 10	`Router(conf-isa-prof)# client authentication list list-name`	Configures Internet Key Exchange (IKE) extended authentication (XAUTH) in an Internet Security Association and Key Management Protocol (ISAKMP) profile. • *list-name*—Character string used to name the list of authentication methods activated when a user logs in. The list name must match the list name that was defined during the authentication, authorization, and accounting (AAA) configuration.

	Command	Purpose
Step 11	`Router(conf-isa-prof)#` **`isakmp authorization list`** *`list-name`*	Configures an IKE shared secret and other parameters using the AAA server in an ISAKMP profile. The shared secret and other parameters are generally pushed to the remote peer via mode configuration (MODECFG). • *list-name*—AAA authorization list used for configuration mode attributes or preshared keys for aggressive mode.
Step 12	`Router(conf-isa-prof)#` **`client configuration address`** `[`**`initiate`** `|` **`respond`**`]`	Configures IKE mode configuration (MODECFG) in the ISAKMP profile. • **initiate**—(Optional) Switch will attempt to set IP addresses for each peer. • **respond**—(Optional) Switch will accept requests for IP addresses from any requesting peer.
Step 13	`Router(conf-isa-prof)#` **`accounting`** *`list-name`*	Enables AAA accounting services for all peers that connect via this ISAKMP profile. • *list-name*— Name of a client accounting list.
Step 14	`Router(conf-isa-prof)#` **`exit`**	Exits isakmp profile configuration mode and returns to global configuration mode.
Step 15	`Router(config)#` **`crypto dynamic-map`** *`dynamic-map-name dynamic-seq-num`*	Creates a dynamic crypto map template and enters the crypto map configuration command mode. • *dynamic-map-name*—Name of the dynamic crypto map set that should be used as the policy template. • *dynamic-seq-num*—Sequence number you assign to the dynamic crypto map entry.
Step 16	`Router(config-crypto-map)#` **`set transform-set`** *`transform-set-name`*	Specifies which transform sets can be used with the crypto map template. A transform set defines IPsec security protocols and algorithms. Transform sets and their accepted values are described in the *Cisco IOS Security Command Reference*. • *transform-set-name*—Name of the transform set.
Step 17	`Router(config-crypto-map)#` **`set isakmp-profile`** *`profile-name`*	Sets the ISAKMP profile name. • *profile-name*—Name of the ISAKMP profile.
Step 18	`Router(config-crypto-map)#` **`reverse-route`** `[`**`remote-peer`**`]`	Allows routes (IP addresses) to be injected for destinations behind the VPN remote tunnel endpoint and may include a route to the tunnel endpoint itself (using the **remote-peer** keyword for the crypto map). • **remote-peer**—(Optional) Routes of public IP addresses and IP security (IPsec) tunnel destination addresses are inserted into the routing table.

	Command	Purpose
Step 19	Router(config-crypto-map)# **exit**	Exits crypto map configuration mode and returns to global configuration mode.
Step 20	Router(config)# **crypto map** *map-name* **ipsec-isakmp dynamic** *dynamic-map-name*	Creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-nam*e—Name that identifies the crypto map set. • *dynamic-map-name*—Name of the dynamic crypto map set that should be used as the policy template.
Step 21	Router(config)# **radius-server host** *ip-address* [**auth-port** *auth-port-number*] [**acct-port** *acct-port-number*]	Specifies a RADIUS server host. • *ip-address* —IP address of the RADIUS server host. • *auth-port-number*—(Optional) UDP destination port number for authentication requests; the host is not used for authentication if set to 0. If unspecified, the port number defaults to 1645. • *acct-port-number*—(Optional) UDP destination port number for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.
Step 22	Router(config)# **radius-server key** *string*	Sets the authentication and encryption key for all RADIUS communications between the switch and the RADIUS daemon. • *string*—The unencrypted (cleartext) shared key.
Step 23	Router(config)# **interface** *type slot*/[*subslot*]/*port*	Configures an interface type and enters interface configuration mode. • *slot*/[*subslot*]/ *port*—Number of the slot, subslot (optional), and port to be configured.
Step 24	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to an interface. • *map-nam*e—Name that identifies the crypto map set.

For complete configuration information for IPsec VPN Accounting, refer to this URL:

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

For IPsec VPN accounting configuration examples, see the "IPsec VPN Accounting Configuration Example" section on page 12-9.

Configuration Examples

This section provide examples of the following configurations:

- IPsec VPN Accounting Configuration Example, page 12-9

- IPsec VPN Monitoring Configuration Example, page 12-10

IPsec VPN Accounting Configuration Example

The following example shows how to enable the IPsec VPN accounting feature:

```
aaa new-model
!
!
aaa group server radius r1
 server-private 10.30.1.52 auth-port 1812 acct-port 1813 key allegro
!
aaa authentication login test_list group r1
aaa authorization network test_list group r1
aaa accounting update periodic 10 jitter maximum 0
aaa accounting network test_list start-stop group r1
!
ip vrf ivrf1
 rd 1:2
!
crypto engine mode vrf
!
crypto isakmp policy 5
 encr 3des
 authentication pre-share
 group 2
 lifetime 14400
!
crypto isakmp client configuration group test
 key world
 pool pool1
!
crypto isakmp profile test_pro
   vrf ivrf1
   match identity group test
   client authentication list test_list
   isakmp authorization list test_list
   client configuration address respond
   accounting test_list
!
crypto ipsec transform-set t3 esp-3des esp-sha-hmac
!
!
crypto dynamic-map dyn-ra 10
 set transform-set t3
 set isakmp-profile test_pro
 reverse-route
!
!
crypto map map-ra local-address GigabitEthernet3/15
crypto map map-ra 1 ipsec-isakmp dynamic dyn-ra
!
!
interface GigabitEthernet1/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,100,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
!

interface GigabitEthernet3/15
 mtu 9216
 ip address 120.0.0.254 255.255.255.0
crypto engine outside
!
!
!
interface Vlan100
 ip vrf forwarding ivrf1
 ip address 120.0.0.100 255.255.255.0
 ip flow ingress
 crypto map map-ra
crypto engine slot 1/0 inside
!
!
!
ip local pool pool1 100.0.1.1 100.0.5.250
```

