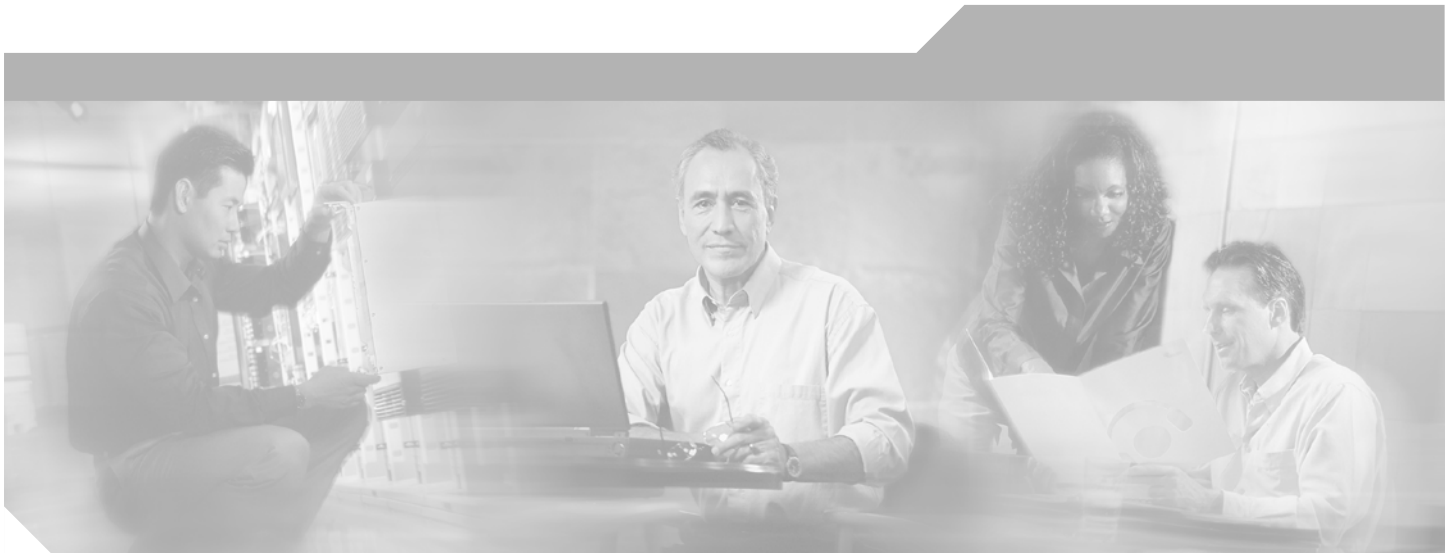


Cisco Network-Based IPSec VPN Solution Release 1.5 Solution Operations, Maintenance, and Troubleshooting Guide

May, 2003

Corporate 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: 408 526-4100

Customer Order Number:
Text Part Number: OL-3134-01

About This Guide

This operations, maintenance, and troubleshooting guide is designed to be used with the Cisco Network-Based IPSec VPN Release 1.5.

This guide provides a high-level view only, and does not attempt to describe all of the features and details of the applications. Always rely on the standard documentation for those applications for the details of installing, using, and troubleshooting. Links to the latest documentation appear in the appropriate chapters of this guide.

The most recent versions of this guide and related documentation can be found at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/vpn/solution/aswan15/index.htm.

Note All Cisco solutions documents can be found under Cisco Solutions, at the following URL: http://www.cisco.com/univercd/cc/td/doc/solution/index.htm

This preface presents the following major topics:

Document and Solution Release

This release of this document covers Release 1.5 of the Cisco network-based IPSec solution.

This solution is referred to generically in this document.

Audience

It is assumed that administrators of the Cisco network-based IPSec VPN solution Release 1.5 have experience with installation of the products covered by this solution. In addition, it is assumed that the administrator understands the procedures required to upgrade and troubleshoot remote access methods at a basic level. Typical users of this guide include the following groups:

- Customers with technical networking background and experience
- System administrators who are familiar with the fundamentals of router-based internetworking, but who may not be familiar with Cisco IOS software
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software

Document Organization

The chapters of this document are as follows:

This guide describes software installation and configuration procedures which are presented in the following chapters:

- Chapter 1, “Introduction,” provides information on using this guide and Operations and Maintenance.
- Chapter 2, “Upgrade Considerations” presents information on upgrading solution components and customer premises equipment.
- Chapter 3, “Troubleshooting” presents general troubleshooting information, as well as troubleshooting information on solution components, customer premises equipment, IPSec, strategy, common IPSec error messages, and supported topologies.
- Appendix A, “Troubleshooting Commands” provides information on **show** commands, **clear** commands, and **debug** commands
- Appendix B, “Sample Problem Scenarios” presents transform set mismatch, access list mismatch, key mismatch, ISAKMP mismatch, crypto map not applied, missing SAs, and transform and proposal mismatches.
- Index

Related Documents

Cisco Network-Based IPSec VPN Solution Release 1.5 Documentation Set

In addition to this guide, the Cisco network-based IPSec VPN solution Release 1.5 documentation set includes the following documents that you can find at:

- *Cisco Network-Based IPSec VPN Solution Release 1.5 Overview and Planning Guide*
- *Cisco Network-Based IPSec VPN Solution Release 1.5 Implementation Guide*
- *Release Notes for the Cisco Network-Based IPSec VPN Solution Release 1.5*

You can find these documents at http://www.cisco.com/univercd/cc/td/doc/product/vpn/solution/index.htm.

Related Documentation

The majority of the other documents referred to in this document are available online. In the electronic (PDF) version of this document you can click the URL (Uniform Resource Locator, often referred to as the website) associated with the title of a document, and the selected document appears within the Adobe Acrobat application window. You can also use the Text Select Tool (third icon from the top, at the left of the Acrobat application window) to copy a URL from the PDF document and paste it into the location field of your browser.

Viewing Online Documents in Your Browser

As you click links, the files you select may be added to the current document. When you close the file, you are prompted to save the file. (You are not able to save the file to a CD.) If you choose not to save the larger file that is created, click **No** when prompted to save the file. However, if you acquire documents that you want to save in a new file, you can save that file to another disk or drive with a new name. Set the following preferences within the Acrobat application to open web links in your browser, rather than within Acrobat.

You can obtain the latest version of Adobe Acrobat Reader at http://www.adobe.com.

Step 1 Select the browser you want to use.

a. From the Acrobat main menu, choose **File** > **Preferences** > **Weblink**. The Weblink Preferences window opens.

b. In the Weblink Preferences window, click **Browse** (or **Select**) and locate the browser you wish to use.

c. Then select **Connection Type** from the pull-down menu. Choose **Standard** if your browser is not listed.

d. Click **OK** to save your settings.

Step 2 Make sure that Acrobat opens weblinks in your browser.

a. From the Acrobat main menu, choose **File** > **Preferences** > **Web Capture**. The Web Capture Preferences window opens.

b. In the Web Capture Preferences Window, choose **Open Weblinks**: In Web Browser.

c. Click **OK** to save your settings.

Document Conventions

Command descriptions use the following conventions:

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 }	Alternate keywords are grouped in braces and separated by vertical bars.
[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 the following conventions:

`screen` font	Terminal sessions and information the system displays are in `screen` font.
`boldface screen` font	Information you must enter is in **`boldface screen`** font.
italic screen font	Arguments for which you supply values are in *italic screen* font.
⟶	This pointer highlights an important line of text in an example.
< >	Nonprinting characters, such as passwords, are in angle brackets in contexts where italic font is not available. Also used to represent variables in command line examples where `screen` font is used.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Notes use the following conventions:

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

Cautions use the following conventions:

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

Tips use the following conventions:

Tip This symbol means the following information *will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Obtaining Documentation

Cisco provides several ways to obtain documentation, technical assistance, and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com/univercd/home/home.htm

You can access the Cisco website at this URL:

http://www.cisco.com

International Cisco websites can be accessed from this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which may have shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Registered Cisco.com users can order the Documentation CD-ROM (product number DOC-CONDOCCD=) through the online Subscription Store:

http://www.cisco.com/go/subscription

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/en/US/partner/ordering/index.shtml

- Registered Cisco.com users can order the Documentation CD-ROM (Customer Order Number DOC-CONDOCCD=) through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can submit comments electronically on Cisco.com. On the Cisco Documentation home page, click **Feedback** at the top of the page.

You can e-mail your comments to bug-doc@cisco.com.

You can submit your comments by mail by using the response card behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com, which includes the Cisco Technical Assistance Center (TAC) Website, as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from the Cisco TAC website. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC website, including TAC tools and utilities.

Cisco.com

Cisco.com offers a suite of interactive, networked services that let you access Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

To obtain customized information and service, you can self-register on Cisco.com at this URL:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC website and the Cisco TAC Escalation Center. The avenue of support that you choose depends on the priority of the problem and the conditions stated in service contracts, when applicable.

We categorize Cisco TAC inquiries according to urgency:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Cisco TAC Website

You can use the Cisco TAC website to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC website, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC website. Some services on the Cisco TAC website require a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

http://tools.cisco.com/RPF/register/register.do

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC website, you can open a case online at this URL:

http://www.cisco.com/en/US/support/index.html

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC website so that you can describe the situation in your own words and attach any necessary files.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before you call, check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, have your service agreement number and your product serial number available.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- The *Cisco Product Catalog* describes the networking products offered by Cisco Systems as well as ordering and customer support services. Access the *Cisco Product Catalog* at this URL:

  http://www.cisco.com/en/US/products/products_catalog_links_launch.html

- Cisco Press publishes a wide range of networking publications. Cisco suggests these titles for new and experienced users: *Internetworking Terms and Acronyms Dictionary, Internetworking Technology Handbook, Internetworking Troubleshooting Guide,* and the *Internetworking Design Guide.* For current Cisco Press titles and other information, go to Cisco Press online at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco monthly periodical that provides industry professionals with the latest information about the field of networking. You can access *Packet* magazine at this URL:

  http://www.cisco.com/en/US/about/ac123/ac114/about_cisco_packet_magazine.html

- *iQ Magazine* is the Cisco monthly periodical that provides business leaders and decision makers with the latest information about the networking industry. You can access *iQ Magazine* at this URL:

  http://business.cisco.com/prod/tree.taf%3fasset_id=44699&public_view=true&kbns=1.html

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in the design, development, and operation of public and private internets and intranets. You can access the *Internet Protocol Journal* at this URL:

  http://www.cisco.com/en/US/about/ac123/ac147/about_cisco_the_internet_protocol_journal.html

- Training—Cisco offers world-class networking training, with current offerings in network training listed at this URL:

  http://www.cisco.com/en/US/learning/le31/learning_recommended_training_list.html

Introduction

This chapter provides general information on operations and maintenance. It is assumed that all of the components of this solution have been correctly installed, configured, and provisioned, and that a basic solution network is in service.

For an overview of the topics in each chapter, refer to Document Organization, page Document Organization.

This chapter covers the following major topics:

Note This guide is meant to provide a high-level view only, and does not attempt to cover all the features and details of the related applications. Always rely on the standard documentation for those applications for the details of installing, using, and troubleshooting. Links to the latest documentation are provided in the appropriate chapters of this guide. While this document has tried to be as current as possible, the documentation for applications is subject to revision. Information is subject to reorganization, section headings are subject to renaming, and hyperlinks are subject to change.

To properly operate and maintain the Cisco network-based IPSec VPN solution Release 1.5, make sure you have read the following documents:

- *Cisco Network-Based IPSec VPN Solution Release 1.5 Overview and Planning Guide*
- *Cisco Network-Based IPSec VPN Solution Release 1.5 Implementation Guide*
- *Release Notes for the Cisco Network-Based IPSec VPN Solution Release 1.5*

These are available at the Network-Based IPSec VPN website, at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/vpn/solution/index.htm

Operation and Maintenance Tasks

There are a variety of tasks that we recommend you attend to on a routine basis. Other tasks can be performed as needed, although you may want to schedule certain critical tasks depending on the needs of your network. This section presents the following topics:

- Routine Tasks
- General Operations and Maintenance Guidelines

Routine Tasks

Table 1-1 identifies, at a high level, the tasks that service providers must perform on a daily, weekly, monthly, and annual basis to operate and maintain the health of their Cisco network-based IPSec VPN solution Release 1.5 network.

Table 1-1 Routine Operations and Maintenance Tasks

Frequency	Task	Notes
Daily	Monitor alarms from all platforms in network	View alarms on the platform directly, or use ISC Version 3.0.
	Review system logs	View alarms on the platform directly.
Weekly	Back up all relevant data and configuration information for all network platforms.	The service provider must develop a process for determining relevant data and platforms.
	Visit Cisco websites regularly to see if solution release notes have been updated to recommend new software releases.	As new releases become available and caveats are added or resolved, the solution release notes are updated.
Monthly	In a maintenance window, test the ability of the solution components to failover from active to standby.	If failover is not tested regularly, redundant equipment is of little value.
Annually	Plan for the possibility of a major network upgrade of Cisco software.	Besides Cisco IOS software, changes in hardware, particularly in memory, may be required.
	Review overall network traffic requirements to ensure that traffic is being served properly by existing network.	—

General Operations and Maintenance Guidelines

To maintain your solution network, follow these general best practices:

- Develop a general strategy for monitoring the Cisco 7204 router and Cisco 7206 router using the ISC Version 3.0.
- Use the software tools available on each platform to monitor and report critical data such as fault alarms.
- Check equipment status.
- Regularly monitor system log entries.

- Regularly issue status queries, using either a variety of GUI element-management tools, or Cisco IOS software (see "Using Cisco IOS Software for Operations and Maintenance") or MML (Man Machine Language) commands entered at the CLI.
- When you remove or add any solution components, read the most recent applicable hardware and software documents.
- Be sure that you provide for redundancy before upgrading any solution components.

Table 1-2 lists general operations and maintenance guideline links for the hardware components of the Cisco network-based IPSec VPN solution Release 1.5.

Table 1-2 General Operations and Maintenance Guidelines for Solution Components

Component	Maintenance Guideline Links
Cisco 7204 router	http://www.cisco.com/univercd/cc/td/doc/product/core/7204/7204ig/main4icg.htm
Cisco 7206 router	http://www.cisco.com/univercd/cc/td/doc/product/core/7206/7206ig/addpr6ug.htm
AAA server	Any RADIUS server (such as Cisco Access Registrar) that understands Cisco AV pairs can be used. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fsecsp/scfrad.htm#79543

Using IP Solution Center Version 3.0 for Operations and Maintenance

The Cisco IP Solution Center (ISC) Version 3.0 provides management of IPSec VPN services throughout the service life cycle including service provisioning and activation on customer-edge and provider-edge routers, service auditing and service-level agreement (SLA).

For service providers using the IPSec or MPLS transport framework, the Cisco ISC Version 3.0 provides a full complement of provisioning, monitoring, and administration tools that simplify the inherent complexities of managing a VPN infrastructure.

You can use ISC Version 3.0 to direct the operations of the following components of the Cisco network-based IPSec VPN solution Release 1.5:

- Cisco 7204
- Cisco 7206

You can also us the Cisco ISC Version 3.0 application to direct the operations of the following customer premise equipment:

- Cisco PIX Firewall with EzVPN client
- Cisco VPN 3002 hardware client
- Cisco 800 series routers
- Cisco 1700 series routers
- Cisco 2600 series routers
- Cisco 3600 series routers
- Cisco 7200 series routers

For information on using the Cisco ISC Version 3.0, refer to Cisco ISC Version 3.0 documentation at the following URL: http://www.cisco.com/univercd/cc/td/doc/.

Routine Maintenance

Use ISC Version 3.0 to perform the following operational tasks.

Defining IPSec Networks and Customers

- Define the IPSec network elements
  - Defining target routers
  - Defining targets by importing device configuration files
  - Specifying the transport mechanism for multiple edge devices
  - Setting passwords for multiple edge devices
  - Specifying the IP address for the router's loopback interface
  - Setting SNMPv3 parameters for VPNSC target routers
- Adding a new edge device to the network
  - Specifying device passwords, and SNMP and Telnet parameters
  - Specifying IP addresses for the new target
  - Specifying SNMPv3 parameters
  - Populating IP address information to the repository
  - Deleting targets from a network
- Creating default policies
- Defining a new IPSec VPN customer
  - Defining customer sites
  - Assigning an edge device to a site
  - Assigning the management interface
  - Defining devices as managed and SA agent enabled
  - Defining a device's secured, nonsecured, and tunnel endpoint interfaces
  - Creating customer-specific policies
  - Deleting a customer

Defining VPNs and Provisioning Service Requests

- Creating a new IPSec VPN and provisioning a service request
  - Defining a service request for the VPN
  - Specifying edge devices in the service request
  - Assigning secondary edge devices
- Deploying service requests
  - Redeploying after changing a deployed service request
- Opening an existing service request
- Getting detailed information on service requests

- Decommissioning a VPN service
  - Creating a request to decommission a VPN service
  - Deploying and auditing the decommissioned VPN service request
  - Removing a VPN service from the repository
- Editing a device's configuration file
- Auditing IPSec VPN service requests

Provisioning the Cisco VPN 3002

- Specifying a transport mechanism for configuration files
- Setting passwords for multiple VPN 3000 devices
- Performing additional setup tasks
- Assigning VPN 3000 devices to their sites
- Specifying a management interface
- Defining the device's public and private interfaces
- Specifying address pools for remote access service
- Creating default policies
- Defining VPN groups for remote access service
- Configuring AA servers
- Creating a new VPN
- Creating a remote-access service request
- Creating a LAN-to-LAN service request

Monitoring IPSec VPN Performance

- Updating router configuration information
- Monitoring performance through service-level agreements (SLAs)
- Provisioning SLAs
- Collecting only changed configuration files
- Enabling traps for SLA data
- Disabling traps
- Collecting interface statistics
- Retrieving SLA definitions from edge devices
- Retrieving SA agent data
- Viewing data reports
- Viewing SLA reports

Using Cisco IOS Software for Operations and Maintenance

Monitoring Network Performance Using Cisco IOS Commands

It is critical to monitor the operating environments of network devices, such as voltage, temperature, and airflow, and ensure that they are operating within specifications. Software components such as buffers and memory can have a significant impact on the protocols running on the device.

CPU utilization is a useful performance indicator on the Cisco devices. By measuring CPU use over time, a trend can be established to determine traffic patterns. Devices running constantly at high utilization levels can affect the overall performance of forwarding and processing packets. CLI commands on the Cisco devices can display the CPU utilization and information on running processes. You can access information returned on the CPU load by means of objects defined in MIB files.

Target Platforms

The Cisco IOS software CLI manages the following components of the Cisco network-based IPSec VPN solution Release 1.5:

- Cisco 7204 router
- Cisco 7206 router

Cisco IOS Software References

The following paragraphs provide references to URLs for Cisco IOS, system error messages, and debug commands.

Cisco IOS Software

For the details of Cisco IOS software Release 12.2, refer to the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/

System Error Messages

The system software sends these error messages to the console (and, optionally, to a logging server on another system) during operation. Not all system error messages indicate problems with your system. Some are purely informational, and others may help diagnose problems with communications lines, internal hardware, or the system software.

See Cisco IOS System Error Messages, Cisco IOS Release 12.2 at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122sems/

Debug Command Reference

For details on debugging commands, refer to Cisco IOS Debug Command Reference, Cisco IOS Release 12.2, at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122debug/

Upgrade Considerations

This chapter describes general considerations that customers need to make when upgrading the Cisco network-based IPSec VPN Release 1.5. In general, redundancy must be provided to support upgrades without affecting service availability.

Although it is the service provider's decision to determine the service availability required by its customers, it remains good practice to ensure that traffic is not interrupted. In addition to providing redundancy to cover outages of equipment or communications channels, it is necessary to provide redundancy to support traffic during upgrades of the solution components.

Caution It is the responsibility of the service provider to engineer the network in such a way as to provide the required service availability for their customers.

This chapter presents the following major topics:

- Upgrading Solution Components
- Upgrading Customer Premises Equipment

Before You Upgrade

Before you begin the solution upgrade, make sure that you keep the following issues in mind:

- Begin the upgrade process during a maintenance window or low-traffic period, and plan for system downtime accordingly.
- Ensure that all systems are working properly and that there are no alarms. Make sure there is no major congestion.
- Review the software terms and conditions.
- Review the software and tool requirements and procedural overview.
- Review the hardware and software requirements found in the *Release Notes for Cisco Network-Based IPSec VPN Solution 1.5.*
- Gather all required software and hardware. Your system must meet the minimum requirements as shown in the review the hardware and software requirements found in the *Release Notes for Cisco Network-Based IPSec VPN Solution Release 1.5* at http://www.cisco.com/univercd/cc/td/doc/product/vpn/solution/index.htm.
- The target machine must have a terminal connected by using a serial cable inserted into the console port.

Upgrading Solution Components

This section provides upgrade information for the following components of the Cisco network-based IPSec VPN solution Release 1.5:

- Cisco 7204 and Cisco 7206 routers
- AAA/RADIUS server

Upgrading the Cisco 7204 and the Cisco 7206 Routers

This section provides information for upgrading your Cisco IOS software images on the Cisco 7204 and 7206 routers.

Note The most recent information about upgrading the Cisco IOS software can be found in the release notes for your software.

To upgrade software on the Cisco 7204 and 7206 routers, see the information at http://www.cisco.com/warp/public/130/sw_upgrade_highendrouters_23233.html.

Upgrading RADIUS Servers

Any RADIUS server (such as Cisco Access Registrar) that understands Cisco AV pairs can be used. It is essential that client configuration information (authorization and authentication) not be lost. For information on upgrading Cisco Access Register servers, see:

http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cnsar/3_0/install/upgrade.htm.

Upgrading Customer Premise Equipment

This section provides upgrade information for the following customer premises equipment used with the Cisco network-based IPSec VPN solution Release 1.5:

- Cisco VPN 3002 hardware client
- Cisco 800 series routers
- Cisco 1700 series routers
- Cisco 2600 series routers
- Cisco 3600 series routers
- Cisco 7200 series routers

Note The most recent information about upgrading the Cisco IOS software can be found in the release notes for your software.

Table 2-1 Customer Premise Equipment Upgrade Information

Customer Premise Equipment	Upgrade Information
Cisco VPN 3002 hardware client	See the information at http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/vpn3kco/vcoug/usr_3_0/admin.htm#xtocid2259936
Cisco 800 series routers	See the information at http://www.cisco.com/warp/public/63/IOSupgrade_800.shtml
Cisco 1700 series routers	See the information at http://www.cisco.com/warp/public/130/sw_upgrade_proc_ram.shtml
Cisco 2600 series routers	See the information at http://www.cisco.com/warp/public/130/sw_upgrade_proc_ram.shtml
Cisco 3600 series routers	See the information at http://www.cisco.com/warp/public/130/sw_upgrade_proc_ram.shtml
Cisco 7200 series routers	See the information at http://www.cisco.com/warp/public/130/sw_upgrade_highendrouters_23233.html

Upgrading from Previous Versions of the Cisco Network-Based IPSec VPN Solution

The VRF-Aware IPSec feature introduces IP security (IPSec) tunnel mapping to Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). Using the VRF-Aware IPSec feature, you can map IPSec tunnels to virtual routing and forwarding (VRF) instances using single, public-facing addresses.

The VRF Aware IPSec feature in the Cisco network-based IPSec VPN solution Release 1.5 requires that you change your existing configurations. For more information about the VRF aware IPSec feature, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t15/ft_vrfip.htm.

The sample configurations that follow indicate the changes you must make to your existing configurations. These samples include the following:

- Site-to-Site Configuration Migration, page 2-3
- Remote Access Configuration Upgrade, page 2-5
- Combination Site-to-Site and Remote Access Configuration, page 2-7

Site-to-Site Configuration Migration

The following configurations show the changes necessary for site-to-site configuration migration from a previous version of this solution to the current Cisco network-based IPSec VPN solution Release 1.5.

Previous Version Site-to-Site Configuration

The following configuration uses a previous version of a site-to-site network-based IPSec VPN solution:

```
crypto isakmp key VPN1 address 172.21.25.74
crypto isakmp key VPN2 address 172.21.21.74
!
```

```
crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
!
crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
!
interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
!
```

New Version Site-to-Site Configuration

The following configuration is an upgraded version of the same site-to-site configuration to the Cisco network-based IPSec VPN solution Release 1.5 solution.

Note You must change to keyrings. The VRF Aware IPSec feature requires keys to be associated with a VRF if the IKE local endpoint is in the VRF.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
crypto keyring VPN2-KEYS vrf VPN2
 pre-shared-key address 172.21.21.74 key VPN2
!
crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
!
crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
!
interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
interface FastEthernet0/0.2
```

```
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
!
```

Remote Access Configuration Upgrade

The following configurations show the changes necessary for a remote access configuration upgrade from an earlier version of this solution to the Cisco network-based IPSec VPN solution Release 1.5.

Previous Version Remote Access Configuration

```
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 reverse-route
!
crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 reverse-route
!
!
crypto map VPN1 client authentication list VPN1-RA-LIST
crypto map VPN1 isakmp authorization list VPN1-RA-LIST
crypto map VPN1 client configuration address initiate
crypto map VPN1 client configuration address respond
crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
crypto map VPN2 client authentication list VPN2-RA-LIST
crypto map VPN2 isakmp authorization list VPN2-RA-LIST
crypto map VPN2 client configuration address initiate
crypto map VPN2 client configuration address respond
crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
!
```

New Version Remote Access Configuration

In this instance there is no migration; we recommend that you change to the following configuration:

```
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
 client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
!
```

Combination Site-to-Site and Remote Access Configuration

The following configurations show the changes necessary for a site-to-site and remote access configuration upgrade from a previous version of the network-based IPSec VPN solution to the Cisco network-based IPSec VPN solution Release 1.5.

Previous Version Site-to-Site and Remote Access Configuration

```
crypto isakmp key VPN1 address 172.21.25.74 no-xauth
crypto isakmp key VPN2 address 172.21.21.74 no-xauth
!
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 reverse-route
!
crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 reverse-route
!
crypto map VPN1 client authentication list VPN1-RA-LIST
crypto map VPN1 isakmp authorization list VPN1-RA-LIST
crypto map VPN1 client configuration address initiate
crypto map VPN1 client configuration address respond
crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
crypto map VPN2 client authentication list VPN2-RA-LIST
crypto map VPN2 isakmp authorization list VPN2-RA-LIST
crypto map VPN2 client configuration address initiate
crypto map VPN2 client configuration address respond
crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
```

```
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Site-to-Site and Remote Access Configuration

You must migrate to this configuration:

Note For site-to-site configurations that do not require XAUTH, configure an ISAKMP profile without XAUTH configuration.For remote access configurations that require XAUTH, configure an isakmp profile with XAUTH.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
crypto keyring VPN2-KEYS vrf VPN2
 pre-shared-key address 172.21.21.74 key VPN2
!
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
crypto isakmp profile VPN1
 keyring VPN1-KEYS
 match identity address 172.21.25.74 VPN1
!
crypto isakmp profile VPN2
 keyring VPN2-KEYS
 match identity address 172.21.21.74 VPN2
!
crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
 client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
crypto dynamic-map VPN2-RA 1
```

```
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 set isakmp-profile VPN1
 match address 101
crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 set isakmp-profile VPN2
 match address 102
crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Troubleshooting

This chapter provides information on troubleshooting the Cisco network-based IPSec VPN solution Release 1.5. It provides links to information on troubleshooting solution components as well as IPSec. It also provides solution-specific troubleshooting.

Troubleshooting Solution Components

Cisco 7200 Series Internet Router Troubleshooting

See Cisco 7202 and Cisco 7204 router troubleshooting at:

http://www.cisco.com/univercd/cc/td/doc/product/core/7200vx/7200trbl.htm

RADIUS Troubleshooting

Any RADIUS server (such as Cisco Access Registrar) that understands Cisco AV pairs can be used as an AAA server. For information on troubleshooting the Cisco Access Registrar, see http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cnsar/3_0/users/trouble.htm.

Troubleshooting Customer Premise Equipment

This section provides references to troubleshooting information for the following customer premises equipment used with the Network-Based IPSec VPN:

- Cisco VPN 3002 hardware client
- Cisco 800 series routers
- Cisco 1700 series routers
- Cisco 2600 series routers
- Cisco 3600 series routers
- Cisco 7200 series routers

Table 3-1 Customer Premise Equipment Troubleshooting Information

Customer Premise Equipment	Upgrade Information
Cisco VPN 3002 hardware client	See the information at: http://www.cisco.com/en/US/products/hw/vpndevc/ps2286/products_user_guide_chapter09186a00800bcd46.html.
Cisco 800 series routers	See the information at: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_fix/800/800hwins/trouble.htm.
Cisco 1700 series routers	See the information at: http://www.cisco.com/en/US/products/hw/routers/ps221/prod_troubleshooting.html.
Cisco 2600 series routers	See the information at: http://www.cisco.com/warp/public/108/hwts_2600_16138.html.
Cisco 3600 series routers	See the information at: http://www.cisco.com/en/US/products/hw/routers/ps274/prod_troubleshooting.html.
Cisco 7200 series routers	See the information at: http://www.cisco.com/warp/public/63/hwts_7200_16122.html.

Troubleshooting IP Solution Center Version 3.0

See the information at: http://www.cisco.com/univercd/home/home.htm.

Troubleshooting Cisco IOS Software

For general information on troubleshooting Cisco IOS software, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/taclinks.htm.

Troubleshooting Cisco VPNS

For information on troubleshooting Cisco VPNS, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_configuration_guide_chapter09186a00800d97ca.html#1002964.

Troubleshooting IPSec

Troubleshooting IPSec-related connectivity problems is fairly straightforward if you use a logical and methodical approach.

When IPSec works correctly, the normal flow of events is as follows:

- Router receives *interesting* traffic, which is traffic that must be encrypted.

- Router negotiates and sets up an IKE tunnel with the remote peer specified in the configuration.
- Router negotiates IPSec tunnels (one for the inbound direction and one for the outbound) with the remote peer.
- Routers pass encrypted traffic back and forth.

ISAKMP and IPSec Troubleshooting Commands

Note As with all other debug commands, **ISAKMP** and **IPSec debug** commands consume CPU cycles and produce copious output.

The ISAKMP and IPSec troubleshooting commands provide detailed information about IPSec failure. See Appendix A, "Troubleshooting Commands," for more examples of these commands as well as the information they provide.

Before using these commands, see the information at http://www.cisco.com/warp/public/793/access_dial/debug.html

For general information on using **ISAKMP** and **IPSec debug** commands, see http://www.cisco.com/en/US/tech/tk648/tk367/technologies_tech_note09186a00800949c5.shtml.

debug crypto engine

Use this command to view encrypted traffic. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122debug/dbfclns.htm#xtocid14.

The following is sample output from the debug crypto engine command. The first sample output shows messages from a router that successfully generates RSA keys. The second sample output shows messages from a router that decrypts the RSA key during Internet Key Exchange (IKE) negotiation.

```
Router# debug crypto engine
00:25:13:CryptoEngine0:generate key pair
00:25:13:CryptoEngine0:CRYPTO_GEN_KEY_PAIR
00:25:13:CRYPTO_ENGINE:key process suspended and continued
00:25:14:CRYPTO_ENGINE:key process suspended and continuedcr
Router# debug crypto engine
00:27:45:%SYS-5-CONFIG_I:Configured from console by console
00:27:51:CryptoEngine0:generate alg parameter
00:27:51:CRYPTO_ENGINE:Dh phase 1 status:0
00:27:51:CRYPTO_ENGINE:Dh phase 1 status:0
00:27:51:CryptoEngine0:generate alg parameter
00:27:52:CryptoEngine0:calculate pkey hmac for conn id 0
00:27:52:CryptoEngine0:create ISAKMP SKEYID for conn id 1
00:27:52:Crypto engine 0:RSA decrypt with public key
00:27:52:CryptoEngine0:CRYPTO_RSA_PUB_DECRYPT
00:27:52:CryptoEngine0:generate hmac context for conn id 1
00:27:52:CryptoEngine0:generate hmac context for conn id 1
00:27:52:Crypto engine 0:RSA encrypt with private key
00:27:52:CryptoEngine0:CRYPTO_RSA_PRIV_ENCRYPT
00:27:53:CryptoEngine0:clear dh number for conn id 1
00:27:53:CryptoEngine0:generate hmac context for conn id 1
00:27:53:validate proposal 0
00:27:53:validate proposal request 0
00:27:54:CryptoEngine0:generate hmac context for conn id 1
00:27:54:CryptoEngine0:generate hmac context for conn id 1
```

```
00:27:54:ipsec allocate flow 0
00:27:54:ipsec allocate flow 0
```

debug crypto isakmp

Use this command to view to see the Internet Security Association and Key Management Protocol (ISAKMP) phase 1 negotiations. See http://www.cisco.com/en/US/tech/tk648/tk367/technologies_tech_note09186a00800949c5.shtml#crypto_isakmp.

The following is sample output from the **debug crypto isakmp** command for an IKE peer that initiates an IKE negotiation. First, IKE negotiates its own security association (SA), checking for a matching IKE policy:

```
Router# debug crypto isakmp
20:26:58: ISAKMP (8): beginning Main Mode exchange
20:26:58: ISAKMP (8): processing SA payload. message ID = 0
20:26:58: ISAKMP (8): Checking ISAKMP transform 1 against priority 10 policy
20:26:58: ISAKMP: encryption DES-CBC
20:26:58: ISAKMP: hash SHA
20:26:58: ISAKMP: default group 1
20:26:58: ISAKMP: auth pre-share
20:26:58: ISAKMP (8): atts are acceptable. Next payload is 0
```

IKE has found a matching policy. Next, the IKE SA is used by each peer to authenticate the other peer:

```
20:26:58: ISAKMP (8): SA is doing pre-shared key authentication
20:26:59: ISAKMP (8): processing KE payload. message ID = 0
20:26:59: ISAKMP (8): processing NONCE payload. message ID = 0
20:26:59: ISAKMP (8): SKEYID state generated
20:26:59: ISAKMP (8): processing ID payload. message ID = 0
20:26:59: ISAKMP (8): processing HASH payload. message ID = 0
20:26:59: ISAKMP (8): SA has been authenticated
```

Next, IKE negotiates to set up the IPSec SA by searching for a matching transform set:

```
20:26:59: ISAKMP (8): beginning Quick Mode exchange, M-ID of 767162845
20:26:59: ISAKMP (8): processing SA payload. message ID = 767162845
20:26:59: ISAKMP (8): Checking IPSec proposal 1
20:26:59: ISAKMP: transform 1, ESP_DES
20:26:59: ISAKMP: attributes in transform:
20:26:59: ISAKMP: encaps is 1
20:26:59: ISAKMP: SA life type in seconds
20:26:59: ISAKMP: SA life duration (basic) of 600
20:26:59: ISAKMP: SA life type in kilobytes
20:26:59: ISAKMP: SA life duration (VPI) of
0x0 0x46 0x50 0x0
20:26:59: ISAKMP: authenticator is HMAC-MD5
20:26:59: ISAKMP (8): atts are acceptable.
```

A matching IPSec transform set has been found at the two peers. Now the IPSec SA can be created (one SA is created for each direction):

```
20:26:59: ISAKMP (8): processing NONCE payload. message ID = 767162845
20:26:59: ISAKMP (8): processing ID payload. message ID = 767162845
20:26:59: ISAKMP (8): processing ID payload. message ID = 767162845
20:26:59: ISAKMP (8): Creating IPSec SAs
20:26:59: inbound SA from 155.0.0.2 to 155.0.0.1 (proxy 155.0.0.2 to 155.0.0.1)
20:26:59: has spi 454886490 and conn_id 9 and flags 4
20:26:59: lifetime of 600 seconds
20:26:59: lifetime of 4608000 kilobytes
20:26:59: outbound SA from 155.0.0.1 to 155.0.0.2 (proxy 155.0.0.1
to 155.0.0.2 )
```

```
20:26:59: has spi 75506225 and conn_id 10 and flags 4
20:26:59: lifetime of 600 seconds
20:26:59: lifetime of 4608000 kilobytes
```

debug crypto ipsec

Use this command to view the IPSec phase 2 negotiations. For a description and an example, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122debug/dbfclns.htm#xtocid16.