IPsec VPN Monitoring Configuration Example

The following example shows how to configure an IKE peer for IPsec VPN monitoring:

```
!
upgrade fpd auto
version 12.2
service timestamps debug datetime
service timestamps log datetime
no service password-encryption
service counters max age 5
!
hostname Ez-DCM-CC
!
boot-start-marker
boot system disk1:s72033-adventerprisek9_wan-mz.122-33.SXI
boot-end-marker
!
logging buffered 1000000 debugging
enable secret 5 $1$i5FZ$47ybx5dEaUKc3eRaDIZ/z.
!
aaa new-model
aaa authentication login myuserlist local
aaa authorization network myuserlist local
!
aaa session-id common
clock timezone PST -7
ip subnet-zero
```

```
!
no ip domain-lookup
ip domain-name cisco.com
ipv6 mfib hardware-switching replication-mode ingress
vtp mode transparent
no mls acl tcam share-global
mls netflow interface
no mls flow ip
no mls flow ipv6
mls cef error action freeze
!
redundancy
 keepalive-enable
 mode sso
 linecard-group 0 feature-card
  class load-sharing
  subslot 4/0
 main-cpu
  auto-sync running-config
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
diagnostic monitor syslog
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
!
power redundancy-mode combined
port-channel per-module load-balance
!
vlan internal allocation policy descending
vlan access-log ratelimit 2000
!
vlan 2-3,16-17
!
crypto logging session
crypto logging ezvpn
!
crypto logging ezvpn group mygroup
!
crypto isakmp policy 10
 encr aes
 authentication pre-share
 group 2
 lifetime 43200
crypto isakmp key WorldCup2006 address 0.0.0.0 0.0.0.0
!
crypto isakmp client configuration group mygroup
 key mykey
 pool mypool
!
crypto isakmp peer address 16.0.0.3
 description first-ezvpn-client
!
crypto isakmp peer address 16.0.0.4
 description second-ezvpn-client
!
crypto ipsec security-association lifetime seconds 21600
!
crypto ipsec transform-set MyTranSet esp-aes esp-sha-hmac
no crypto ipsec nat-transparency udp-encaps
!
crypto call admission limit ike in-negotiation-sa 10
!
crypto dynamic-map DynMap1 10
```

```
 set transform-set MyTranSet
 reverse-route
!
crypto map MyMap1 client authentication list myuserlist
crypto map MyMap1 isakmp authorization list myuserlist
crypto map MyMap1 client configuration address respond
crypto map MyMap1 500 ipsec-isakmp dynamic DynMap1
!
interface GigabitEthernet1/25
 no ip address
 crypto connect vlan 16
!
interface GigabitEthernet1/27
 no ip address
 crypto connect vlan 17
!
interface GigabitEthernet1/29
 ip address 26.0.0.2 255.255.255.0
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 16,17,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos vlan-based
 mls qos trust cos
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust cos
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet5/2
 ip address 44.0.111.114 255.0.0.0
 media-type rj45
!
interface Vlan1
 no ip address
 ip flow ingress
 ip igmp snooping querier
 shutdown
!
interface Vlan16
 ip address 16.0.0.2 255.255.224.0
 no mop enabled
 crypto map MyMap1
 crypto engine slot 4/0
!
interface Vlan17
 ip address 16.0.32.2 255.255.224.0
 no mop enabled
 crypto map MyMap1
 crypto engine slot 4/0
!
```

```
ip local pool mypool 36.0.0.1 36.0.15.254
ip local pool mypool 36.0.16.1 36.0.31.254
ip local pool mypool 36.0.32.1 36.0.47.254
ip local pool mypool 36.0.48.1 36.0.63.254
ip default-gateway 44.0.100.1
ip classless
ip route 43.0.0.0 255.0.0.0 44.0.100.1
ip route 45.0.0.0 255.0.0.0 44.0.100.1
ip route 223.255.254.53 255.255.255.255 44.0.100.1
ip route 223.255.254.54 255.255.255.255 44.0.100.1
!
no ip http server
no ip http secure-server
!
radius-server source-ports 1645-1646
!
control-plane
!
dial-peer cor custom
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password cisco
 transport input lat pad mop udptn telnet rlogin ssh nasi acercon
line vty 5 15
 transport input lat pad mop udptn telnet rlogin ssh nasi acercon
!
monitor event-trace platform cmfi lc agg-label
monitor event-trace platform cmfi lc error
ntp clock-period 17280219
ntp update-calendar
ntp server 223.255.254.254
ntp server 223.255.254.53
mac-address-table aging-time 0
!
end
```

CHAPTER 13

Troubleshooting

This chapter describes techniques that you can use to troubleshoot the operation of the VSPA in a Catalyst 6500 Series switch.

It includes the following sections:

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

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

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

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

General Troubleshooting Information

This section describes general information for troubleshooting the VSPA and the SSC-600. It includes the following sections:

Interpreting Console Error Messages

The Catalyst 6500 Series switch can generate error messages and other system messages to inform the operator of events that might require attention. These messages can be displayed on the console, or sent to a logging host using the System Logging (Syslog) protocol or Simple Network Management Protocol (SNMP).

System error messages are organized in the documentation according to the particular system facility that produces the messages. The VSPA and SSC-600 use the following facility names in error messages:

- VSPA—WS_IPSEC_3 (also VSPA, VPNSPA)
- SSC-600—CAT6000_SSC (also C7600_SSC600)

To view the explanations and recommended actions for Catalyst 6500 Series switch error messages, including messages related to service modules, refer to the following documents:

- *Cisco IOS Release 12.2SX System Message Guide* at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2sx/system/messages/122sxsms.html
- *System Messages for 12.2S* (for error messages in Release 12.2S) at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2s/system/messages/122sdebu.html

Using debug Commands

For information about **debug** commands specific to the Cisco IOS software release 12.2SX, see the *Cisco IOS Master Command List, Release 12.2SX* at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/122sxmcl/12_2sx_mcl_book.html

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support personnel. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

For information about available crypto conditional debugging commands, see the "Using Crypto Conditional Debug" section on page 13-22.