The following is sample output from the **debug crypto ipsec** command. In this example, security associations (SAs) have been successfully established.

```
Router# debug crypto ipsec
```

IPSec requests SAs between 172.21.114.123 and 172.21.114.67, on behalf of the **permit ip host 172.21.114.123 host 172.21.114.67** command. It prefers to use the transform set esp-des w/esp-md5-hmac, but it also considers ah-sha-hmac.

```
00:24:30: IPSEC(sa_request): ,
(key eng. msg.) src= 172.21.114.123, dest= 172.21.114.67,
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
dest_proxy= 172.21.114.67/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 120s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
00:24:30: IPSEC(sa_request): ,
(key eng. msg.) src= 172.21.114.123, dest= 172.21.114.67,
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
dest_proxy= 172.21.114.67/255.255.255.255/0/0 (type=1).,
protocol= AH, transform= ah-sha-hmac ,
lifedur= 120s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x0.
```

IKE asks for SPIs from IPSec. For inbound security associations, IPSec controls its own SPI space.

```
00:24:34: IPSEC(key_engine): got a queue event...
00:24:34: IPSEC(spi_response): getting spi 3029740121d for SA
from 172.21.114.67 to 172.21.114.123 for prot 3
00:24:34: IPSEC(spi_response): getting spi 5250759401d for SA
from 172.21.114.67 to 172.21.114.123 for prot 2
```

IKE asks IPSec if it accepts the SA proposal. In this case, it will be the one sent by the local IPSec:

```
00:24:34: IPSEC(validate_proposal_request): proposal part #1,
(key eng. msg.) dest= 172.21.114.67, src= 172.21.114.123,
dest_proxy= 172.21.114.67/255.255.255.255/0/0 (type=1),
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
```

After the proposal is accepted, IKE finishes the negotiations, generates the keying material, and then notifies IPSec of the new security associations (one security association for each direction).

```
00:24:35: IPSEC(key_engine): got a queue event...
```

The following output pertains to the inbound SA. The conn_id value references an entry in the crypto engine connection table.

```
00:24:35: IPSEC(initialize_sas): ,
(key eng. msg.) dest= 172.21.114.123, src= 172.21.114.67,
dest_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
```

```
src_proxy= 172.21.114.67/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 120s and 4608000 kb,
spi= 0x120F043C(302974012), conn_id= 29, keysize= 0, flags= 0x4
```

The following output pertains to the outbound SA:

```
00:24:35: IPSEC(initialize_sas): ,
(key eng. msg.) src= 172.21.114.123, dest= 172.21.114.67,
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
dest_proxy= 172.21.114.67/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 120s and 4608000kb,
spi= 0x38914A4(59315364), conn_id= 30, keysize= 0, flags= 0x4
```

IPSec now installs the SA information into its SA database.

```
00:24:35: IPSEC(create_sa): sa created,
(sa) sa_dest= 172.21.114.123, sa_prot= 50,
sa_spi= 0x120F043C(302974012),
sa_trans= esp-des esp-md5-hmac , sa_conn_id= 29
00:24:35: IPSEC(create_sa): sa created,
(sa) sa_dest= 172.21.114.67, sa_prot= 50,
sa_spi= 0x38914A4(59315364),
sa_trans= esp-des esp-md5-hmac , sa_conn_id= 30
```

The following is sample output for the **debug crypto ipsec** command as seen on the peer router. In this example, IKE asks IPSec to accept an SA proposal. Although the peer sent two proposals, IPSec accepts the first proposal.

```
00:26:15: IPSEC(validate_proposal_request): proposal part #1,
(key eng. msg.) dest= 172.21.114.67, src= 172.21.114.123,
dest_proxy= 172.21.114.67/255.255.255.255/0/0 (type=1),
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
```

IKE asks for SPIs.

```
00:26:15: IPSEC(key_engine): got a queue event...
00:26:15: IPSEC(spi_response): getting spi 59315364ld for SA
from 172.21.114.123 to 172.21.114.67 for prot 3
```

IKE performs the remaining processing, completing the negotiation and generating keys. It then tells IPSec about the new SAs.

```
00:26:15: IPSEC(key_engine): got a queue event...
```

The following output pertains to the inbound SA:

```
00:26:15: IPSEC(initialize_sas): ,
(key eng. msg.) dest= 172.21.114.67, src= 172.21.114.123,
dest_proxy= 172.21.114.67/0.0.0.0/0/0 (type=1),
src_proxy= 172.21.114.123/0.0.0.0/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 120s and 4608000kb,
spi= 0x38914A4(59315364), conn_id= 25, keysize= 0, flags= 0x4
```

The following output pertains to the outbound SA:

```
00:26:15: IPSEC(initialize_sas): ,
(key eng. msg.) src= 172.21.114.67, dest= 172.21.114.123,
src_proxy= 172.21.114.67/0.0.0.0/0/0 (type=1),
dest_proxy= 172.21.114.123/0.0.0.0/0/0 (type=1),
```

```
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 120s and 4608000kb,
spi= 0x120F043C(302974012), conn_id= 26, keysize= 0, flags= 0x4
```

IPSec installs the SA information into its SA database:

```
00:26:15: IPSEC(create_sa): sa created,
(sa) sa_dest= 172.21.114.67, sa_prot= 50,
sa_spi= 0x38914A4(59315364),
sa_trans= esp-des esp-md5-hmac , sa_conn_id= 25
00:26:15: IPSEC(create_sa): sa created,
(sa) sa_dest= 172.21.114.123, sa_prot= 50,
sa_spi= 0x120F043C(302974012),
sa_trans= esp-des esp-md5-hmac , sa_conn_id= 26
```

show crypto ipsec sa

Use this command to view the IPSec SAs built between peers. See http://www.cisco.com/en/US/tech/tk648/tk367/technologies_tech_note09186a00800949c5.shtml#ipsec_sa.

The encrypted tunnel is built between 12.1.1.1 and 12.1.1.2 for traffic traveling between networks 20.1.1.0 and 10.1.1.0. You can view the two Encapsulating Security Payload (ESP) SAs built inbound and outbound. Authentication Header (AH) is not used because there are no AH SAs. Below is an example of the **show crypto ipsec sa** command.

```
interface: FastEthernet0
Crypto map tag: test, local addr. 12.1.1.1
local ident (addr/mask/prot/port): (20.1.1.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.1.1.0/255.255.255.0/0/0)
current_peer: 12.1.1.2
PERMIT, flags={origin_is_acl,}
#pkts encaps: 7767918, #pkts encrypt: 7767918, #pkts digest 7767918
#pkts decaps: 7760382, #pkts decrypt: 7760382, #pkts verify 7760382
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
#send errors 1, #recv errors 0
local crypto endpt.: 12.1.1.1, remote crypto endpt.: 12.1.1.2
path mtu 1500, media mtu 1500
current outbound spi: 3D3
inbound esp sas:
spi: 0x136A010F(325714191)
transform: esp-3des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 3442, flow_id: 1443, crypto map: test
sa timing: remaining key lifetime (k/sec): (4608000/52)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound pcp sas:
inbound pcp sas:
outbound esp sas:
spi: 0x3D3(979)
transform: esp-3des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 3443, flow_id: 1444, crypto map: test
sa timing: remaining key lifetime (k/sec): (4608000/52)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound pcp sas:
```

show crypto engine connection active

Use this command to view each phase 2 SA built and the amount of traffic sent. See http://www.cisco.com/en/US/tech/tk648/tk367/technologies_tech_note09186a00800949c5.shtml#crypto_engine.

Note Remember that phase 2 SAs are unidirectional, so each SA shows traffic in one direction only (encryptions are outbound, decryptions are inbound).

IPSec Troubleshooting Strategy

This section provides a general IPSec troubleshooting path.

1. Remove crypto maps from the interfaces and check basic IP connectivity.
2. Compare configuration on both sides for symmetry.
3. Apply maps back on, turn on the following debugs:
   a. **debug crypto ipsec**
   b. **debug crypto isakmp**
   c. **debug crypto engine**
4. Generate interesting traffic (often an extended ping is necessary)
5. Observe debugs for error messages and exceptions
6. Check to see if the match address access list is receiving hits
7. Use the following show commands:
   a. **show crypto isakmp sa**
   b. **show crypto ipsec sa**
   c. **sh crypto engine connections active**
8. Use the following commands to clear established sessions and regenerate debugs:
   a. **clear crypto sa**
   b. **clear crypto isakmp**
9. Use the following show commands:
   a. **show crypto isakmp sa**
   b. **show crypto ipsec sa**
   c. **sh crypto engine connections active**
10. Use the following commands to clear established sessions and regenerate debugs:
    a. **clear crypto sa**
    b. **clear crypto isakmp**

Hardware Accelerator Card Problems

Determine if the router is using a hardware accelerator (encryption) card by issuing the following command:

show crypto engine config

If the hardware card is not being used, the 'crypto engine type' is reported as 'software,' otherwise, it reports the actual card type in use, such as ISA/ISM.

If the hardware card is being used, it might sometimes lapse into a limbo state and cause problems with establishment of tunnels. You can disable the card to determine if this is the case by issuing the following command:

no crypto engine accelerator <slot_#>

You can then recheck for the tunnels. If the tunnels resume operation successfully, the problem resides with the accelerator card.

You can put the card back in service by issuing the following command:

crypto engine accelerator <slot_#>

If you continue to experience problems with the card, you can try reloading the router. If the problem persists, try swapping out the card for a new one.

Note Each time you disable or enable the accelerator card, all existing crypto connections are torn down.

The following commands can be used to obtain statistics from the hardware card for the 7000 platforms and other platforms, respectively:

show pas isa interface

sho crypto engine accelerator stats

Questions for Effective Troubleshooting

IPSec connectivity fails if problems occur during any of the stages listed above. To effectively troubleshoot connectivity problems, consider the key questions in the following sections.

- Is the router receiving and recognizing interesting traffic?, page 3-9
- Does the router successfully negotiate an IKE tunnel with its peer?, page 3-10
- Does the router successfully negotiate IPSec tunnels with its peer?, page 3-11

Is the router receiving and recognizing interesting traffic?

Step 1 To check for interesting traffic, issue the following command:

s**how access-list <access-list-name>**

The number of matches for the access list should increment for each interesting data packet; if the number of matches is increasing, you can skip the following step.

Step 2 If the number of matches is not increasing, check to make sure that the source interface for the traffic is operational by using the following command:

show interface <interface name>

If the interface is operational, check the 'match address' section of your crypto map to make sure that the access list specified there indicates the correct source and destination addresses / networks.

Step 3 Next, check to make sure that the interesting traffic is being routed to the correct interface by issuing the following command:

show ip routes

If the interesting traffic is not being routed to the correct interface, configure the required route entries.

Step 4 Finally, verify that the crypto map is being applied to the outbound interface by issuing the following command:

show run interface <interface_name>

Does the router successfully negotiate an IKE tunnel with its peer?

When the router sees interesting traffic, it attempts to negotiate an IKE tunnel with the remote peer specified in the 'set peer' statement of the crypto map.

Step 1 To verify if IKE is active, issue the following command:

show crypto isakmp sa

If the IKE is active, output similar to the following appears:

```
dst         src         state      conn-id slot
192.1.66.1 192.168.1.1 QM_IDLE    468     0
```

If the IKE is active, you can skip Step 2.

If output similar to either of the following examples appears, the peers are still attempting to negotiate IKE:

```
dst         src         state          conn-id     slot
192.1.66.1 192.168.1.1 MM_KEY_EXCH    320         0

dst src     state      conn-id slot
192.1.66.1 192.168.1.1 MM_SA_SETUP 36 0
```

If output similar to the following example appears, the IKE is not established:

```
dst         src         state          conn-id     slot
192.1.66.1 192.168.1.1 MM_NO_STATE    36          0
```

Step 2 If the IKE does not become active, ping to make sure that the remote peer (tunnel endpoint) is reachable. (Remember that the tunnel endpoints **must** be routable addresses). If you are unable to ping the remote peer, check your routing tables by issuing the following command:

show ip routes

Step 3 Configure any necessary routes. Check the status of all intervening interfaces along the routing path, and make sure that they are all operational.

Step 4 If the remote peer is reachable, check to make sure that the ISAKMP policies are the same on both peers. Issue the following command on both peers:

show crypto isakmp policy

Step 5 The encryption algorithm, hash algorithm, authentication method, ISAKMP lifetimes and Diffie-Helman group must match each other. Correct the configuration if you see discrepancies. If preshared keys are being used, such as the authentication method, the keys must be the same on both peers. Check this by issuing the following command:

show crypto isakmp key

You should then see output similar to the following:

```
7200-Gen#show crypto isakmp key
Hostname/Address Preshared key
192.168.1.1 cisco
192.16.7.1 beehive
```

Step 6 Compare the ISAKMP policies manually on the two peers to identify discrepancies. An alternate method of identifying problems is referencing error messages that are logged to the syslog by such discrepancies. (It is a good idea to keep the syslog turned on).

Step 7 Another important safeguard is checking to make sure that the ISAKMP packets are not blocked. To implement this procedure, simply run debugs on both peers and check to see if there is any activity.

Does the router successfully negotiate IPSec tunnels with its peer?

Step 1 When the IKE is operational, IPSec tunnels (one for inbound and one for outbound traffic) are negotiated. To check, issue the following command:

show crypto engine connection active

If the IPSec tunnels are active, you should see output similar to the following:

```
UUT#sho cry eng conn active
ID Interface IP-Address State Algorithm Encrypt Decrypt
1 <none> <none> set HMAC_SHA+3DES_56_C 0 0
2029 Serial5/0 192.168.1.1 set HMAC_SHA+3DES_56_C 0 1987
2030 Serial5/0 192.168.1.1 set HMAC_SHA+3DES_56_C 2378 0
UUT#
```

Step 2 The numbers in the ‘encrypted’ and ‘decrypted’ columns must increment continuously depending upon whether the router is encrypting packets, decrypting packets, or both. If this is not happening, check to verify that there is at least one matching crypto IPSec transform on the two peers by issuing the following command:

sho crypto ipsec transform

Step 3 There must be at least one transform common to both peers. The first transform that matches (not necessarily the strongest common transform) is used to build the IPSec tunnels. If there is no transform common to both peers, IPSec tunnels do not come up. Be sure to configure transforms correctly.

Step 4 You can also display detailed information and statistics for the IPSec tunnels by issuing the following command:

show crypto ipsec sa

The information provided by this command might be useful for more in-depth troubleshooting. The output from this command will be similar to the following example:

```
7200-Gen#show crypto ipsec sa
interface: ATM5/0.1
Crypto map tag: crypmap1, local addr. 192.1.1.1
local ident (addr/mask/prot/port): (10.1.1.1/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (100.1.1.1/255.255.255.255/0/0)
```

```
current_peer: 192.168.1.1
PERMIT, flags={origin_is_acl,}
#pkts encaps: 3010, #pkts encrypt: 3010, #pkts digest 3010
#pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
#send errors 3, #recv errors 0
local crypto endpt.: 192.1.1.1, remote crypto endpt.: 192.168.1.1
path mtu 4470, media mtu 4470
current outbound spi: 5ECB6495
inbound esp sas:
spi: 0xD90CEC8(227593928)
transform: esp-3des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 3739, flow_id: 1711, crypto map: crypmap1
sa timing: remaining key lifetime (k/sec): (4608000/590)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound pcp sas:
outbound esp sas:
spi: 0x5ECB6495(1590387861)
transform: esp-3des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 3740, flow_id: 1712, crypto map: crypmap1
sa timing: remaining key lifetime (k/sec): (4607415/590)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound pcp sas:
```

Sample IPSec Debug

The following is a sample IPSec debug:

```
irmg.cayman#sh debug
Cryptographic Subsystem:
  Crypto ISAKMP debugging is on
  Crypto Engine debugging is on
  Crypto IPSEC debugging is onirmg.cayman#ping
Protocol [ip]:
Target IP address: 10.146.1.1
Repeat count [5]: 10
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 205.136.238.33
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 10.146.1.1, timeout is 2 seconds:

1d09h: IPSEC(sa_request): ,
(key eng. msg.) src= 205.136.238.33, dest= 209.146.47.206,
    src_proxy= 205.136.238.0/255.255.255.0/0/0 (type=4),
    dest_proxy= 10.146.1.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des ,
    lifedur= 3600s and 4608000kb,
```

```
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
1d09h: ISAKMP (16): beginning Main Mode exchange1d09h: ISAKMP (16): processing SA payload.
message ID = 0
1d09h: ISAKMP (16): Checking ISAKMP transform 1 against priority 1 policy
1d09h: ISAKMP:      encryption DES-CBC
1d09h: ISAKMP:      hash MD5
1d09h: ISAKMP:      default group 1
1d09h: ISAKMP:      auth pre-share.
1d09h: ISAKMP (16): atts are acceptable. Next payload is 0
1d09h: Crypto engine 0: generate alg param

1d09h: CRYPTO_ENGINE: Dh phase 1 status: 0
1d09h: ISAKMP (16): SA is doing pre-shared key authentication
1d09h: ISAKMP (16): processing KE payload. message ID = 0
1d09h: Crypto engine 0: generate alg param

1d09h: ISAKMP (16): processing NONCE payload. message ID = 0
1d09h: Crypto engine 0: create ISAKMP SKEYID for conn id 16
1d09h: ISAKMP (16): SKEYID state generated
1d09h: ISAKMP (16): processing vendor id payload
1d09h: ISAKMP (16): speaking to another IOS box!
1d09h: generate hmac context for conn id 16
1d09h: ISAKMP (16): processing ID payload. message ID = 0
1d09h: ISAKMP (16): processing HASH payload. message ID = 0
1d09h: generate hmac context for conn id 16
1d09h: ISAKMP (16): SA has been authenticated with 209.146.47.206
1d09h: ISAKMP (16): beginning Quick Mode exchange, M-ID of 102008272
1d09h: IPSEC(key_engine): got a queue event...
1d09h: IPSEC(spi_response): getting spi 207749755ld for SA
        from 209.146.47.206  to 205.136.238.33  for prot 3
1d09h: generate hmac context for conn id 16
1d09h: generate hmac context for conn id 16
1d09h: ISAKMP (16): processing SA payload. message ID = 102008272
1d09h: ISAKMP (16): Checking IPSec proposal 1
1d09h: ISAKMP: transform 1, ESP_DES
1d09h: ISAKMP:   attributes in transform:
1d09h: ISAKMP:      encaps is 1
1d09h: ISAKMP:      SA life type in seconds
1d09h: ISAKMP:      SA life duration (basic) of 3600
1d09h: ISAKMP:      SA life type in kilobytes
1d09h: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
1d09h: ISAKMP (16): atts are acceptable.
1d09h: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) dest= 209.146.47.206, src= 205.136.238.33,
    dest_proxy= 10.146.1.0/255.255.255.0/0/0 (type=4),
    src_proxy= 205.136.238.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
1d09h: ISAKMP (16): processing NONCE payload. message ID = 102008272
1d09h: ISAKMP (16): processing ID payload. message ID = 102008272
1d09h: ISAKMP (16): processing ID payload. message ID = 102008272
1d09h: generate hmac context for conn id 16
1d09h: ISAKMP (16): Creating IPSec SAs
1d09h:         inbound SA from 209.146.47.206  to 205.136.238.33  (proxy 10.146.
1.0      to 205.136.238.0  )
1d09h:         has spi 207749755 and conn_id 17 and flags 4
1d09h:         lifetime of 3600 seconds
1d09h:         lifetime of 4608000 kilobytes
1d09h:         outbound SA from 205.136.238.33  to 209.146.47.206  (proxy 205.13
6.238.0   to 10.146.1.0     )
1d09h:         has spi 440535111 and conn_id 18 and flags 4
1d09h:         lifetime of 3600 seconds
1d09h:         lifetime of 4608000 kilobytes
```

```
1d09h: IPSEC(key_engine): got a queue event...
1d09h: IPSEC(initialize_sas): ,
  (key eng. msg.) dest= 205.136.238.33, src= 209.146.47.206,
dest_proxy= 205.136.238.0/255.255.255.0/0/0 (type=4),
    src_proxy= 10.146.1.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des ,
    lifedur= 3600s and 4608000kb,
    spi= 0xC62027B(207749755), conn_id= 17, keysize= 0, flags= 0x4
1d09h: IPSEC(initialize_sas): ,
  (key eng. msg.) src= 205.136.238.33, dest= 209.146.47.206,
    src_proxy= 205.136.238.0/255.255.255.0/0/0 (type=4),
    dest_proxy= 10.146.1.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, .transform= esp-des ,
    lifedur= 3600s and 4608000kb,
    spi= 0x1A420847(440535111), conn_id= 18, keysize= 0, flags= 0x4
1d09h: IPSEC(create_sa): sa created,
  (sa) sa_dest= 205.136.238.33, sa_prot= 50,
    sa_spi= 0xC62027B(207749755),
    sa_trans= esp-des , sa_conn_id= 17
1d09h: IPSEC(create_sa): sa created,  (sa) sa_dest= 209.146.47.206, sa_prot= 50,
    sa_spi= 0x1A420847(440535111),
    sa_trans= esp-des , sa_conn_id= 18!!!!!!!!
Success rate is 80 percent (8/10), round-trip
min/avg/max = 180/185/205 ms
irmg.cayman#sh crypto engine conn ac

ID Interface IP-Address State Algorithm Encrypt decrypt
16 no idb    no address set   DES_56_CBC  0        0
17 Se0/0.1 205.136.238.33  set DES_56_CBC 0     8
18 Se0/0.1 205.136.238.33  set DES_56_CBC 8     0
irmg.cayman#sh crypto isa sa
    dst           src          state        conn-id   slot
209.146.47.206 205.136.238.33 QM_IDLE           16      0
irmg.cayman#sh crypto ipsec sa

interface: Serial0/0.1
    Crypto map tag: charlie2peer, local addr. 205.136.238.33

   local  ident (addr/mask/prot/port): (205.136.238.0/255.255.255.0/0/0)
  remote ident (addr/mask/prot/port): (10.146.1.0/255.255.255.0/0/0)
   current_peer: 209.146.47.206
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 8, #pkts encrypt: 8, #pkts digest 0
    #pkts decaps: 8, #pkts decrypt: 8, #pkts verify 0
    #send errors 2, #recv errors 0

     local crypto endpt.: 205.136.238.33, remote crypto endpt.: 209.146.47.206
     path mtu 1500, media mtu 1500
     current outbound spi: 1A420847

     inbound esp sas:
      spi: 0xC62027B(207749755)
        transform: esp-des ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 17, crypto map: charlie2peer
        sa timing: remaining key lifetime (k/sec): (4607998/3455)
        IV size: 8 bytes
        replay detection support: N
inbound ah sas:
    outbound esp sas:
      spi: 0x1A420847(440535111)
        transform: esp-des ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 18, crypto map: charlie2peer
```

```
        sa timing: remaining key lifetime (k/sec): (4607999/3446)
        IV size: 8 bytes
        replay detection support: N

     outbound ah sas:

irmg.cayman#sh ru
Building configuration...

Current configuration:
hostname irmg.cayman
crypto isakmp policy 1
 hash md5
 authentication pre-share
crypto isakmp key orientexpress address 209.146.47.206
crypto ipsec transform-set encrypt-des esp-des
crypto map charlie2peer local-address Ethernet0/0
 crypto map charlie2peer 10 ipsec-isakmp
 set peer 209.146.47.206
 set transform-set encrypt-des
 match address 110
interface Ethernet0/0
 description connected to EthernetLAN
 ip address 205.136.238.33 255.255.255.224
 no ip directed-broadcast
 no keepalive
interface Serial0/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip route-cache
 no ip mroute-cache
 frame-relay lmi-type ansi
!
interface Serial0/0.1 point-to-point
 description connected to Internet
 ip unnumbered Ethernet0/0
 no ip directed-broadcast
 no ip route-cache
no ip mroute-cache
 bandwidth 64
 frame-relay interface-dlci 16 IETF
 crypto map charlie2peer
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0.1
access-list 110 permit ip 205.136.238.0 0.0.0.255 10.146.1.0 0.0.0.255
```

Common IPSec Error Messages

Table 3-2 shows some common IPSec error messages, reasons for them, as well as possible solutions.

Table 3-2 Common IPSec Error Messages

Message	Reason	Solution
`invalid local address 1.2.3.4 no IPSEC cryptomap exists for local address 1.2.3.4`	The specified address is not one of our local IPSec addresses.	Check to see if the outgoing interfaces address has been specified on the peer as peer id. Check peer id address for error. Use the **crypto map local address** command.
`proxy identities not supported`	The flow identities do not match a crypto map ACL permit statement.	The address being pinged is not included in the match address access-list. Change the access-list. Use reflected access-lists on the two ends of the tunnel.

`atts are not acceptable. Next payload is 0`
`no offers accepted!`
`SA not acceptable!`
`%CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main Mode failed with peer at 155.0.0.1`
	The polices and configurations on the two peers do not match.	Check to make sure that the policies configured on the two ends of the tunnel are correct and correlate.

`reserved no zero on payload 5!` `%CRYPTO-4-IKMP_BAD_MESSAGE: IKE message from 155.0.0.1 failed its sanity check or is malformed`	The encryption keys on the two ends do not match.	Check to make sure that the preshared keys are correctly configured. Regenerate RSA encrypted keys.
`Rec'd packet not an IPSEC packet. (ip) dest_addr= 1.2.3.4, src_addr= 2.3.4.5, prot= 1`	An incoming packet matched the match address access list but was not encrypted. Due to security concerns such packets are dropped.	Check to make sure that the routing between the two routers is correct (that is, packets are being sent between the interfaces doing encryption). Verfiy NAT is not occurring for IPSEC traffic. Do not use match address access list with any any. Turn off fast switching. Some Cisco IOS software versions issue a bug if partial subnets are configured in the match address access list.
`ISAKMP (0:1): SA is still budding. Attached new ipsec request to it.`	Indicates IPSec has sent another request to IKE to create SAs on its behalf; however, IKE is in the process of fulfilling a similar request. Thus, IKE attaches the new request to the SA-negotiation already in progress.	This is an informational message and does not necessarily signify an error.

Table 3-2 Common IPSec Error Messages (continued)

Message	Reason	Solution
`ISAKMP (0:1): deleting node-45086069 error FALSE [reason]`	Indicates that a memory structure that is not needed has been cleared during normal housekeeping. The error False portion of this message indicates that an error has not occurred; an error True portion of this message would indicate that a memory structure that is not needed has been cleared, but an error has occurred.	—
`%CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSec packet has invalid spi for destaddr=16.0.0.3, prot=50, spi=0xdeadbeef.`	Indicates that the IPSec SAs between router A and router B are out of sync. This can occur if router A has cleared its IPSec SAs but router B has not. Thus, when router B attempts to send encrypted traffic corresponding to the SAs, router A drops the packet and reports this message.	This is an informational message and does not necessarily signify an error.
`ISAKMP (0:2): deleting SA reason "P1 delete notify (in)" state (I) CONF_ADDR (peer 10.13.1.21) input queue 0`	Indicates that the IKE SA with the connection ID 2 has been deleted. *P1 delete notify* indicates that the router received a phase 1 delete notification from its peer. The (I) (in 'state') means that the router was the initiator for this SA; an (R) would mean that the responder was the initiator. This message also indicates the peer IP address.	—
`ISAKMP (0:2): Notify has no hash. Rejected`	Indicates that the notify message received from the peer lacked a valid hash. This means that the notify message was not authenticated. For security reasons, this message is ignored.	—

Table 3-2 Common IPSec Error Messages (continued)

Message	Reason	Solution

`IPSEC(validate_proposal): invalid local address 12.2.6.2`
`ISAKMP (0:3): atts not acceptable. Next payload is 0`
`ISAKMP (0:3): SA not acceptable!`
	This error message is attributed to one of the following two common problems:
	• The **crypto map map-name local-address interface-id** command causes the router to use an incorrect address as the identity because it forces the router to use a specified address.
	• Crypto map is applied to the incorrect interface or, it is not applied at all.
		Check the configuration to ensure that crypto map is applied to the correct interface.

`1d00H:%CRPTO-4-IKMP_BAD_MESSAGE: IKE message from 150.150.150.1 failed its sanity check or is malformed`	The preshared key on the peers do not match.	Check the preshared key on both sides.

`1d00h: ISAKMP (0:1): atts are not acceptable. Next payload is 0`
`1d00h: ISAKMP (0:1); no offers accepted!`
`1d00h: ISAKMP (0:1): SA not acceptable!`
`1d00h: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main Mode failed with peer at 150.150.150.1`
	This is an example of the Main Mode error message. The failure of Main Mode suggests that the phase I policy does not match on both sides.	Verify that the phase I policy is on both peers and ensure that all the attributes match, for example:
		`Encryption DES or 3DES`
		`Hash MD5 or SHA`
		`Diffie-Hellman Group 1 or 2`
		`Authentcation {rsa-sig | rsa-encr | pre-share`

`1d00h:`
`IPSec(validate_transform_proposal): proxy identities not supported`
`1d00h: ISAKMP: IPSec policy invalidated proposal`
`1d00h: ISAKMP (0:2): SA not acceptable!`
	The message appears in debugs if the access-list for IPSec traffic does not match.	The access-list on each peer should mirror each other (all entries should be reversible). See below:
		`Peer A`
		`access-list 150 permit ip 172.21.113.0 0.0.0.255 172.21.114.0 0.0.0.255`
		`access-list 150 permit ip host 15.15.15.1 host 172.21.114.123`
		`Peer B`
		`access-list 150 permit ip 172.21.114.0 0.0.0.255 172.21.113.0 0.0.0.255`
		`access-list 150 permit ip host 172.21.114.123 host 15.15.15.1`

Table 3-2 Common IPSec Error Messages (continued)

Message	Reason	Solution

`1d00h: IPSec (validate_proposal): transform proposal (port 3, trans 2, hmac_alg 2) not supported`
`1d00h: ISAKMP (0:2) : atts not acceptable. Next payload is 0`
`1d00h: ISAKMP (0:2) SA not acceptable`
	Indicates the Phase II (IPSec) does not match on both sides.	Occurs if there is a mismatch in the transform-set. Verify that the transform-set matches on both sides:
		`crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]`
		`? ah-md5-hmac`
		`? ah-sha-hmac`
		`? esp-des`
		`? esp-des and esp-md5-hmac`
		`? esp-des and esp-sha-hmac`
		`? esp-3des and esp-md5-hmac`
		`? esp-3des and esp-sha-hmac`
		`? comp-lzs`

`1d00h: ISAKMP: No cert, and no keys (public or pre-shared) with remote peer 150.150.150.2`	Indicates that the peer address configured on the router is incorrect or has changed.	Verify that the peer address is correct and reachable.
`20:44:44: IPSEC(validate_proposal_request): proposal part #1, (key eng. msg.) dest= 194.70.240.150, src= 198.174.236.6, dest_proxy= 10.0.0.76/255.255.255.255/0/0 (type=1), src_proxy= 198.174.238.203/255.255.255.255/0/0 (type=1), protocol= ESP, transform= esp-3des esp-md5-hmac , lifedur= 0s and 0kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4` `20:44:44: IPSEC(validate_transform_proposal): peer address 198.174.236.6 not found`	Normally appears with the corresponding Cisco VPN 3000 concentrator message: No proposal chosen(14). This is a result of the connections being host-to-host. The router configuration had the IPSec proposals ordered so that the proposal chosen for the router matched the access-list, but not the peer. The access-list had a larger network that included the host that was intersecting traffic.	Make the router proposal for this concentrator-to-router connection first in line, so that it matches the specific host first.
`21:57:57: IPSEC(validate_proposal_request): proposal part #1, (key eng. msg.) dest= 192.1.1.1, src= 192.1.1.2, dest_proxy= 10.1.1.1/255.255.255.0/0/0 (type=4), src_proxy= 20.1.1.1/255.255.255.0/0/0 (type=4)`	The debug command output of this proposal request shows the corresponding access-list 103 permit ip 10.1.1.0 0.0.0.255 20.1.1.0 0.0.0.255 does not match. The access-list is network specific on one end and host specific on the other.	Match access lists.
`21:57:57: IPSEC(initialize_sas): invalid proxy IDs`	Indicates that the received proxy identity does not match the configured proxy identity as per the access-list.	Check to ensure that the received proxy identity and the configured proxy identity both match by checking the output from the **debug** command.