For more information about other **debug** commands that can be used on a Catalyst 6500 Series switch, see the *Cisco IOS Debug Command Reference, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/debug/command/reference/122debug.html

Using show Commands

There are several **show** commands that you can use to monitor and troubleshoot modules on the Catalyst 6500 Series switch. This chapter describes using the **show hw-module slot** command to perform troubleshooting of your VSPA.

For more information about using **show** commands to verify and monitor your VSPA, see the "Displaying the Module Hardware Type" section on page 1-10.

For more information about security-related **show** commands, see the *Cisco IOS Security Command Reference* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Using Online Diagnostics

Using the online diagnostics features provided by the Catalyst 6500 Series switch, you can test and verify the hardware functionality of the VSPA at bootup, as part of background health monitoring, or on demand.

Note In Cisco IOS Release 12.2(33)SXI, a bootup diagnostics failure in the VSPA results in a restart of the VSPA and the SSC-600. Health monitoring failures exceeding a certain threshold result in a syslog message and a restart of the VSPA.

In Cisco IOS Release 12.2(33)SXI1 and later releases, a bootup diagnostics failure in the VSPA results in a restart of only the VSPA. Health monitoring failures exceeding a certain threshold result in a syslog message.

For more information about online diagnostics, see the *Catalyst 6500 Release 12.2SXH and Later Software Configuration Guide* at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/diags.html

Monitoring the VSPA

This section describes commands that can be used to display information about the VSPA hardware and configuration. It consists of the following subsections:

- Displaying VSPA Hardware and System Information, page 13-3
- Displaying VSPA Configuration Information, page 13-6

Displaying VSPA Hardware and System Information

To display hardware and system information, use the following commands:

- **show diagbus, show module, show crypto eli**—See the "Displaying Information About VSPA Ports" section on page 13-3.
- **show crypto engine accelerator statistic slot**—See the "Displaying Platform and Network Interface Controller Statistics for the VSPA" section on page 13-4.
- **show hw-module slot fpd**—See the "Displaying Information About Hardware Revision Levels" section on page 13-6.

Displaying Information About VSPA Ports

To display information about the type of modules that are installed in the switch, use the **show diagbus** command.

The following example shows output from the **show diagbus** command on a Catalyst 6500 Series switch with a VSPA installed in subslot 1 of a SSC-600 that is installed in slot 5:

```
Router# show diagbus

Slot 5: Logical_index 10
        2-subslot Services SPA Carrier-600 controller
        Board is analyzed ipc ready
        HW rev 0.3, board revision A01
        Serial Number: abc Part number: 73-6348-01

        Slot database information:
        Flags: 0x2004   Insertion time: 0x3DB5F4BC (4d20h ago)

        Controller Memory Size:
                248 MBytes CPU Memory
                8 MBytes Packet Memory
                256 MBytes Total on Board SDRAM
        IOS (tm) cwlc Software (smsc-DWDBG-M), Experimental Version 12.2(20050623:231413)

        SPA Information:
        subslot 5/1: WS-IPSEC-3 (0x3D7), status: ok
```

For information about the **show module** and **show crypto eli** commands, see the "Displaying the Module Hardware Type" section on page 1-10.

Displaying Platform and Network Interface Controller Statistics for the VSPA

To display platform statistics, and, optionally, network interface controller statistics, use the **show crypto engine accelerator statistic slot** command.

The following example shows output from the **show crypto engine accelerator statistic** command on a Catalyst 6500 Series switch with a VSPA in subslot 0 of a SSC-600 that is installed in slot 1. The output displays platform statistics for the VSPA and also displays the network interface controller statistics.

```
Router# show crypto engine accelerator statistic slot 1/0 detail

VPN module in slot 1/0


Decryption Side Data Path Statistics
====================================
Packets RX...............: 454260
Packets TX...............: 452480

IPSec Transport Mode.....: 0
IPSec Tunnel Mode........: 452470
AH Packets...............: 0
ESP Packets..............: 452470
GRE Decapsulations.......: 0
NAT-T Decapsulations.....: 0
Clear....................: 8
ICMP.....................: 0

Packets Drop.............: 193
Authentication Errors....: 0
Decryption Errors........: 0
Replay Check Failed......: 0
Policy Check Failed......: 0
Illegal CLear Packet.....: 0
GRE Errors...............: 0
SPD Errors...............: 0
```

```
HA Standby Drop..........: 0

Hard Life Drop...........: 0
Invalid SA...............: 191
SPI No Match.............: 0
Destination No Match.....: 0
Protocol No Match........: 0

Reassembly Frag RX.......: 0
IPSec Fragments..........: 0
IPSec Reasm Done.........: 0
Clear Fragments..........: 0
Clear Reasm Done.........: 0
Datagrams Drop...........: 0
Fragments Drop...........: 0


Decryption Side Controller Statistics

=====================================
Frames RX................: 756088
Bytes RX.................: 63535848
Mcast/Bcast Frames RX....: 2341
RX Less 128Bytes.........: 756025
RX Less 512Bytes.........: 58
RX Less 1KBytes..........: 2
RX Less 9KBytes..........: 3
RX Frames Drop...........: 0

Frames TX................: 452365
Bytes TX.................: 38001544
Mcast/Bcast Frames TX....: 9
TX Less 128Bytes.........: 452343
TX Less 512Bytes.........: 22
TX Less 1KBytes..........: 0
TX Less 9KBytes..........: 0


Encryption Side Data Path Statistics
====================================
Packets RX...............: 756344
Packets TX...............: 753880
IPSec Transport Mode.....: 0
IPSec Tunnel Mode........: 753869
GRE Encapsulations.......: 0
NAT-T Encapsulations.....: 0
LAF prefragmented........: 0

Fragmented...............: 0
Clear....................: 753904
ICMP.....................: 0

Packets Drop.............: 123
IKE/TED Drop.............: 27
Authentication Errors....: 0
Encryption Errors........: 0
HA Standby Drop..........: 0

Hard Life Drop...........: 0
Invalid SA...............: 191

Reassembly Frag RX.......: 0
Clear Fragments..........: 0
Clear Reasm Done.........: 0
```

```
Datagrams Drop...........: 0
Fragments Drop...........: 0


Encryption Side Controller Statistics
=====================================
Frames RX................: 454065
Bytes RX.................: 6168274/
Mcast/Bcast Frames RX....: 1586
RX Less 128Bytes.........: 1562
RX Less 512Bytes.........: 452503
RX Less 1KBytes..........: 0
RX Less 9KBytes..........: 0
RX Frames Drop...........: 0

Frames TX................: 753558
Bytes TX.................: 100977246
Mcast/Bcast Frames TX....: 2
TX Less 128Bytes.........: 3
TX Less 512Bytes.........: 753555
TX Less 1KBytes..........: 0
TX Less 9KBytes..........: 0

Router#
```

Displaying Information About Hardware Revision Levels

To display information about the hardware revision of the SSC-600 and the VSPA as well as the version of the field-programmable devices (FPDs) that are on the carrier card and the module, use the **show hw-module slot fpd** command. Cisco technical engineers might need this information to debug or troubleshoot problems with a module installation.