Common IPSec Issues

This section shows samples of the following common IPSec issues:

Mismatch in Phase 1 Policy Parameters

```
*Nov 17 16:52:03.373: ISAKMP (0:3): Checking ISAKMP transform 1 against priority 1 policy
*Nov 17 16:52:03.373: ISAKMP: encryption 3DES-CBC
*Nov 17 16:52:03.373: ISAKMP: hash MD5
*Nov 17 16:52:03.373: ISAKMP: default group 1
*Nov 17 16:52:03.373: ISAKMP: auth pre-share
*Nov 17 16:52:03.373: ISAKMP: life type in seconds
*Nov 17 16:52:03.373: ISAKMP: life duration (VPI) of 0x0 0x1 0x51 0x80
*Nov 17 16:52:03.373: ISAKMP (0:3): Hash algorithm offered does not match policy!
*Nov 17 16:52:03.373: ISAKMP (0:3): atts are not acceptable. Next payload is 0
*Nov 17 16:52:03.373: ISAKMP (0:3): Checking ISAKMP transform 1 against priority 65535
policy
*Nov 17 16:52:03.373: ISAKMP: encryption 3DES-CBC
*Nov 17 16:52:03.373: ISAKMP: hash MD5
*Nov 17 16:52:03.373: ISAKMP: default group 1
*Nov 17 16:52:03.373: ISAKMP: auth pre-share
*Nov 17 16:52:03.373: ISAKMP: life type in seconds
*Nov 17 16:52:03.373: ISAKMP: life duration (VPI) of 0x0 0x1 0x51 0x80
*Nov 17 16:52:03.373: ISAKMP (0:3): Encryption algorithm offered does not match policy!
*Nov 17 16:52:03.373: ISAKMP (0:3): atts are not acceptable. Next payload is 0
*Nov 17 16:52:03.373: ISAKMP (0:3): no offers accepted!
*Nov 17 16:52:03.373: ISAKMP (0:3): phase 1 SA not acceptable!
```

No Preshared Key for the Peer

```
*Nov 19 13:12:00.562: ISAKMP (0:3): processing SA payload. message ID = 0
*Nov 19 13:12:00.562: ISAKMP (0:3): No pre-shared key with 20.1.1.1!
*Nov 19 13:12:00.562: ISAKMP (0:3): Checking ISAKMP transform 1 against priority 1 policy
*Nov 19 13:12:00.562: ISAKMP: encryption 3DES-CBC
*Nov 19 13:12:00.562: ISAKMP: hash SHA
```

```
*Nov 19 13:12:00.562: ISAKMP: default group 1
*Nov 19 13:12:00.562: ISAKMP: auth pre-share
*Nov 19 13:12:00.562: ISAKMP: life type in seconds
*Nov 19 13:12:00.562: ISAKMP: life duration (VPI) of 0x0 0x1 0x51 0x80
*Nov 19 13:12:00.562: ISAKMP (0:3): Preshared authentication offered but does not match
policy!
*Nov 19 13:12:00.562: ISAKMP (0:3): atts are not acceptable. Next payload is 0
*Nov 19 13:12:00.562: ISAKMP (0:3): Checking ISAKMP transform 1 against priority 65535
policy
*Nov 19 13:12:00.562: ISAKMP: encryption 3DES-CBC
*Nov 19 13:12:00.562: ISAKMP: hash SHA
*Nov 19 13:12:00.562: ISAKMP: default group 1
*Nov 19 13:12:00.562: ISAKMP: auth pre-share
*Nov 19 13:12:00.562: ISAKMP: life type in seconds
*Nov 19 13:12:00.562: ISAKMP: life duration (VPI) of 0x0 0x1 0x51 0x80
*Nov 19 13:12:00.562: ISAKMP (0:3): Encryption algorithm offered does not match policy!
*Nov 19 13:12:00.562: ISAKMP (0:3): atts are not acceptable. Next payload is 0
*Nov 19 13:12:00.562: ISAKMP (0:3): no offers accepted!
```

Incorrect Preshared Key

```
*Nov 19 13:14:51.946: %CRYPTO-4-IKMP_BAD_MESSAGE: IKE message from 20.1.1.1 failed its
sanity check or is malformed
```

No Matching Peer

```
*Nov 17 16:58:51.789: ISAKMP (0:1): Checking IPSec proposal 1
*Nov 17 16:58:51.789: ISAKMP: transform 1, ESP_3DES
*Nov 17 16:58:51.789: ISAKMP: attributes in transform:
*Nov 17 16:58:51.789: ISAKMP: encaps is 1
*Nov 17 16:58:51.789: ISAKMP: SA life type in seconds
*Nov 17 16:58:51.789: ISAKMP: SA life duration (basic) of 3600
*Nov 17 16:58:51.789: ISAKMP: SA life type in kilobytes
*Nov 17 16:58:51.789: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
*Nov 17 16:58:51.789: ISAKMP: authenticator is HMAC-SHA
*Nov 17 16:58:51.789: IPSEC(validate_proposal): peer address 20.1.1.1 not found
*Nov 17 16:58:51.789: ISAKMP (0:1): atts not acceptable. Next payload is 0
*Nov 17 16:58:51.789: ISAKMP (0:1): phase 2 SA not acceptable!
```

Mismatched Transform Sets

```
*Nov 17 17:03:41.457: ISAKMP (0:2): Checking IPSec proposal 1
*Nov 17 17:03:41.457: ISAKMP: transform 1, ESP_3DES
*Nov 17 17:03:41.457: ISAKMP: attributes in transform:
*Nov 17 17:03:41.457: ISAKMP: encaps is 1
*Nov 17 17:03:41.457: ISAKMP: SA life type in seconds
*Nov 17 17:03:41.457: ISAKMP: SA life duration (basic) of 3600
*Nov 17 17:03:41.457: ISAKMP: SA life type in kilobytes
*Nov 17 17:03:41.457: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
*Nov 17 17:03:41.457: ISAKMP: authenticator is HMAC-SHA
*Nov 17 17:03:41.457: IPSEC(validate_proposal): transform proposal (prot 3, trans 3,
hmac_alg 2) not
supported
*Nov 17 17:03:41.457: ISAKMP (0:2): atts not acceptable. Next payload is 0
*Nov 17 17:03:41.457: ISAKMP (0:2): phase 2 SA not acceptable!
```

Crypto Map Not Applied to Interface

```
*Nov 17 17:07:38.289: ISAKMP (0:3): Checking IPSec proposal 1
```

```
*Nov 17 17:07:38.289: ISAKMP: transform 1, ESP_3DES
*Nov 17 17:07:38.289: ISAKMP: attributes in transform:
*Nov 17 17:07:38.289: ISAKMP: encaps is 1
*Nov 17 17:07:38.289: ISAKMP: SA life type in seconds
*Nov 17 17:07:38.289: ISAKMP: SA life duration (basic) of 3600
*Nov 17 17:07:38.289: ISAKMP: SA life type in kilobytes
*Nov 17 17:07:38.289: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
*Nov 17 17:07:38.289: ISAKMP: authenticator is HMAC-SHA
*Nov 17 17:07:38.289: IPSEC(validate_proposal): invalid local address 30.1.1.2
*Nov 17 17:07:38.289: ISAKMP (0:3): atts not acceptable. Next payload is 0
*Nov 17 17:07:38.289: ISAKMP (0:3): phase 2 SA not acceptable!
```

Incorrect Access List—Mismatched Proxies

```
*Nov 17 17:14:07.505: IPSEC(validate_proposal_request): proposal part #1,
(key eng. msg.) INBOUND local= 30.1.1.2, remote= 20.1.1.1,
local_proxy= 101.1.2.0/255.255.255.0/0/0 (type=4),
remote_proxy= 101.1.1.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-3des esp-sha-hmac ,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
*Nov 17 17:14:07.505: IPSEC(validate_transform_proposal): proxy identities not supported
*Nov 17 17:14:07.505: ISAKMP (0:4): IPSec policy invalidated proposal
*Nov 17 17:14:07.505: ISAKMP (0:4): phase 2 SA not acceptable!
```

Nonmatching VPN Group for VPN Clients

```
*Nov 19 16:29:46.389: AAA: parse name=ISAKMP-ID-AUTH idb type=-1 tty=-1
*Nov 19 16:29:46.389: AAA/MEMORY: create_user (0x63D75E28) user='ezvpn' ruser='NULL' ds0=0
port='ISAKMP-ID-AUTH' rem_addr='50.1.1.1' authen_type=NONE service=LOGIN priv=0
initial_task_id='0'
*Nov 19 16:29:46.389: ISAKMP (0:5): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
*Nov 19 16:29:46.389: ISAKMP (0:5): Old State = IKE_READY New State = IKE_R_AM_AAA_AWAIT
*Nov 19 16:29:46.389: ISAKMP-ID-AUTH AAA/AUTHOR/CRYPTO AAA(1787711329):
Port='ISAKMP-ID-AUTH'
list='localist' service=NET
*Nov 19 16:29:46.389: AAA/AUTHOR/CRYPTO AAA: ISAKMP-ID-AUTH(1787711329) user='ezvpn'
*Nov 19 16:29:46.389: ISAKMP-ID-AUTH AAA/AUTHOR/CRYPTO AAA(1787711329): send AV
service=ike
*Nov 19 16:29:46.389: ISAKMP-ID-AUTH AAA/AUTHOR/CRYPTO AAA(1787711329): send AV
protocol=ipsec
*Nov 19 16:29:46.389: ISAKMP-ID-AUTH AAA/AUTHOR/CRYPTO AAA(1787711329): found list
"localist"
*Nov 19 16:29:46.389: ISAKMP-ID-AUTH AAA/AUTHOR/CRYPTO AAA(1787711329): Method=LOCAL
*Nov 19 16:29:46.389: AAA/AUTHOR/IKMP/LOCAL: group ezvpn does not exist
*Nov 19 16:29:46.389: AAA/AUTHOR (1787711329): Post authorization status = ERROR
```

Failed Authentication—XAUTH for VPN Clients

Undefined user:

```
*Nov 19 16:36:18.829: AAA/AUTHEN(2414480242): Status=GETPASS
*Nov 19 16:36:18.829: AAA/AUTHEN/CONT (2414480242): Method=LOCAL
*Nov 19 16:36:18.829: AAA/AUTHEN(2414480242): User not found
*Nov 19 16:36:18.829: AAA/AUTHEN(2414480242): Status=FAIL
```

Incorrect password:

```
*Nov 19 16:38:50.373: AAA/AUTHEN(243054726): Status=GETPASS
*Nov 19 16:38:50.373: AAA/AUTHEN/CONT (243054726): Method=LOCAL
```

```
*Nov 19 16:38:50.373: AAA/AUTHEN(243054726): password incorrect
*Nov 19 16:38:50.373: AAA/AUTHEN(243054726): Status=FAIL
```

Incorrect Pool Definition for VPN Clients

```
*Nov 19 16:58:05.573: ISAKMP: Config payload REQUEST
*Nov 19 16:58:05.573: ISAKMP (0:1): checking request:
*Nov 19 16:58:05.573: ISAKMP: IP4_ADDRESS
*Nov 19 16:58:05.573: ISAKMP: IP4_NETMASK
*Nov 19 16:58:05.573: ISAKMP: IP4_DNS
*Nov 19 16:58:05.573: ISAKMP: IP4_DNS
*Nov 19 16:58:05.573: ISAKMP: IP4_NBNS
*Nov 19 16:58:05.573: ISAKMP: IP4_NBNS
*Nov 19 16:58:05.573: ISAKMP: SPLIT_INCLUDE
*Nov 19 16:58:05.573: ISAKMP: DEFAULT_DOMAIN
*Nov 19 16:58:05.573: ISAKMP (0:1): Input = IKE_MESG_FROM_PEER, IKE_CFG_REQ
*Nov 19 16:58:05.573: ISAKMP (0:1): Old State = IKE_P1_COMPLETE New State
_CONFIG_AUTHOR_AAA_AWAIT
*Nov 19 16:58:05.573: ISAKMP: got callback 1
*Nov 19 16:58:05.573: ISAKMP (0:1): attributes sent in message:
*Nov 19 16:58:05.573: Address: 0.2.0.0
*Nov 19 16:58:05.573: ISAKMP (0:1): Could not get address from pool!
*Nov 19 16:58:05.573: ISAKMP: Unknown Attr: IP4_NETMASK (0x2)
```

Incompatible ISAKMP Policy or Preshared Secrets

If no ISAKMP policies configured match, or if no preshared key for the negotiating peer is configured, the router tries the default policy (65535); if that does not match it fails ISAKMP negotiation.

A **show crypto isakmp sa** command shows the ISAKMP SA to be in MM_NO_STATE, meaning the main-mode failed:

```
ISAKMP (17): processing SA payload. Message ID = 0
ISAKMP (17): Checking ISAKMP transform 1 against priority 10 policy
                      encryption DES-CBC
                      hash SHA
                      default group 1
                      auth pre-share
ISAKMP (17): Checking ISAKMP transform 1 against priority 65535 policy
                      encryption DES-CBC
                      hash SHA
                      default group 1
                      auth pre-share
ISAKMP (17): atts are not acceptable. Next payload is 0
ISAKMP (17); no offers accepted!
ISAKMP (17): SA not acceptable!
%CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main Mode failed with peer
at 155.0.0.1
```

If the preshared secrets are not the same on both sides, the negotiation fails again, with the router complaining about sanity check failed. A **show crypto isakmp sa** command shows the ISAKMP SA to be in MM_NO_STATE, meaning the main mode failed:

```
ISAKMP (62): processing SA payload. message ID = 0
ISAKMP (62): Checking ISAKMP transform 1 against priority 10 policy
                      encryption DES-CBC
                      hash SHA
                      default group 1
                      auth pre-share
ISAKMP (62): atts are acceptable. Next payload is 0
ISAKMP (62): SA is doing preshared key authentication
```

```
ISAKMP (62): processing KE payload. message ID = 0
ISAKMP (62): processing NONCE payload. message ID = 0
ISAKMP (62): SKEYID state generated
ISAKMP (62); processing vendor id payload
ISAKMP (62): speaking to another IOS box!
ISAKMP: reserved no zero on payload 5!
%CRYPTO-4-IKMP_BAD_MESSAGE: IKE message from 155.0.0.1   failed its
sanity check or is malformed
```

Incompatible or Incorrect Access Lists

If the access lists on the two routers do not match or at least overlap, INVALID PROXY IDS or PROXY IDS NOT SUPPORTED results. It is recommended that access lists on the two routers be 'reflections' of each other. It is also recommended that you do not use the key word "any" in match address access lists.

```
3d00h: IPSec(validate_proposal_request): proposal part #1,
(key eng. msg.) dest= 172.16.171.5, src= 172.16.171.27,
    dest_proxy= 172.16.171.5/255.255.255.255/0/0 (type=1),
    src_proxy= 172.16.171.27/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
      lifedur= 0s and 0kb,
      spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
3d00h: validate proposal request 0
3d00h: IPSec(validate_transform_proposal): proxy identities not supported
3d00h: ISAKMP (0:3): IPSec policy invalidated proposal
3d00h: ISAKMP (0:3): phase 2 SA not acceptable!

Access List:
access list 110 permit ip host 172.16.171.5 host 172.16.171.30
```

Crypto Map on Incorrect Interface

The crypto map needs to be applied to the outgoing interface of the router. If you do not want to use the outside interface's IP as the local ID, use the **crypto map local address** command to specify the correct interface.

If there are physical as well as logical interfaces involved in carrying outgoing traffic, you must apply the crypto map to both.

Incorrect SA Selection by the Router

If there are multiple peers to a router, make sure that the match address access lists for each of the peers are mutually exclusive from the match address access list for the other peers.

If this is not done, the router chooses the incorrect crypto map to try to establish a tunnel with one of the peers.

```
Identity doesn't match negotiated identity
  (ip) dest_addr= 1.2.3.4, src_addr= 2.3.4.5, prot= 1
  (ident) local=5.5.5.5, remote=6.6.6.6
  local_proxy=1.2.3.5/255.255.255.255/0/0,
  remote_proxy=2.3.4.5/255.255.255.255/0/0

Access list for 5.6.7.8:
Access-list 100 permit ip host 1.2.3.5 host 5.6.7.9
Access-list 100 permit ip host 1.2.3.5 host 2.3.4.5

Access list for 1.2.3.4:
```

```
Access-list 110 permit ip host 1.2.3.5 host 2.3.4.5
```

Routing Establishment

A packet needs to be routed to the interface which has the crypto map configured on it before IPSec successfully starts.

Routes need to be established not only for the router to reach its peers address but also for the IP subnets in the packets after they are decrypted.

Use the **debug ip packet detailed** command to see if the routing is occurring correctly.

Note Different switching methods use completely different code paths. It is possible to have one switching method break IPSec and another function correctly. Try a different switching path (cef, fast switching, process switching) if you are running into an obscure problem.

Troubleshooting Tunnel Establishment

The following paragraphs provide information about troubleshooting IPSec tunnel establishment.

Interesting Traffic Received

The ping source and destination addresses matched the match address access list for the crypto map multipeer.

```
05:59:42: IPSec(sa_request): ,
(key eng. msg.) src= 172.21.114.123,
dest= 172.21.114.68,
```

The ‘src’ is the local tunnel end-point, the ‘dest’ is the remote crypto end point as configured in the map.

```
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
dest_proxy= 172.21.114.68/255.255.255.255/0/0 (type=1),
```

The src proxy is the src interesting traffic as defined by the match address access list; The dst proxy is the destination interesting traffic as defined by the match address access list

```
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
```

The protocol and the transforms are specified by the active crypto map, as are the lifetimes.

```
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
05:59:42: ISAKMP (1): beginning Main Mode exchange.....
```

Note that the SPI is still 0; the main mode of negotiation is being started.

Main Mode IKE Negotiation

```
05:59:51: ISAKMP (1): processing SA
payload. message ID = 0
05:59:51: ISAKMP (1): Checking ISAKMP
transform 1 against
priority 10 policy
```

Policy 10 is the only isakmp policy configured on the router.

```
05:59:51: ISAKMP:      encryption DES-CBC
05:59:51: ISAKMP:      hash SHA
05:59:51: ISAKMP:      default group 1
05:59:51: ISAKMP:      auth pre-share
```

These are the isakmp attributes being offered by the other side.

```
05:59:51: ISAKMP (1): atts are acceptable. Next payload is 0
```

The policy 10 on this router and the atts offered by the other side matched.

```
05:59:53: ISAKMP (1): SA is doing preshared key authentication
```

Preshared key authentication now begins.

ISAKMP Authentication

```
05:59:53: ISAKMP (1): processing KE payload. message ID = 0
05:59:55: ISAKMP (1): processing NONCE payload. message ID =0
```

Nonce from the far end is being processed.

```
05:59:55: ISAKMP (1): SKEYID state generated
05:59:55: ISAKMP (1): processing ID payload. message ID = 0
05:59:55: ISAKMP (1): processing HASH payload. message ID = 0
05:59:55: ISAKMP (1): SA has been authenticated
```

Preshared authentication has succeeded; the ISAKMP SA has been successfully negotiated.

Quick Mode Negotiation

Quick mode starts and the IPSec SA negotiations begin; ISAKMP negotiates for IPSec as well.

```
ISAKMP (1): beginning Quick Mode exchange, M-ID of 132876399
IPSec(key_engine): got a queue event...
IPSec(spi_response): getting spi 6008371161d for SA from 172.21.114.68   to 172.21.114.123
for prot 3
```

ISAKMP gets the SPI from the IPSec routine to offer to the far side.

```
ISAKMP (1): processing SA payload. message ID = 132876399
ISAKMP (1): Checking IPSec proposal 1
```

ISAKMP processes the IPSec attributes offered by the remote end:

```
SAKMP: transform 1, ESP_DES
```

This is the protocol offered by the remote end in accordance with its transform set.

```
ISAKMP:   attributes in transform:
ISAKMP:      encaps is 1
ISAKMP:      SA life type in seconds
ISAKMP:      SA life duration (basic) of 3600
ISAKMP:      SA life type in kilobytes
ISAKMP:      SA life duration (VPI) of
0x0 0x46 0x50 0x0
ISAKMP:      authenticator is HMAC-MD5
```

This is the payload authentication hash offered by the remote end in accordance with it s transform set.

```
ISAKMP (1): atts are acceptable.
```

The IPSec SA is now successfully negotiated. ISAKMP enters a state known as QM-IDLE.

IPSec SA Establishment

```
05:59:55: IPSec(validate_proposal_
request): proposal part #1,
(key eng. msg.) dest= 172.21.114.68,
src= 172.21.114.123,
dest_proxy= 172.21.114.68/255.255.
255.255/0/0 (type=1),
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
```

ISAMKP asks the IPSec routine to validate the IPSec proposal that it has negotiated with the remote side.

```
05:59:55: ISAKMP (1): Creating IPSec SAs
05:59:55:         inbound SA from 172.21.114.68   to   172.21.114.123
(proxy 172.21.114.68   to 172.21.114.123 )
05:59:55:         has spi 600837116 and conn_id 2 and flags 4
05:59:55:         lifetime of 3600 seconds
05:59:55:         lifetime of 4608000 kilobytes
05:59:55:         outbound SA from 172.21.114.123  to 172.21.114.68
(proxy 172.21.114.123  to 172.21.114.68  )
05:59:55:         has spi 130883577 and conn_id 3 and flags 4
05:59:55:         lifetime of 3600 seconds
05:59:55:         lifetime of 4608000 kilobytes
```

Two IPSec SAs are negotiated; an incoming SA with the SPI generated by the local machine and an outbound SA with the SPIs proposed by the remote end. Crypto engine entries have been created.

The ISAKMP routine informs the IPSec routine of the IPSec SA so that the SADB can be populated:

```
05:59:55: IPSec(initialize_sas): ,
(key eng. msg.) dest= 172.21.114.123, src= 172.21.114.68,
dest_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
src_proxy= 172.21.114.68/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
spi= 0x23D00BFC(600837116), conn_id= 2, keysize= 0, flags= 0x4
05:59:56: IPSec(initialize_sas): ,
(key eng. msg.) src= 172.21.114.123, dest= 172.21.114.68,
src_proxy= 172.21.114.123/255.255.255.255/0/0 (type=1),
dest_proxy= 172.21.114.68/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
spi= 0x7CD1FF9(130883577), conn_id= 3, keysize= 0, flags= 0x4
```

The IPSec routine populates the SADB with the IPSec entries:

```
05:59:56: IPSec(create_sa): sa created,
(sa) sa_dest= 172.21.114.123, sa_prot= 50,
sa_spi= 0x23D00BFC(600837116),
sa_trans= esp-des esp-md5-hmac , sa_conn_id= 2
05:59:56: IPSec(create_sa): sa created,
(sa) sa_dest= 172.21.114.68, sa_prot= 50,
sa_spi= 0x7CD1FF9(130883577),
sa_trans= esp-des esp-md5-hmac , sa_conn_id= 3
```

The SADB is updated and the IPSec SAs is initialized. The tunnel is now fully functional.

Routing Issues

Before IPSec begins functioning, a packet must be routed to the interface which has the crypto map configured on it.

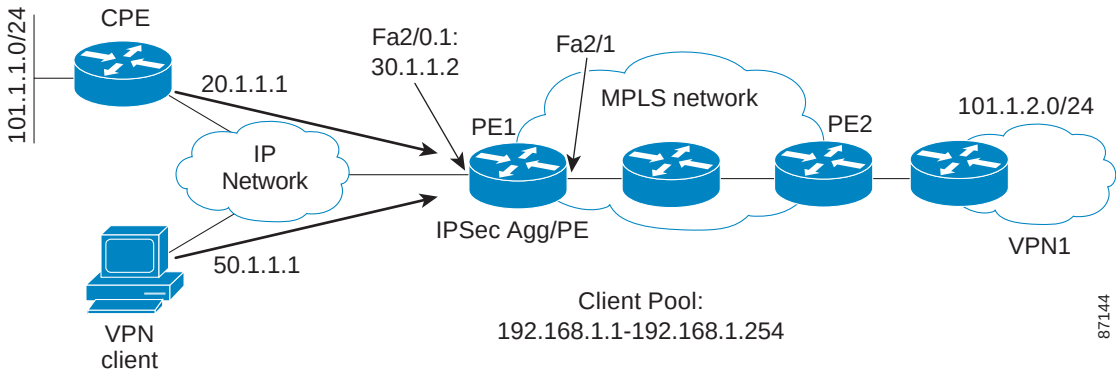
Routes are necessary for the router to reach its peers address and for the IP subnets in the packets after they have been decrypted.

Use the **debug ip packet** command to see if the routing is occurring correctly (be careful on busy networks).

Troubleshooting Example

Use Figure 3-1 for the following troubleshooting example.

Figure 3-1 Troubleshooting Flow



No Connectivity Between the Peers

Check if the peer address is reachable.

```
pe1#sh ip route 99.1.1.3
% Subnet not in table
pe1#sh ip bgp nei 99.1.1.3
BGP neighbor is 99.1.1.3, remote AS 100, internal link
BGP version 4, remote router ID 0.0.0.0
BGP state = Active
Should read:
pe1#sh ip bgp nei 99.1.1.3
BGP neighbor is 99.1.1.3, remote AS 100, internal link
BGP version 4, remote router ID 99.1.1.3
BGP state = Established, up for 00:01:38
```

Note To list routes in the global routing table use the **sh ip route** command. Use **sh ip route vrf** command for listing routes in a specific VRF. You can also use the **sh ip bgp vpnv4** *vrf* command to check routes learned/advertised through BGP.

```
pe1#sh ip route vrf vpn1
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
i - IS-IS, 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 30.1.1.1 to network 0.0.0.0

```
101.0.0.0/24 is subnetted, 2 subnets
S 101.1.1.0 [1/0] via 30.1.1.1
B 101.1.2.0 [200/0] via 99.1.1.3, 00:06:25
30.0.0.0/24 is subnetted, 1 subnets
C 30.1.1.0 is directly connected, FastEthernet2/0.1
S* 0.0.0.0/0 [1/0] via 30.1.1.1
pe1#sh ip bgp vpnv4 vrf vpn1
BGP table version is 5, local router ID is 99.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf vpn1)
*> 101.1.1.0/24 30.1.1.1 0 32768 ?
*>i101.1.2.0/24 99.1.1.3 0 100 0 ?
```

BGP Peer Miscommunication

A number of issues can arise if the VRF does not learn the BGP peer routes:

- Mismatch in route-target community strings for route import/export.

```
pe1#deb ip bgp up
BGP updates debugging is on
*Nov 18 13:33:47.756: BGP(2): 99.1.1.3 recv UPDATE w/ attr: nightspot 99.1.1.3, origin ?,
localpref 100, metric 0, extended community RT:200:1
*Nov 18 13:33:47.756: BGP(2): 99.1.1.3 rcvd 100:1:101.1.2.0/24 -- DENIED due to:
extended community not supported;
```

- Maximum number of route restrictions in the VRF.

```
ip vrf vpn1
rd 100:1
route-target export 100:1
route-target import 100:1
maximum routes 5 60
*Nov 18 14:16:40.356: BGP(2): Revise route installing 1 of 1 route for 1.1.1.0/24 ->
99.1.1.3
to vpn1 IP table
*Nov 18 14:16:40.356: BGP(2): Revise route installing 1 of 1 route for 2.1.1.0/24 ->
99.1.1.3
to vpn1 IP table
*Nov 18 14:16:40.356: %IPRT-3-ROUTELIMITEXCEEDED: IP routing table limit exceeded -
vpn1, 2.1.1.0/24
```

- Any kind of control information filtering based on AS_PATH, route-targets, or IP address.

Problems Forwarding Packets—Data Plane

Assuming that the route is in the VRF table, use the following checks to identify the problem:

Step 1 Verify the VPN labels on both the ends.

a. PE advertising the route.

```
pe3#sh mpls for vrf vpn1
Local Outgoing Prefix Bytes tag Outgoing Next Hop
tag tag or VC or Tunnel Id switched interface
17 Aggregate 101.1.2.0/24[V] 0
20 Aggregate 1.1.1.0/24[V] 0
21 Aggregate 2.1.1.0/24[V] 0
```

b. PE receiving the route.

```
pe1#sh ip bgp vpnv4 vrf vpn1 la
Network Next Hop In label/Out label
Route Distinguisher: 100:1 (vpn1)
1.1.1.0/24 99.1.1.3 nolabel/20
2.1.1.0/24 99.1.1.3 nolabel/21
101.1.1.0/24 30.1.1.1 19/nolabel
101.1.2.0/24 99.1.1.3 nolabel/17
```

Step 2 Verify that the CEF adjacency has been built out the correct interface.

```
pe1#sh ip cef vrf vpn1
Prefix Next Hop Interface
0.0.0.0/0 30.1.1.1 FastEthernet2/0.1
0.0.0.0/32 receive
1.1.1.0/24 125.1.10.1 FastEthernet2/1.1
2.1.1.0/24 125.1.10.1 FastEthernet2/1.1
30.1.1.0/24 attached FastEthernet2/0.1
30.1.1.0/32 receive
30.1.1.1/32 30.1.1.1 FastEthernet2/0.1
30.1.1.2/32 receive
30.1.1.255/32 receive
101.1.1.0/24 30.1.1.1 FastEthernet2/0.1
101.1.2.0/24 125.1.10.1 FastEthernet2/1.1
224.0.0.0/4 drop
224.0.0.0/24 receive
255.255.255.255/32 receive
```

Step 3 If this is in a controlled environment, then perform a **debug mpls packet** to verify MPLS packet switching on the PE in the MPLS to IP path.

```
*Nov 18 17:46:12.023: MPLS: Fa2/1.1: recvd: CoS=0, TTL=254, Label(s)=19
*Nov 18 17:46:12.023: MPLS: Fa2/0.1: xmit: (no label)
```

Step 4 Check for any kind of data traffic filtering at the egress/ingress PEs.

Step 5 CEF related packet drops can be troubleshot using **show cef drop** and **debug ip cef drop**.

Troubleshooting Tips

- Keep things simple (for example, mode config and xauth); use preshare; work your way up the feature list.
- Start from one host behind Cisco to one host behind the other device.
- Try to establish the connection from both sides; there might be issues starting it in a particular direction.
- Configure the two ends side by side.
- Make sure life time entries are matching both ends.
- Try transport mode if tunnel mode does not work.

- Remember that Cisco does not initiate aggressive mode but does accept it.
- In the current releases of the solution a default route (or a route to the endpoint) is needed both in the global as well as VRF routing table for endpoint reachability. Lack of global route will lead to IKE negotiation right away characterized by retransmissions in MM_SA_SETUP state.

```
*Nov 18 18:04:54.559: ISAKMP (0:2): sending packet to 20.1.1.1 (R) MM_SA_SETUP
*Nov 18 18:04:54.559: ISAKMP (0:2): Input = IKE_MESG_INTERNAL,
IKE_PROCESS_COMPLETE
*Nov 18 18:04:54.559: ISAKMP (0:2): Old State = IKE_R_MM1 New State = IKE_R_MM2
*Nov 18 18:05:04.547: ISAKMP (0:2): retransmitting due to retransmit phase 1
*Nov 18 18:05:04.547: ISAKMP (0:2): retransmitting phase 1 MM_SA_SETUP...
*Nov 18 18:05:05.047: ISAKMP (0:2): retransmitting phase 1 MM_SA_SETUP...
```

- Ensure that the reachability to the remote LAN segment is available in the VRF table either through statics or through GRE and a routing protocol.

```
ip route vrf vpn1 101.1.1.0 255.255.255.0 30.1.1.1
```

- The current release of the solution also requires a public facing interface/VPN to terminate the IPSec sessions (GRE being an exception). Thus ensure that the session is terminating on the correct interface.
- With VPN clients connecting to a multipoint interface and using RRI to install a route in the VRF routing table, Cisco IOS software does not forward packets in the return path (MPLS to IP) out a multipoint interface due to a lack off next-hop. This can be seen in the following debug that packet is received with a VPN tag but is not untagged and forwarded.

```
*Nov 19 15:53:02.757: MPLS: Fa2/1.1: recvd: CoS=0, TTL=254, Label(s)=20
*Nov 19 15:53:04.753: MPLS: Fa2/1.1: recvd: CoS=0, TTL=254, Label(s)=20
*Nov 19 15:53:06.749: MPLS: Fa2/1.1: recvd: CoS=0, TTL=254, Label(s)=20
```

With multipoint interfaces, avoid using RRI for route injection. Instead, use a static route for the subnet pointing to a next hop that can then be redistributed into BGP.

Troubleshooting Commands

Helpful commands for troubleshooting IPSec include the **show**, **clear**, and **debug** commands listed below. This appendix contains examples of these commands.

- Show Commands
- Clear Commands
- Debug Commands

For a detailed explanation of the show and clear commands and additional examples, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122csum/csum2/index.htm.

For a detailed explanation of the debug commands and additional examples, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122debug/index.htm.

Show Commands

This section describes the following commands:

- show cry isakmp sa
- show crypto engine configuration
- show crypto engine connections active
- show crypto engine connections dropped-packet
- show crypto ip transform
- show crypto ipsec sa
- show crypto ipsec session/show crypto ipsec sa
- show crypto ipsec session-key
- show crypto isakmp policy
- show crypto isakmp sa
- show crypto map
- show crypto map interface serial 0
- show crypto map tag test

show cry isakmp sa

Note The command output below displays an ISAKMP (IKE) security association (SA) table. In the example, an SA exists between 172.21.30.71 and 172.21.30.70. The peer should have an SA entry in the same state as this router's output.

```
dt1-7ka#show cry isakmp sa
dst src state conn-id slot
172.21.30.70 172.21.30.71 QM_IDLE 47 5
```

2. show cry isakmp policy

Note The command output below shows the policy objects configured. In this case, policies 1, 2, and 4 are used, in addition to the default. The policies are proposed to the peer in order, with **1** preferable.

```
dt1-45a#show cry isakmp policy
Protection suite of priority 1
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Message Digest 5
authentication method: Rivest-Shamir-Adleman Signature
Diffie-Hellman group: #1 (768 bit)
lifetime: 180 seconds, no volume limit
Protection suite of priority 2
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Secure Hash Standard
authentication method: Pre-Shared Key
Diffie-Hellman group: #2 (1024 bit)
lifetime: 180 seconds, no volume limit
Protection suite of priority 4
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Message Digest 5
authentication method: Pre-Shared Key
Diffie-Hellman group: #2 (1024 bit)
lifetime: 180 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
```

show crypto engine configuration

```
wan2611#show crypto engine configuration
slot: 0
engine name: unknown
engine type: software
serial number: 01496536
platform: rp crypto engine
crypto lib version: 10.0.0
Encryption Process Info:
input queue top: 140
input queue bot: 140
input queue count: 0
```

show crypto engine connections active

```
wan2611# show crypto engine connections active
ID Interface IP-Address State Algorithm Encrypt Decrypt
9 Serial0 20.20.20.21 set HMAC_SHA 0 240
10 Serial0 20.20.20.21 set HMAC_SHA 240 0
```

show crypto engine connections dropped-packet

```
lab-isdn1#show crypto engine connections dropped-packet
Interface IP-Address Drop Count
BRI0 12.12.12.13 4
```

show crypto ip transform

Note The configuration below uses the same router as above, but different **show** commands. The transform proposals, the settings they will negotiate, and the defaults are all provided.

```
S3-2611-2#show cry ip transform
Transform proposal Elvis: { ah-sha-hmac }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
{ esp-des }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
Transform proposal Bubba: { ah-rfc1828 }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
{ esp-des esp-md5-hmac }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
Transform proposal BarneyDino: { ah-md5-hmac }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
```

show crypto ipsec sa

```
wan2611#show crypto ipsec sa
interface: Serial0
Crypto map tag: test, local addr. 20.20.20.21
local ident (addr/mask/prot/port): (50.50.50.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (60.60.60.0/255.255.255.0/0/0)
current_peer: 20.20.20.20
PERMIT, flags={origin_is_acl,ident_is_ipsec,}
#pkts encaps: 320, #pkts encrypt: 320, #pkts digest 320
local crypto endpt.: 20.20.20.21, remote crypto endpt.: 20.20.20.20
path mtu 1500, media mtu 1500
current outbound spi: 6625CD
inbound esp sas:
```

```
spi: 0x1926112F(421859631)
transform: esp-des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 11, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607971/3354)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
spi: 0x12050DD2(302321106)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 9, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607958/3354)
replay detection support: Y
outbound esp sas:
spi: 0x3262313(52830995)
transform: esp-des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 12, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607971/3354)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
spi: 0x6625CD(6694349)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 10, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607958/3354)
replay detection support: Y
```

show crypto ipsec session/show crypto ipsec sa

Note The command output below shows the current IPSec security associations of this router, which is configured for AH SA for both incoming and outgoing traffic.