The following example shows output from the **show hw-module slot** command on a Catalyst 6500 Series switch with a VSPA installed in subslot 0 of an SSC-600 that is installed in slot 6:

```
Router# show hw-module slot 2 fpd

==== ====================== ====== ==============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device: "ID-Name"    Version      Version
==== ====================== ====== ================== =========== ==============
   2 WS-SSC-600              0.302 1-SSC-600 I/O FPGA     1.0         1.0
                                   2-SSC-600 DPRX FPG     0.9         0.9
                                   3-SSC-600 DPTX FPG     0.13        0.13
                                   4-ROMMON               1.7         1.7
---- ---------------------- ------ ------------------ ----------- --------------
 2/0 WS-IPSEC-3              0.29  1-BOOTLOADER           1.1         1.1
                                   2-IO FPGA              1.0         1.0
                                   3-CAM FPGA             1.0         1.0
==== ====================== ====== ==============================================
```

Displaying VSPA Configuration Information

Except where noted, examples of command output in this section use the following example topology:

```
                              gi3/2              gi3/2
    loopback -------- Router1 ------------------------ Router2 -------- loopback
             12.0.0.0              10.10.40.0              13.0.0.0
              network               network                 network
```

To display information about the VSPA configuration, use the following commands:

- **show crypto vlan**—See the "Displaying Information About Access and Routed Ports That Are Connected" section on page 13-7, "Displaying the VPN Running State" section on page 13-8, and "Displaying Information About IP Multicast Over a GRE Tunnel" section on page 13-19.
- **show interfaces trunk**—See the "Displaying Information About the VLANs Allowed by a Trunk Port" section on page 13-8.
- **show crypto isakmp policy**—See the "Displaying Information About IKE Policies" section on page 13-8.
- **show crypto ipsec transform-set**—See the "Displaying Information About IPsec Transform Sets" section on page 13-9.
- **show crypto map**—See the "Displaying Information About Crypto Maps" section on page 13-9.
- **show crypto isakmp sa**—See the "Displaying Information About SAs at a Peer" section on page 13-10.
- **show crypto ipsec sa**—See the "Displaying Information About IPsec Security Associations" section on page 13-9.
- **show crypto key mypubkey rsa**—See the "Displaying Information About RSA Public Keys" section on page 13-11.
- **show crypto key pubkey-chain rsa**—See the "Displaying Information About RSA Public Keys" section on page 13-11.
- **show crypto pki certificates**—See the "Displaying Information About Certificates" section on page 13-12.
- **show crypto pki trustpoints**—See the "Displaying Information About Trustpoints" section on page 13-12.
- **show crypto session**—See the "Displaying Information About Crypto Sessions" section on page 13-13.
- **show interfaces tunnel**—See the "Displaying Tunnel Interface Information" section on page 13-13.
- **show redundancy linecard-grou**p—See the "Displaying Information About a BFG Configuration" section on page 13-18.
- **show crypto ace redundancy**—See the "Displaying Information About a BFG Configuration" section on page 13-18.
- **show ip nhrp**—See the "Displaying Information About the NHRP Cache" section on page 13-18.

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

Displaying Information About Access and Routed Ports That Are Connected

To verify that an access or routed port is connected, use the **show crypto vlan** command. The following is sample output from the command when a port is crypto-connected to a port VLAN:

```
Router1# show crypto vlan

Interface VLAN 2 on IPSec Service Module port GigabitEthernet2/0/1 connected to VLAN 502
with crypto map set mymap1
```

The following is sample output from the command when a port is crypto-connected to a physical interface:

```
Router1# show crypto vlan

Interface VLAN 2 on IPSec Service Module port GigabitEthernet2/0/1 connected to Gi3/2 with
crypto map set mymap2
```

Displaying Information About the VLANs Allowed by a Trunk Port

To display information about the VLANs allowed by a trunk port, use the **show interfaces trunk** command. The following is sample output from the command:

```
Router1# show interfaces trunk

Port                Mode         Encapsulation  Status        Native vlan
Gi2/0/1             on           802.1q         trunking      1
Gi2/0/2             on           802.1q         trunking      1

Port                Vlans allowed on trunk
Gi2/0/1             2
Gi2/0/2             502

Port                Vlans allowed and active in management domain
Gi2/0/1             2
Gi2/0/2             502

Port                Vlans in spanning tree forwarding state and not pruned
Gi2/0/1             2
Gi2/0/2             502
```

Displaying the VPN Running State

To display the VPN running state, use the **show crypto vlan** command.

In the following example, the interface VLAN belongs to the VSPA inside port:

```
Router1# show crypto vlan

Interface VLAN 2 on IPSec Service Module port GigabitEthernet2/0/1 connected to Gi3/2
```

In the following example, either the interface VLAN is missing on the VSPA inside port, the VSPA is removed from the chassis, or the VSPA was moved to a different subslot:

```
Router1# show crypto vlan

  Interface VLAN 2 connected to VLAN 3 (no IPSec Service Module attached)
```

Displaying Information About IKE Policies

To display information about IKE policies, use the **show crypto isakmp policy** command. The following is sample output from the command:

```
Router1# show crypto isakmp policy

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

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** command output:

```
WARNING:encryption hardware does not support the configured encryption method for ISAKMP
policy value
```

Displaying Information About IPsec Transform Sets

To display information about transform set configurations, use the **show crypto ipsec transform-set** command. The following is sample output from the command:

```
Router1# show crypto ipsec transform-set

Transform set tset: { esp-3des esp-sha-hmac  }
   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** command output:

```
WARNING:encryption hardware does not support transform.
```

Displaying Information About Crypto Maps

To display information about crypto map configurations, use the **show crypto map** command. The following is sample output from the command:

```
Router1# show crypto map

Crypto Map "mymap1" 10 ipsec-isakmp
        Peer = 10.10.40.2
        Extended IP access list 101
            access-list 101 permit ip host 12.0.0.1 host 13.0.0.1
        Current peer: 10.10.40.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                tset,
        }
        Interfaces using crypto map test:
                Vlan2
 using crypto engine WS-IPSEC-3[2/0]
```

Displaying Information About IPsec Security Associations

To display information about IPsec security associations, use the **show crypto ipsec sa** command.

Note When you first enter the **show crypto ipsec sa** command, the packet counters will not show the correct values. Subsequent instances of the command will display the correct values.

The following is sample output from the command:

```
Router1# show crypto ipsec sa

interface: Vlan2
    Crypto map tag: mymap1, local addr 10.10.40.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (12.0.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (13.0.0.1/255.255.255.255/0/0)
   current_peer 10.10.40.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.10.40.1, remote crypto endpt.: 10.10.40.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0xDAADD709(3668825865)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0x7D99ACA4(2107223204)
        transform: esp-3des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: :1, sibling flags 80000040,  crypto map: test
        sa timing: remaining key lifetime (k/sec): (4391325/3403)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xDAADD709(3668825865)
        transform: esp-3des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: :2, sibling flags 80000040,  crypto map: test
        sa timing: remaining key lifetime (k/sec): (4391325/3403)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:
```

Displaying Information About SAs at a Peer

To display information about all current IKE SAs at a peer, use the **show crypto isakmp sa** command. The following is sample output from the command:

```
Router1# show crypto isakmp sa

IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
10.10.40.2      10.10.40.1      QM_IDLE          68001 ACTIVE
```

Displaying Information About RSA Public Keys

To display information the RSA public keys configured for your switch, use the **show crypto key mypubkey rsa** command. The following is sample output from the command:

```
Router1# show crypto key mypubkey rsa

% Key pair was generated at: 13:43:52 PST Jan 15 2009
Key name: router1.example.com
 Storage Device: not specified
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
  00D6D5A6 2D42D542 805FAAFB 65B72A10 21AA7ACC F86D6C5A 7E290221 0EEC370B
  FF790FAC AE654451 E038BEAE 7AD85C45 75A2E5D3 95866F6C 6C888C1F 197C3BC3
  3EB02714 A5F43640 D1951900 053F080E 99B547A7 A317C852 BF9B2D91 20D3BABC
  6EA39317 FF22EF39 308F9838 1F395A56 F04473F7 52D41353 3C294E44 B7BDC8A0
  7BB82EBF 405D7853 1E26F8DA 3251FD9F 072E8697 D131F3E2 042D8F4F 0D421361
  846FCB1F B266E0AB 3B2AB6E6 1FA5927A B6E5C29E CFFA77C0 EF07831E D060A950
  C6769919 E2C54267 226FC037 DF996B5D 04D0F0DD 5C0340D4 F8F17CB7 D74531E8
  BF378EFA EADB73A8 7F02370A EE4AA992 F2B7A9D0 DA0CFB9C 2B3C3F2C 79B854EA
  A1020301 0001
% Key pair was generated at: 13:43:53 PST Jan 15 2009
Key name: router1.example.com.server
Temporary key
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C18A19 1B4BE742
  31692B90 FD290B6A CC085CF8 2E9591F0 9B70951C 07E7942B 55AB8540 2BCDEF8D
  A17629D9 8C9369CD AB693089 E4DA150F 6ADF7CFB 78197CF3 84FBC959 90F37EC5
  83B0136D 1A93ACF4 040CD25F 22D7428E 7950E0ED AB45B011 DD020301 0001
```

To display a list of all the RSA public keys stored on your switch (including the public keys of peers that have sent your switch their certificates during peer authentication for IPsec), or to display details of a particular RSA public key stored on your switch, use the **show crypto key pubkey-chain rsa** command. The following is sample output from the command:

```
Router1# show crypto key pubkey-chain rsa

Codes: M - Manually configured, C - Extracted from certificate

Code Usage         IP-Address/VRF         Keyring          Name
C    Signing                              default          X.500 DN name:
                              cn=mscavpn1
                              ou=isbu
                              o=example

C    Signing                              default          router1.example.com
```

Displaying Information About Certificates

To display information about your certificate, the certificate of the CA, and any RA certificates, use the **show crypto pki certificates** command. The following is sample output from the command:

```
Router1# show crypto pki certificates

Certificate
  Status: Available
  Certificate Serial Number: 18FBAA1C000000000062
  Certificate Usage: General Purpose
  Issuer:
    cn=mscavpn1
    ou=isbu
    o=example
  Subject:
    Name: router1.example.com
    Serial Number: 0001925D
    cn=router1
    ou=isbu
    o=example
    hostname=router1.example.com
    serialNumber=1925D
  CRL Distribution Points:
    http://caserver/CertEnroll/mscavpn1.crl
  Validity Date:
    start date: 13:41:42 PST Jan 15 2009
    end   date: 13:51:42 PST Jan 15 2010
  Associated Trustpoints: MSCA

CA Certificate
  Status: Available
  Certificate Serial Number: 30EF1D0A2F5B98A44B0AE1BF562636EC
  Certificate Usage: Signature
  Issuer:
    cn=mscavpn1
    ou=isbu
    o=example
  Subject:
    cn=mscavpn1
    ou=isbu
    o=example
  CRL Distribution Points:
    http://caserver/CertEnroll/mscavpn1.crl
  Validity Date:
    start date: 16:12:15 PST Dec 22 2008
    end   date: 16:19:24 PST Dec 22 2013
  Associated Trustpoints: MSCA
```

Displaying Information About Trustpoints

To display the trustpoints that are configured in the switch, use the **show crypto pki trustpoints** command. The following is sample output from the command:

```
Router1# show crypto pki trustpoints

Trustpoint MSCA:
    Subject Name:
    cn=mscavpn1
    ou=isbu
    o=example
```

```
      Serial Number: 30EF1D0A2F5B98A44B0AE1BF562636EC
Certificate configured.
SCEP URL: http://43.0.111.111:80/certsrv/mscep/mscep.dll
```