```
S3-2611-2#show cry ipsec session
Session key lifetime: 4608000 kilobytes/3600 seconds
S3-2611-2#show cry ipsec sa
interface: Ethernet0
Crypto map tag: ToOtherRouter, local addr. 192.168.1.2
local ident (addr/mask/prot/port): (192.168.45.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (192.168.3.0/255.255.255.0/0/0)
current_peer: 192.168.1.1
PERMIT, flags={origin_is_acl,}
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
#pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
#send errors 5, #recv errors 0
local crypto endpt.: 192.168.1.2, remote crypto endpt.: 192.168.1.1
path mtu 1500, media mtu 1500
current outbound spi: 25081A81
inbound esp sas:
inbound ah sas:
spi: 0x1EE91DDC(518594012)
transform: ah-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 16, crypto map: ToOtherRouter
sa timing: remaining key lifetime (k/sec): (4608000/3423)
replay detection support: Y
outbound esp sas:
```

```
outbound ah sas:
spi: 0x25081A81(621288065)
transform: ah-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 17, crypto map: ToOtherRouter
sa timing: remaining key lifetime (k/sec): (4608000/3424)
replay detection support: Y
```

show crypto ipsec session-key

```
wan2611#show crypto ipsec session-key
Session key lifetime: 4608000 kilobytes/3600 seconds
wan2611#show crypto ipsec transform-proposal
Transform proposal auth2: { ah-sha-hmac }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
{ esp-des esp-sha-hmac }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
```

show crypto isakmp policy

```
wan2611#show crypto isakmp policy
Protection suite of priority 1
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Secure Hash Standard
authentication method: Pre-Shared Key
Diffie-Hellman group: #2 (1024 bit)
lifetime: 240 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
```

show crypto isakmp sa

```
lab-isdn1#show crypto isakmp sa
dst src state conn-id slot
12.12.12.12 12.12.12.13 QM_IDLE 4 0
lab-isdn1#show crypto ipsec sa
interface: BRI0
Crypto map tag: test, local addr. 12.12.12.13
local ident (addr/mask/prot/port): (40.40.40.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (20.20.20.0/255.255.255.0/0/0)
current_peer: 12.12.12.12
PERMIT, flags={origin_is_acl,ident_is_ipsec,}
#pkts encaps: 89, #pkts encrypt: 89, #pkts digest 89
#pkts decaps: 89, #pkts decrypt: 89, #pkts verify 89
#send errors 11, #recv errors 0
local crypto endpt.: 12.12.12.13, remote crypto endpt.: 12.12.12.12
path mtu 1500, media mtu 1500
current outbound spi: 6B024AB
inbound esp sas:
```

```
spi: 0x21240B07(556010247)
transform: esp-des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 7, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607989/3062)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
spi: 0x4F60465(83231845)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 5, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607984/3062)
replay detection support: Y
outbound esp sas:
spi: 0x19591660(425268832)
transform: esp-des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 8, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607989/3062)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
spi: 0x6B024AB(112207019)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 6, crypto map: test
sa timing: remaining key lifetime (k/sec): (4607984/3062)
replay detection support: Y
lab-isdn1#show crypto ipsec session-key
Session key lifetime: 4608000 kilobytes/3600 seconds
lab-isdn1#show crypto ipsec transform-proposal
Transform proposal mypolicy: { ah-sha-hmac }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
{ esp-des esp-sha-hmac }
supported settings = { Tunnel, },
default settings = { Tunnel, },
will negotiate = { Tunnel, },
lab-isdn1#show crypto map interface bri 0
Crypto Map "test" 10 ipsec-isakmp
Peer = 12.12.12.12
Extended IP access list 144
access-list 144 permit ip
source: addr = 40.40.40.0/0.0.0.255
dest: addr = 20.20.20.0/0.0.0.255
Current peer: 12.12.12.12
Session key lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform proposals={ mypolicy, }
lab-isdn1#show crypto map tag test
Crypto Map "test" 10 ipsec-isakmp
Peer = 12.12.12.12
Extended IP access list 144
access-list 144 permit ip
source: addr = 40.40.40.0/0.0.0.255
dest: addr = 20.20.20.0/0.0.0.255
Current peer: 12.12.12.12
Session key lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform proposals={ mypolicy, }
```

show crypto map

Note The command output below shows the **crypto map** command, the ACLs and the transform proposals applied to this crypto map, the peers, and the key lifetime.

```
S3-2611-2#show cry map
Crypto Map "ToOtherRouter" 10 ipsec-isakmp
Peer = 192.168.1.1
Extended IP access list 101
access-list 101 permit ip
source: addr = 192.168.45.0/0.0.0.255
dest: addr = 192.168.3.0/0.0.0.255
Connection Id = UNSET (0 established, 0 failed)
Current peer: 192.168.1.1
Session key lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform proposals={ Elvis, Bubba, BarneyDino, }
```

show crypto map interface serial 0

```
wan2611#show crypto map interface serial 0
Crypto Map "test" 10 ipsec-isakmp
Peer = 20.20.20.20
Extended IP access list 133
access-list 133 permit ip
source: addr = 50.50.50.0/0.0.0.255
dest: addr = 60.60.60.0/0.0.0.255
Current peer: 20.20.20.20
Session key lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform proposals={ auth2, }
```

show crypto map tag test

```
wan2611#show crypto map tag test
Crypto Map "test" 10 ipsec-isakmp
Peer = 20.20.20.20
Extended IP access list 133
access-list 133 permit ip
source: addr = 50.50.50.0/0.0.0.255
dest: addr = 60.60.60.0/0.0.0.255
Current peer: 20.20.20.20
Session key lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform proposals={ auth2,}
```

Clear Commands

This section describes the following commands:

- clear crypto isakmp
- clear crypto sa

clear crypto isakmp

```
lab-isdn1#clear crypto isakmp
lab-isdn1#
*Mar 21 20:58:34.503: ISADB: reaper checking SA, conn_id = 4 DELETE IT!
*Mar 21 20:58:34.507: generate hmac context for conn id 4
*Mar 21 20:58:34.519: CRYPTO(epa_release_crypto_conn_entry): released conn 4
lab-isdn1#
```

clear crypto sa

```
lab-isdn1#clear crypto sa
lab-isdn1#
*Mar 21 20:58:42.495: IPSEC(delete_sa): deleting SA,
(sa) sa_dest= 12.12.12.13, sa_prot= 51,
sa_spi= 0x4F60465(83231845),
sa_trans= ah-sha-hmac , sa_conn_id= 5
*Mar 21 20:58:42.499: CRYPTO(epa_release_crypto_conn_entry): released conn 5
*Mar 21 20:58:42.499: IPSEC(delete_sa): deleting SA,
(sa) sa_dest= 12.12.12.12, sa_prot= 51,
sa_spi= 0x6B024AB(112207019),
sa_trans= ah-sha-hmac , sa_conn_id= 6
*Mar 21 20:58:42.503: CRYPTO(epa_release_crypto_conn_entry): released conn 6
*Mar 21 20:58:42.503: IPSEC(delete_sa): deleting SA,
(sa) sa_dest= 12.12.12.13, sa_prot= 50,
sa_spi= 0x21240B07(556010247),
*Mar 21 20:58:42.507: CRYPTO(epa_release_crypto_conn_entry): released conn 7
*Mar 21 20:58:42.507: IPSEC(delete_sa): deleting SA,
(sa) sa_dest= 12.12.12.12, sa_prot= 50,
sa_spi= 0x19591660(425268832),
sa_trans= esp-des esp-sha-hmac , sa_conn_id= 8
*Mar 21 20:58:42.511: CRYPTO(epa_release_crypto_conn_entry): released conn 8
lab-isdn1#
```

Debug Commands

This section provides sample outputs from executing the **debug** command during a normal IKE/IPSec session between two routers. The routers in the example use a pre-shared key. The source router has the **debug cry isakmp** and **debug cry ipsec** commands enabled. The peer has the same two commands, plus the **debug ip packet** command enabled.

For information on debug scenarios that illustrate configuration problems (for example, a policy mismatch), see Appendix B, “Sample Problem Scenarios.”

Configuring on the Source Router

```
dt3-4kb#ping 192.168.10.66
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.66, timeout is 2 seconds:
IPSEC(sa_request): ,
(key eng. msg.) SRC= 192.168.10.38, dest= 192.168.10.66,
src_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
dest_proxy= 192.168.10.66/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-rfc1829 ,
lifedur= 190s and 4608000kb,
```

```
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
IPSEC(sa_request): ,
(key eng. msg.) SRC= 192.168.10.38, dest= 192.168.10.66,
src_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
dest_proxy= 192.168..10.66/255.255.255.255/0/0 (type=1),
protocol= AH, transform= ah-md5-hmac ,
lifedur= 190s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
IPSEC(sa_request): ,
(key eng. msg.) SRC= 192.168.10.38, dest= 192.168.10.66,
src_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
dest_proxy= 192.168.10.66/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des ,
lifedur= 190s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
IPSEC(sa_request): ,
(key eng. msg.) SRC= 192.168.10.38, dest= 192.168.10.66,
src_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
dest_proxy= 192.168.10.66/255.255.255.255/0/0 (type=1),
protocol= AH, transform= ah-rfc1828 ,
lifedur= 190s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
```

Note that the router offers all of the available transforms to the peer.

```
ISAKMP (134): beginning Main Mode exchange
ISAKMP (134): processing SA payload. message ID = 0
ISAKMP (134): Checking ISAKMP transform 1 against priority 1 policy
ISAK.!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 8/8/8 ms
dt3-4kb#MP: encryption DES-CBC
ISAKMP: hash MD5
ISAKMP: default group 1
ISAKMP: auth pre-share
ISAKMP (134): atts are acceptable. Next payload is 0
ISAKMP (134): SA is doing pre-shared key authentication
ISAKMP (134): processing KE payload. message ID = 0
ISAKMP (134): processing NONCE payload. message ID = 0
ISAKMP (134): SKEYID state generated
ISAKMP (134): processing ID payload. message ID = 0
ISAKMP (134): processing HASH payload. message ID = 0
ISAKMP (134): SA has been authenticated
ISAKMP (134): beginning Quick Mode exchange, M-ID of -1517735742
IPSEC(key_engine): got a queue event...
IPSEC(spi_response): getting spi 383061134 for SA
from 192.168.10.66 to 192.168.10.38 for prot 3
IPSEC(spi_response): getting spi 542967145 for SA
from 192.168.10.66 to 192.168.10.38 for prot 2
IPSEC(spi_response): getting spi 359212595 for SA
from 192.168.10.66 to 192.168.10.38 for prot 3
IPSEC(spi_response): getting spi 366153874 for SA
from 192.168.10.66 to 192.168.10.38 for prot 2
ISAKMP (134): processing SA payload. message ID = -1517735742
ISAKMP (134): Checking IPSec proposal 1
ISAKMP: transform 1, ESP_DES_IV64
ISAKMP: attributes in transform:
ISAKMP: encaps is 1
ISAKMP: SA life type in seconds
ISAKMP: SA life duration (basic) of 190
ISAKMP: SA life type in kilobytes
ISAKMP: SA life duration (VPI) of
0x0 0x46 0x50 0x0
ISAKMP (134): atts are acceptable.
```

If debug entries appear for ISAKMP even after the IKE has been established, remember that IKE is the entity that negotiates IPSec SAs on behalf of IPSec.

```
IPSEC(validate_proposal_request): proposal part #1,
(key eng. msg.) dest= 192.168.10.66, SRC= 192.168.10.38,
dest_proxy= 192.168.10.66/255.255.255.255/0/0 (type=1),
src_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-rfc1829 ,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
ISAKMP (134): processing NONCE payload. message ID = -1517735742
ISAKMP (134): processing ID payload. message ID = -1517735742
ISAKMP (134): processing ID payload. message ID = -1517735742
ISAKMP (134): Creating IPSec SAs
inbound SA from 192.168.10.66 to 192.168.10.38 (proxy
192.168.10.66 to 192.168.10.38 )
has spi 383061134 and conn_id 135 and flags 4
lifetime of 190 seconds
lifetime of 4608000 kilobytes
outbound SA from 192.168.10.38 to 192.168.10.66 (proxy
192.168.10.38 to 192.168.10.66 )
has spi 138874178 and conn_id 136 and flags 4
lifetime of 190 seconds
lifetime of 4608000 kilobytes
IPSEC(key_engine): got a queue event...
IPSEC(initialize_sas): ,
IPSEC(initialize_sas): ,
(key eng. msg.) dest= 192.168.10.38, SRC= 192.168.10.66,
dest_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
src_proxy= 192.168.10.66/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-rfc1829 ,
lifedur= 190s and 4608000kb,
spi= 0x16D50C8E(383061134), conn_id= 135, keysize= 0, flags= 0x4
IPSEC(initialize_sas): ,
(key eng. msg.) SRC= 192.168.10.38, dest= 192.168.10.66,
src_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
dest_proxy= 192.168.10.66/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-rfc1829 ,
lifedur= 190s and 4608000kb,
spi= 0x8470D42(138874178), conn_id= 136, keysize= 0, flags= 0x4
IPSEC(create_sa): sa created,
(sa) sa_dest= 192.168.10.38, sa_prot= 50,
sa_spi= 0x16D50C8E(383061134),
sa_trans= esp-rfc1829 , sa_conn_id= 135
IPSEC(create_sa): sa created,
(sa) sa_dest= 192.168.10.66, sa_prot= 50,
sa_spi= 0x8470D42(138874178),
sa_trans= esp-rfc1829 , sa_conn_id= 136
```

The lines above show that IPSec SAs have been created and encrypted traffic can now pass.

```
dt3-4kb#ping 192.168.10.66
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.66, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/11/12 ms
dt3-4kb#
```

Run the **show** command on the source router following IKE/IPSec negotiation.

```
dt3-4kb#show cry is sa
dst src state conn-id slot
192.168.10.66 192.168.10.38 QM_IDLE 134 0
dt3-4kb#show cry is pol
```

```
Protection suite of priority 1
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Message Digest 5
authentication method: Pre-Shared Key
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
dt3-4kb#
dt3-4kb#show cry ipsec sa
interface: TokenRing0
interface: Ethernet0
Crypto map tag: armadillo, local addr. 192.168.10.38
local ident (addr/mask/prot/port): (192.168.10.38/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (192.168.10.66/255.255.255.255/0/0)
current_peer: 192.168.10.66
PERMIT, flags={origin_is_acl,}
#pkts encaps: 8, #pkts encrypt: 8, #pkts digest 0
#pkts decaps: 8, #pkts decrypt: 8, #pkts verify 0
#send errors 2, #recv errors 0
local crypto endpt.: 192.168.10.38, remote crypto endpt.:
192.168.10.66
path mtu 1500, media mtu 1500
current outbound spi: 0
inbound esp sas:
inbound ah sas:
outbound esp sas:
outbound ah sas:
```

The peer router shows the same ping sequence on the destination router.

```
dt3-45a#
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 100, rcvd 1
ISAKMP (165): processing SA payload. message ID = 0
ISAKMP (165): Checking ISAKMP transform 1 against priority 1 policy
ISAKMP: encryption DES-CBC
ISAKMP: hash MD5
ISAKMP: default group 1
ISAKMP: auth pre-share
ISAKMP (165): atts are acceptable. Next payload is 0
ISAKMP (165): SA is doing pre-shared key authentication
IP: s=192.168.10.66 (local), d=192.168.10.38 (Ethernet0), len 100, sending
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 180, rcvd 1
ISAKMP (165): processing KE payload. message ID = 0
ISAKMP (165): processing NONCE payload. message ID = 0
ISAKMP (165): SKEYID state generated
IP: s=192.168.10.66 (local), d=192.168.10.38 (Ethernet0), len 180, sending
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 96, rcvd 1
ISAKMP (165): processing ID payload. message ID = 0
SAKMP (165): processing HASH payload. message ID = 0
ISAKMP (165): SA has been authenticated
IP: s=192.168.10.66 (local), d=192.168.10.38 (Ethernet0), len 96, sending
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 328, rcvd 1
ISAKMP (165): processing SA payload. message ID = -1517735742
ISAKMP (165): Checking IPSec proposal 1
ISAKMP: transform 1, ESP_DES_IV64
ISAKMP: attributes in transform:
ISAKMP: encaps is 1
ISAKMP: SA life type in seconds
ISAKMP: SA life duration (basic) of 190
```

```
ISAKMP: SA life type in kilobytes
ISAKMP: SA life duration (VPI) of
0x0 0x46 0x50 0x0
ISAKMP (165): atts are acceptable
```

IKE performs its operation and then starts IPSec.