Displaying Information About Crypto Sessions

To display status information for active crypto sessions, use the **show crypto session** command. The output will include the following:

- Interface
- IKE peer description, if available
- IKE SAs that are associated with the peer by which the IPsec SAs are created
- IPsec SAs serving the flows of a session

The following is sample output from the command:

```
Router1# show crypto session

Crypto session current status

Interface: Vlan2
Session status: UP-ACTIVE
Peer: 10.10.40.2 port 500
  IKE SA: local 10.10.40.1/500 remote 10.10.40.2/500 Active
  IPSEC FLOW: permit ip host 12.0.0.1 host 13.0.0.1
        Active SAs: 2, origin: crypto map
```

The following is sample output from the command using the **detail** keyword:

```
Router1# show crypto session detail

Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication

Interface: Vlan2
Uptime: 00:00:32
Session status: UP-ACTIVE
Peer: 10.10.40.2 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: router2.example.com
      Desc: (none)
  IKE SA: local 10.10.40.1/500 remote 10.10.40.2/500 Active
          Capabilities:(none) connid:68001 lifetime:23:59:27
  IPSEC FLOW: permit ip host 12.0.0.1 host 13.0.0.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 9 drop 0 life (KB/Sec) 4398172/3567
        Outbound: #pkts enc'ed 9 drop 0 life (KB/Sec) 4398172/3567
```

Displaying Tunnel Interface Information

To display tunnel interface information, use the **show interfaces tunnel** command. The following is sample output from the command:

```
Router# show interfaces tunnel 1
```

```
Tunnel4 is up, line protocol is down
Hardware is Routing Tunnel
Internet address is 10.1.1.1/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
Encapsulation TUNNEL, loopback not set
Keepalive set (10 sec)
Tunnel source 9.2.2.1, destination 6.6.6.2
Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
Tunnel TOS 0xF, Tunnel TTL 128
Checksumming of packets disabled, fast tunneling enabled
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy, fifo
Output queue 0/0, 1 drops; input queue 0/75, 0 drops
30 second input rate 0 bits/sec, 0 packets/sec
30 second output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

Note This sample output does not use the example topology shown at the start of the "Displaying VSPA Configuration Information" section on page 13-6.

Table 13-1 describes significant fields shown in the display.

Table 13-1 show interfaces tunnel Field Descriptions

Field	Description
Tunnel is {up | down}	Interface is currently active and inserted into ring (up) or inactive and not inserted (down).
line protocol is {up | down | administratively down}	Shows line protocol up if a valid route is available to the tunnel destination. Shows line protocol down if no route is available, or if the route would be recursive.
Hardware	Specifies the hardware type.
MTU	Maximum transmission unit of the interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255 (255/255 is 100 percent reliability), calculated as an exponential average over 5 minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes.
Encapsulation	Encapsulation method is always TUNNEL for tunnels.
loopback	Indicates whether loopback is set or not.
Keepalive	Indicates whether keepalives are set or not.

Table 13-1 show interfaces tunnel Field Descriptions (continued)

Field	Description
Tunnel source	IP address used as the source address for the tunnel packets.
destination	IP address of the tunnel destination.
Tunnel protocol	Tunnel transport protocol (the protocol the tunnel is using). This is based on the **tunnel mode** command, which defaults to GRE.
key	(Optional) ID key for the tunnel interface.
sequencing	(Optional) Indicates whether the tunnel interface drops datagrams that arrive out of order.
Last input	Number of hours, minutes, and seconds (or never) since the last packet was successfully received by an interface and processed locally on the switch. Useful for knowing when a dead interface failed. This field is not updated by fast-switched traffic.
output	Number of hours, minutes, and seconds (or never) since the last packet was successfully transmitted by an interface.
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is displayed. If that field overflows, asterisks are displayed.
Last clearing	Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared. Three asterisks (***) indicate the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 231 ms (and less than 232 ms) ago.
Output queue, drops Input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped because of a full queue.

Table 13-1 show interfaces tunnel Field Descriptions (continued)

Field	Description
30 second input rate, 30 second output rate	Average number of bits and packets transmitted per second in the last 30 seconds. The 30-second input and output rates should be used only as an approximation of traffic per second during a given 30-second period. These rates are exponentially weighted averages with a time constant of 30 seconds. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period.
packets input	Total number of error-free packets received by the system.
bytes	Total number of bytes, including data and MAC encapsulation, in the error-free packets received by the system.
no buffer	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernet networks and bursts of noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the minimum packet size of the medium.
giants	Number of packets that are discarded because they exceed the maximum packet size of the medium.
CRC	Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of a station transmitting bad data.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.

Table 13-1 show interfaces tunnel Field Descriptions (continued)

Field	Description
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
abort	Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the far-end transmitter has been running faster than the near-end switch's receiver can handle. This may never be reported on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
collisions	Number of messages retransmitted because of an Ethernet collision. This usually is the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). Some collisions are normal. However, if your collision rate climbs to around 4 or 5 percent, you should consider verifying that there is no faulty equipment on the segment and moving some existing stations to a new segment. A packet that collides is counted only once in output packets.
interface resets	Number of times an interface has been reset. The interface may be reset by the administrator or automatically when an internal error occurs.
restarts	Number of times that the controller was restarted because of errors.

Displaying Information About a BFG Configuration

To display information about a BFG configuration, use the **show redundancy linecard-group** and **show crypto ace redundancy** commands. The following is sample output from the commands:

```
Router# show redundancy linecard-group 1

Line Card Redundancy Group:1 Mode:feature-card
Class:load-sharing
Cards:
Subslot: 1/0
Subslot: 2/0

Router1# show crypto ace redundancy
--------------------------------------
LC Redundancy Group ID            : 1
Pending Configuration Transactions: 0
Current State                     : OPERATIONAL
Number of blades in the group     : 2
Slots
--------------------------------------

Slot: 1 Subslot: 0
Slot state: 0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 0 times
Initialization Timer not running

Slot: 2 Subslot: 0
Slot state: 0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 0 times
Initialization Timer not running
```

Note This sample output does not use the example topology shown at the start of the "Displaying VSPA Configuration Information" section on page 13-6.

Displaying Information About the NHRP Cache

To display information about the Next Hop Resolution Protocol (NHRP) cache, use the **show ip nhrp** and the **show crypto sockets** commands. The following is sample output from the commands:

```
Router# show ip nhrp

10.10.1.75/32 via 10.10.1.75, Tunnel5 created 00:32:11, expire 00:01:46

  Type: dynamic, Flags: authoritative unique registered

  NBMA address: 172.16.175.75

10.10.1.76/32 via 10.10.1.76, Tunnel5 created 00:26:41, expire 00:01:37

  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 172.16.175.76

10.10.1.77/32 via 10.10.1.77, Tunnel5 created 00:31:26, expire 00:01:33

  Type: dynamic, Flags: authoritative unique registered

  NBMA address: 172.17.63.20

Router# show crypto sockets

Number of Crypto Socket connections 1

   Tu0 Peers (local/remote): 9.1.1.1/11.1.1.1
       Local Ident  (addr/mask/port/prot): (9.1.1.1/255.255.255.255/0/47)
       Remote Ident (addr/mask/port/prot): (11.1.1.1/255.255.255.255/0/47)
       IPSec Profile: "MyIpsecProf"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)

Crypto Sockets in Listen state:
Client: "TUNNEL SEC" Profile: "MyIpsecProf" Map-name: "Tunnel0-head-0"
```

Note This sample output does not use the example topology shown at the start of the "Displaying VSPA Configuration Information" section on page 13-6.

Displaying Information About IP Multicast Over a GRE Tunnel

To display information about an IP multicast over a GRE tunnel configuration, enter the **show crypto vlan** and **show ip mroute** commands.

Enter the **show crypto vlan** command to check that the tunnel has been taken over by the VSPA. The following is sample output from the command:

```
Router# show crypto vlan

Interface VLAN 100 on IPSec Service Module port Gi7/0/1 connected to Po1 with crypto map
set map_t3
Tunnel15 is accelerated via IPSec SM in subslot 7/0
```

Enter the **show ip mroute** command and look for the H flag to check that the IP multicast traffic is hardware-switched. The following is sample output from the command:

```
Router# show ip mroute 230.1.1.5

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 230.1.1.5), 01:23:45/00:03:16, RP 15.15.1.1, flags: SJC
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16
```

```
(120.1.0.3, 230.1.1.5), 01:23:46/00:03:25, flags: T
Incoming interface: GigabitEthernet8/1, RPF nbr 0.0.0.0, RPF-MFD
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16, H
```

Note This sample output does not use the example topology shown at the start of the "Displaying VSPA Configuration Information" section on page 13-6.

Troubleshooting Specific Problems on the VSPA

This section provides additional information about troubleshooting specific problems related to the VSPA. It includes the following subsections:

- Clearing IPsec Security Associations, page 13-20
- Troubleshooting Trunk Port Configurations, page 13-20
- Troubleshooting a Blade Failure Group, page 13-21
- Troubleshooting IKE Policy and Transform Sets, page 13-22

Clearing IPsec Security Associations

You can clear (and reinitialize) IPsec security associations by using the **clear crypto sa** command.

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, refer to the **clear crypto sa** command in the *Cisco IOS Security Command Reference*, *Release 12.2*.

If you want to also remove the IKE (phase 1) SAs, follow the **clear crypto sa** command with the **clear crypto isa** command. Alternatively, you can use the **clear crypto session** command to achieve the same result as the **clear crypto sa** and the **clear crypto isa** commands. The **clear crypto session** command supports many of the same parameters as the **clear crypto sa** command.

Troubleshooting Trunk Port Configurations

Caution When you configure an Ethernet port as a trunk port, all the VLANs are allowed on the trunk port by default. This default configuration does not work well with the VSPA and causes network loops. To avoid this problem, you must explicitly specify only the desirable VLANs.

For more information on trunk configuration guidelines, review the "Configuring a Trunk Port" section on page 3-14.

To verify which ports are assigned to the VLAN, enter the **show vlan id** *number* command, using the interface VLAN identifier. Following is an example of a trunk port configuration and the output of the **show vlan id** command:

```
Router# show run interface gi 1/3
Building configuration...
```

```
Current configuration : 175 bytes
!
interface GigabitEthernet1/3
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502-504,1002-1005
 switchport mode trunk
 no ip address
end


Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi7/0/1 connected to VLAN 502 with crypto
map set testtag_1
Interface VLAN 3 on IPSec Service Module port Gi7/0/1 connected to VLAN 503 with crypto
map set testtag_2
Interface VLAN 4 on IPSec Service Module port Gi7/0/1 connected to VLAN 504 with crypto
map set testtag_3

Router# show vlan id 2

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
2    VLAN0002                         active    Gi7/0/1

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
2    enet  100002     1500  -      -      -        -    -        0      0

Remote SPAN VLAN
----------------
Disabled

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------

Router# show vlan id 502

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
502  VLAN0502                         active    Gi1/3, Gi7/0/2

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
502  enet  100502     1500  -      -      -        -    -        0      0

Remote SPAN VLAN
----------------
Disabled

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------

Router#
```

Troubleshooting a Blade Failure Group

To enable VSPA debugging for a blade failure group, enter the **debug crypto ace b2b** command:

```
Router# debug crypto ace b2b
```

```
ACE B2B Failover debugging is on
```

Troubleshooting IKE Policy and Transform Sets

Any IPsec transforms or IKE encryption methods that the current hardware does not support should be disabled; they are ignored whenever an attempt to negotiate with the peer is made.

If a user enters an IPsec transform or an IKE encryption method that the hardware does not support, a warning message will be generated. These warning messages are also generated at boot time. When an encrypted card is inserted, the current configuration is scanned. If any IPsec transforms or IKE encryption methods are found that are not supported by the hardware, a warning message will be generated.

Using Crypto Conditional Debug

The crypto conditional debug feature provides three command-line interface (CLI) commands that allow you to debug an IP Security (IPsec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPsec operations and reducing the amount of debug output, you can better troubleshoot a switch with a large number of tunnels.

The crypto conditional debug commands—**debug crypto condition**, **debug crypto condition unmatched**, and **show crypto debug-condition**—allow you to specify conditions (filter values) in which to generate and display debug messages related only to the specified conditions.

Table 13-2 lists the supported condition types.

Table 13-2 Supported Condition Types for Crypto Conditional Debug Commands

Condition Type (Keyword)	Description
connid	An integer between 1 and 32766. Relevant debug messages will be shown if the current IPsec operation uses this value as the connection-ID to interface with the crypto engine.
flowid	An integer between 1 and 32766. Relevant debug messages will be shown if the current IPsec operation uses this value as the flow-ID to interface with the crypto engine.
fvrf	The name string of a virtual private network (VPN) routing and forwarding (VRF) instance. Relevant debug messages will be shown if the current IPsec operation uses this VRF instance as its front-door VRF (FVRF).
ivrf	The name string of a VRF instance. Relevant debug messages will be shown if the current IPsec operation uses this VRF instance as its inside VRF (IVRF).

Table 13-2 Supported Condition Types for Crypto Conditional Debug Commands (continued)

Condition Type (Keyword)	Description
peer group	A Unity group name string. Relevant debug messages will be shown if the peer is using this group name as its identity.
peer hostname	A fully qualified domain name (FQDN) string. Relevant debug messages will be shown if the peer is using this string as its identity.
peer ipv4	A single IP address. Relevant debug messages will be shown if the current IPsec operation is related to the IP address of this peer.
peer subnet	A subnet and a subnet mask that specify a range of peer IP addresses. Relevant debug messages will be shown if the IP address of the current IPsec peer falls into the specified subnet range.
peer username	A username string. Relevant debug messages will be shown if the peer is using this username as its identity.
spi	A 32-bit unsigned integer. Relevant debug messages will be shown if the current IPsec operation uses this value as the SPI.

Note If **connid**, **flowid**, or **spi** is used as a debug condition, the debug messages for a related IPsec flow are generated. An IPsec flow has two connection-IDs, flow-IDs, and SPI values—one inbound and one outbound. Either one of the two connection-IDs, flow-IDs, and SPI values can be used as the debug condition that triggers debug messages for the IPsec flow.

Crypto Conditional Debug Configuration Guidelines and Restrictions

When configuring crypto conditional debug, follow these guidelines and restrictions:

- This feature does not support debug message filtering for hardware crypto engines.
- Although conditional debugging is useful for troubleshooting peer-specific or functionality-related Internet Key Exchange (IKE) and IPsec problems, conditional debugging may not be able to define and check large numbers of debug conditions.
- Because extra space is needed to store the debug condition values, additional processing overhead is added to the CPU and memory usage is increased. Thus, enabling crypto conditional debugging on a switch with heavy traffic should be used with caution.
- Your switch will perform conditional debugging only after at least one of the global crypto debug commands—**debug crypto isakmp**, **debug crypto ipsec**, or **debug crypto engine**—has been enabled. This requirement helps to ensure that the performance of the switch will not be impacted when conditional debugging is not being used.

Enabling Crypto Conditional Debug Filtering

To enable crypto conditional debug filtering, perform the following tasks:

	Command	Purpose
Step 1	Router# **enable**	Enables privileged EXEC mode.
Step 2	Router# **debug crypto condition** [**connid** *integer* **engine-id** *integer*] [**flowid** *integer* **engine-id** *integer*] [**fvrf** *string*] [**ivrf** *string*] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** ipaddress] [**subnet** *subnet mask*] [**username** *string*]] [**spi** *integer*] [**reset**]	Defines conditional debug filters. See Table 13-2 for descriptions of values.
Step 3	Router# **show crypto debug-condition** {[**peer**] [**connid**] [**spi**] [**fvrf**] [**ivrf**] [**unmatched**]}	Displays crypto debug conditions that have already been enabled in the switch.
Step 4	Router# **debug crypto isakmp**	Enables global IKE debugging.
Step 5	Router# **debug crypto ipsec**	Enables global IPsec debugging.
Step 6	Router# **debug crypto engine**	Enables global crypto engine debugging.
Step 7	Router# **debug crypto condition unmatched** [**isakmp** | **ipsec** | **engine**]	(Optional) Displays debug conditional crypto messages when no context information is available to check against debug conditions. If none of the optional keywords are specified, all crypto-related information will be shown.

Disabling Crypto Conditional Debugging

Before you disable crypto conditional debugging, you must first disable any crypto global debug CLIs that you have issued. You can then disable crypto conditional debugging. To disable crypto conditional debugging, enter the following command:

```
Router# debug crypto condition reset
```

Enabling Crypto Error Debug Messages

Enabling the **debug crypto error** command displays only error-related debug messages, which allows you to easily determine why a crypto operation, such as an IKE negotiation, has failed within your system. To enable crypto error debug messages, enter the following command from privileged EXEC mode:

```
Router# debug crypto {isakmp | ipsec | engine} error
```

Note When enabling this command, ensure that global crypto debug commands are not enabled; otherwise, the global commands will override any possible error-related debug messages.

For complete configuration information for crypto conditional debug support, refer to this URL:

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

Preparing for Online Insertion and Removal of a VSPA or SSC-600

The Catalyst 6500 Series switch supports online insertion and removal (OIR) of the SSC-600, in addition to each of the VSPAs. You can remove an SSC-600 with its VSPAs still intact, or you can remove a VSPA independently from the SSC-600, leaving the SSC-600 installed in the switch.

An SSC-600 can remain installed in the switch with one VSPA remaining active while you remove another VSPA from one of the SSC-600 subslots. If you are not planning to immediately replace a VSPA into the SSC-600, then be sure to install a blank filler plate in the subslot. The SSC-600 should always be fully installed with either functional VSPAs or blank filler plates.

Note When you remove a VSPA that has some ports participating in crypto connection, the crypto connections remain intact. When you reinsert the same type of VSPA into the same slot, the traffic starts to run again on all the crypto connections. To move the VSPA to a different slot, you must first manually remove the crypto connections before removing the VSPA. You can enter the **no crypto connect vlan** command from any interface when the associated physical port is removed.

For more information about activating and deactivating VSPAs in preparation for OIR, see the *Cisco VPN Services Port Adapter Hardware Installation Guide*.

INDEX