```
IPSEC(validate_proposal_request): proposal part #1,
(key eng. msg.) dest= 192.168.10.66, SRC= 192.168.10.38,
dest_proxy= 192.168.10.66/255.255.255.255/0/0 (type=1),
src_proxy= 192.168.10.38/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-rfc1829 ,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
ISAKMP (165): processing NONCE payload. message ID = -1517735742
ISAKMP (165): processing ID payload. message ID = -1517735742
ISAKMP (165): processing ID payload. message ID = -1517735742
IPSEC(key_engine): got a queue event...
IPSEC(spi_response): getting spi 138874178 for SA
from 192.168.10.38 to 192.168.10.66 for prot 3
IP: s=192.168.10.66 (local), d=192.168.10.38 (Ethernet0), len 184, send
ing
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 80, rcvd 1
ISAKMP (165): Creating IPSec SAs
inbound SA from 192.168.10.38 to 192.168.10.66 (proxy
192.168.10.38 to 192.168.10.66 )
has spi 138874178 and conn_id 166 and flags 4
lifetime of 190 seconds
lifetime of 4608000 kilobytes
outbound SA from 192.168.10.66 to 192.168.10.38 (proxy
192.168.10.66 to 192.168.10.38 )
has spi 383061134 and conn_id 167 and flags 4
lifetime of 190 seconds
lifetime of 4608000 kilobytes
IPSEC(key_engine): got a queue event...
IPSEC(initialize_sas): ,
(key eng. msg.) dest= 192.168.10.66, SRC= 192.168.10.38,
dest_proxy= 192.168.10.66/0.0.0.0/0/0 (type=1),
src_proxy= 192.168.10.38/0.0.0.0/0/0 (type=1),
protocol= ESP, transform= esp-rfc1829 ,
lifedur= 190s and 4608000kb,
spi= 0x8470D42(138874178), conn_id= 166, keysize= 0, flags= 0x4
IPSEC(initialize_sas): ,
(key eng. msg.) SRC= 192.168.10.66, dest= 192.168.10.38,
src_proxy= 192.168.10.66/0.0.0.0/0/0 (type=1),
dest_proxy= 192.168.10.38/0.0.0.0/0/0 (type=1),
protocol= ESP, transform= esp-rfc1829 ,
lifedur= 190s and 4608000kb,
spi= 0x16D50C8E(383061134), conn_id= 167, keysize= 0, flags= 0x4
IPSEC(create_sa): sa created,
(sa) sa_dest= 192.168.10.66, sa_prot= 50,
sa_spi= 0x8470D42(138874178),
sa_trans= esp-rfc1829 , sa_conn_id= 166
IPSEC(create_sa): sa created,
(sa) sa_dest= 192.168.10.38, sa_prot= 50,
sa_spi= 0x16D50C8E(383061134),
sa_trans= esp-rfc1829 , sa_conn_id= 167
ISADB: reaper checking SA, conn_id = 165
!-- The IPSec SAs are created, and now we see the packets going back and forth.
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 136, rcvd 1
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 100, rcvd 1
IP: s=192.168.10.66 (local), d=192.168.10.38 (Ethernet0), len 100, sending
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 136, rcvd 1
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 100, rcvd 1
```

```
IP: s=192.168.10.66 (local), d=192.168.10.38 (Ethernet0), len 100, sending
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 136, rcvd 1
IP: s=192.168.10.38 (Ethernet0), d=192.168.10.66, len 100, rcvd 1
IP: s=192.168.10.66 (local), d=192.168.10.38 (Ethernet0), len 100, sending
dt3-45a#
```

Show Commands on the Peer Router

The following illustrates a series of **show** command output after IKE/IPSec negotiation has taken place.

```
dt3-45a#show cry isakmp sa
dst src state conn-id slot
192.168.10.66 192.168.10.38 QM_IDLE 165 0
dt3-45a#show cry isakmp pol
Protection suite of priority 1
encryption algorithm: DES - Data Encryption Standard (56 bit
keys).
hash algorithm: Message Digest 5
authentication method: Pre-Shared Key
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
dt3-45a#show cry ipsec sa
interface: TokenRing0
interface: Ethernet0
Crypto map tag: armadillo, local addr. 192.168.10.66
local ident (addr/mask/prot/port):
(192.168.10.66/255.255.255.255/0dt3-45a#sho cry isakmp sa
dst src state conn-id slot
192.168.10.66 192.168.10.38 QM_IDLE 165 0
dt3-45a#show cry isakmp pol
Protection suite of priority 1
encryption algorithm: DES - Data Encryption Standard (56 bit
keys).
hash algorithm: Message Digest 5
authentication method: Pre-Shared Key
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
dt3-45a#show cry ipsec sa
interface: TokenRing0
interface: Ethernet0
Crypto map tag: armadillo, local addr. 192.168.10.66
local ident (addr/mask/prot/port): (192.168.10.66/255.255.255.255/0
/0)
remote ident (addr/mask/prot/port): (192.168.10.38/255.255.255.255/0/0)
current_peer: 192.168.10.38
PERMIT, flags={origin_isakmp_acl,}
#pkts encaps: 8, #pkts encrypt: 8, #pkts digest 0
#pkts decaps: 8, #pkts decrypt: 8, #pkts verify 0
```

```
#send errors 0, #recv errors 0
local crypto endpt.: 192.168.10.66, remote crypto endpt.:
192.168.10.38
path mtu 1500, media mtu 1500
current outbound spi: 0
inbound esp sas:
inbound ah sas:
outbound esp sas:
outbound ah sas:
dt3-45a#show cry map
Crypto Map "armadillo" 1 ipsec-isakmp
Peer = 192.168.10.38
Extended IP access list 101
access-list 101 permit ip
source: addr = 192.168.10.66/0.0.0.0
dest: addr = 192.168.10.38/0.0.0.0
Connection Id = UNSET (0 established, 0 failed)
Current peer: 192.168.10.38
Session key lifetime: 4608000 kilobytes/190 seconds
PFS (Y/N): N
Transform proposals={ MamaBear, PapaBear, BabyBear, }
dt3-45a#
```

ICMP Echo Request

The following comments address each section of the output and relate the collected information back to the functioning of IKE defined by the RFCs.

These lines show the entry of the **ping** command and the message generated to the operator indicating that the operation has started.

```
R1# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
```

The next lines show the SPD entries for the SA that are included in the SA payload as the proposal and associated transform sets. These lines display the SPD's identification of the destination and source, which allows the definition of packet security. A single protocol is configured, resulting in one proposal that contains two transform sets: DES encryption and HAMC-MD5 authentication. These lines define neither a lifetime limit nor a maximum data transmission; instead, the default settings define these values.

Note The maximum data is a default unit of 4 GB. Because the configuration does not define a lifetime value or an attribute, 4 GB is assumed to be the maximum data possible for the ISA. This value is realized by setting the 32-bit attribute in the SA transform set to 0x0 0x46 0x50 0x0. In addition, Phase 1 operations use cookies for IKE SA creation, and the SPI is set to zero. The connection ID is zero, because this packet is the first generated for the VPN. No keys are defined, because DH has not transpired, and the algorithms in the proposal have not been accepted.

```
IPSEC(sa_request):
(key eng. msg.) src = 10.1.1.2, dest = 10.1.1.1,
src_proxy = 10.1.0.0/255.255.0.0/0/0 (type = 4)
dest_proxy = 10.1.0.0/255.255.0.0/0/0 (type = 4),
protocol = ESP, transform = esp-des esp-md5-hmac,
lifedur = 3600s and 4608000kb,
spi = 0 x 0(0), conn_id =0, keysize=0, flags = 0 x 4004
```

The main mode starts.

```
ISAKMP (10): beginning Main Mode exchange
ISAKMP (10): sending packet to 10.1.1.1 (I) MM_NO_STATE
ISAKMP (10): received packet from 10.1.1.1 (I) MM_NO_STATE
ISAKMP (10): processing SA payload, message ID = 0
```

The section below shows the sending of the Main Mode packet (containing the ISAKMP header and SA payload information) to R2. The responder replies with a single proposal previously established in the original proposal from R1. Should several proposals be provided to R2, only one could be selected and returned unchanged.

This section shows that the SA information has been received from R2. You must to verify the proposal's data has not been changed, and that the accepted proposal is valid according to the policy maintained within the SPD. This section's final line shows that the payload is processed and accepted. The "next payload" statement indicates that the return proposal's generic header is at the end of the SA payload data chain.

```
ISAKMP (10): Checking ISAKMP transform 1 against priority 1 policy
ISAKMP: encryption DES-CBC
ISAKMP: hash SHA
ISAKMP: default group 1
ISAKMP: auth pre-share
ISAKMP (10): atts are acceptable. Next payload is 0
```

The line below shows the router's understanding of the ISAKMP as a pre-shared key b based on the peer's IP address. On this basis, the router makes an alignment for computation. ISAKMP (10): SA is doing pre-shared key authentication using id type ID_IP_IPV4_ADDR. The lines below detail the exchange of the third and fourth IKE payloads containing the nonces and public DH values. After this information is obtained and verified, the keys can be created.

The size and number of keys are indicated by the proposal payload information agreed upon in the first two messages. The two systems assume that a limited relationship has been formed, based solely on the peer's IP address and associated policies. The grouping's final line indicates that the keys have been generated. Note that the SKEYID is generated by the responder prior to sending its KE and nonce payloads to the initiator. In so doing, the responder is prepared for future exchanges. However, it has committed to the generation of the key prior to the initiator or any solid authentication. Therefore, this order of events could be processed as a "weakness" and possible denial-of-service against the router. Because the initiator can send only a few messages to the responder that could result in the responder's creation of keys, the initiator is free to flood the responder, consuming system resources.

```
ISAKMP (10): sending packet to 10.1.1.1 (I) MM_SA_SETUP
ISAKMP (10): receiving packet from 10.1.1.1 (I) MM_SA_SETUP
ISAKMP (10): processing KE payload. message ID = 0
ISAKMP (10): processing NONCE payload. message ID = 0
ISAKMP (10): SKEYID state generated
```

The following lines add one more attribute to the payload chain, taking advantage of the optional vendor ID payload entry.

```
ISAKMP (10): processing vendor id payload
ISAKMP (10): speaking to another IOS box!
```

The following section displays the creation of the ID payload to be presented to the responder. The "next-payload" type of *8* signifies that the HASH_I is to follow. The type of *1* states that the ID payload will contain the IP address of the initiator. The authentication process assumes type 1 from payloads 1 and 2. However, by providing the proper ID payload, there is no confusion, especially if NAT is involved. Protocol and port are defined by the IPSec DOI to be included. For ISAKMP, the protocol is UDP (17), and the port is 500.

Note The "length statement," as opposed to the total length statement, can be confusing. The RFC states that the length represents the payload and the generic header. From the information gathered, it appears that the router calculates the generic header length rather than assumes the entire payload and header.

```
ISAKMP (10): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
ISAKMP (10): Total payload length : 12
```

In the lines below, the last two exchanges (packets 5 and 6) are performed and finalize the ISA. The fact that this information is encrypted is not detailed in the debug data from R1. The HASH included in the payloads serves to authenticate both the message and the information gathered to this point. Because the ID payload is included in the exchange, it is used as part of the HASH. Other information included in the HASH was exchanged previously. A good example is that cookies are included in the HASH which were shared in the original ISAKMP header. Also included in the HASH are the DH public values that were created by the initiator and received by the responder.

Using the HASH exchanged in the last few packets validates all the peer information. Note that if the password is incorrect, the key generation is based on bad information contained in disparate databases. The result is that the final exchange is decrypted into nonsense causing several vague errors to arise.

Preshared key authentication is very popular, especially with remote access solutions, but there are several weaknesses inherent to the process. At this point, IKE Phase I is complete, and Phase II can begin to create SAs for IPSec operations.

```
ISAKMP (10): sending packet to 10.1.1.1 (I) MM_KEY_EXCH
ISAKMP (10): received packet from 10.1.1.1 (I) MM_KEY_EXCH
ISAKMP (10): processing ID payload. message ID = 0
ISAKMP (10): processing HASH payload. message ID = 0
ISAKMP (10): SA has been authenticated with 10.1.1.1
```

The section below shows that R2 has followed through with VPN establishment and has initiated Quick Mode. Note "for prot 3." Protocol 3 is the IPSec domain of interpretation (DOI) identification for IPSec ESP. This can become confusing when it is known that the protocol ID for ESP is 50, but the prot 3 designation is strictly the IPSec's DOI for the Phase II quick mode (QM) negotiation. The SA proposal from R2 that includes the transform sets is missing. In the beginning of the original IKE exchange, you see the identification of ESP, the encryption identification, and the authentication algorithm to utilize.

Note It is important to recognize that the SA payload is defined, authenticated (HASHed) and included in the first QM packet. The creation of the transform set was not part of the R2 debug. All data is encrypted using the keys derived from Phase 1.

```
ISAKMP (10): beginning Quick Mode exchange, M-ID of 953616512
IPSEC (key_engine) : got a queue event...
IPSEC (spi_response) : getting spi 413467620 for SA
from 10.1.1.1 to 10.1.1.2 for prot 3
ISAKMP (10): sending packet to 10.1.1.1 (I) QM_IDLE
ISAKMP (10): received packet from 10.1.1.1 (I) QM_IDLE
ISAKMP (10): processing SA payload. message ID = 953616512
ISAKMP (10): Checking IPsec proposal 1
ISAKMP: transform 1, ESP_DES
ISAKMP: attributes in transform:
```

```
ISAKMP: encaps is 1
ISAKMP: SA life type in seconds
ISAKMP: SA life duration (basic) of 3600
ISAKMP: SA life type in kilobytes
ISAKMP: SA life duration (VPI) of 0 x 0 0 x 46 0 x 50 0 x 0
ISAKMP: authenticator is HMAC-MD5
ISAKMP (10): atts are acceptable.
```

In the following section, the QM packets are exchanged and R1 once again validates the proposal from R2 against the SPD. The line “encap 1” specifies that the SA will use IP encapsulation; therefore, it is a tunnel mode ESP SA. Again, the lifetime is in seconds and kilobytes, which should look familiar from Phase 1. Finally, the authentication type to be used for the SA is derived.

```
ISAKMP (10): processing NONCE payload. message ID = 953616512
ISAKMP (10): processing ID payload. message ID = 953616512
```

After the proposals are validated against the SPD, the packets’ nonce and ID are processed.

```
ISAKMP (10): Creating IPsec SAs
inbound SA from 10.1.1.1 to 10.1.1.2 (proxy 10.1.0.0 to 10.1.0.0)
has spi 413467620 and conn_id 11 and flags 4
lifetime of 3600 seconds
lifetime of 4608000 kilobytes
outbound SA from 10.1.1.2 to 10.1.1.1 (proxy 10.1.0.0 to 10.1.0.0)
has spi 55772818 and conn_id 12 and flags 4
lifetime of 3600 seconds
lifetime of 4608000 kilobytes
```

The section above shows the creation of the SAs; the SPD reflects the policies in the log. The SPD and the SAD communicate to ensure that the SAD creates according to policy requirements.

```
IPSEC (key_engine) : got a queue event...
IPSEC (initialize_sas): ,
(key eng. msg.) dest = 10.1.1.2, src = 10.1.1.1,
dest_proxy = 10.1.0.0/255.255.0.0/0/0 (type = 4),
src_proxy = 10.1.0.0/255.255.0.0/0/0 (type = 4),
protocol = ESP, transform = esp-des esp-md5-hmac
lifedur = 3600s and 4608000kb
spi = 0 x 18A503E4 (413467620), conn_id = 11, keysize = 0, flags = 0 x 4
IPSEC(initialize_sas): ,
(key eng. msg.) src = 10.1.1.2, dest = 10.1.1.1,
src_proxy = 10.1.0.0/255.255.0.0/0/0 (type = 4),
dest_proxy = 10.1.0.0/255.255.0.0/0/0 (type = 4),!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/5/8 ms
R1#
protocol = ESP, transform = esp-des esp-md5-hmac,
lifedur = 3600s and 460800kb,
spi = 0 x 3530692 (55772818), conn_id = 12, keysize = 0, flags = 0 x 4
```

In the above section, the key engine receives an event to initialize an SA. The source and destinations are verified, and the SA proposals containing the necessary transform sets are compiled. These proposals are simply words in a log that reflect the creation of the ISAKMP data to be handed off to the ESP header and associated operations. As the process is completed, and the VPN is established, the results of the original ping command are displayed. The ping used to initiate the VPN is displayed. The time required for the establishment of the VPN exceeds the time taken for the first ICMP echo request. Therefore, “.!!!!,” appears, which indicates that the first echo reply was not received.

The VPN created in this environment is established on Ethernet with no other processes running on the router. The routers in the test are establishing a simple VPN over a high-speed connection. Consequently, the establishment may seem very fast, only a 5-ms average round trip that includes a 20 percent loss in packets. The process is slow. In some real-world implementations, only the second **ping** command

showed successful packets; even then the success rate was limited to the final datagrams. Consequently, this example shows that key creation and establishment of a VPN can have significant impact on system resources.

```
IPSEC (create_sa): sa created,
(sa) sa_dest = 10.1.1.2, sa_prot = 50
sa_spi = 0 x 18A503E4 (413467620),
sa_trans = esp-des esp-md5-hmac, sa_conn_id = 11
IPSEC (create_sa): sa created,
(sa) sa_dest = 10.1.1.1, sa_prot = 50
sa_spi = 0 x 3530692 (55772818),
sa_trans = esp-des esp-md5-hmac, sa_conn_id = 12
ISAKMP (10): sending packet to 10.1.1.1 (I) QM_IDLE
```

The router now displays the current SA's status. The SPIs relate to the two SAs and can be used to identify and track the activities of the respective SAs. The following is the result of a command that displays the ISAKMP SA status and allows the verification of actions within the SAs. The s**how crypto isakmp sa** command provides a detailed list of security association attributes and statistics. The command output provides not only the IKE SA information, but offers the IPSec SA data, as well, for both protocols. This command is very valuable in troubleshooting and allows the operator to determine the system configuration.

```
R1#sh crypto isakmp sa
Crypto map tag: VPN-TO-R2, local addr. 10.1.1.2
local ident (addr/mask/prot/port):
(10.1.0.0/255.255.0.0/0/0)
remote ident (addr/mask/prot/port):
(10.1.0.0/255.255.0.0/0/0)
current_peer: 10.1.1.1
PERMIT, flag = {origin_is_acl,}
#pkts encaps: 7, #pkts encrypt: 7, #pkts digest 7
#pkts decaps: 7, #pkts decrypt: 7, #pkts verify 7
#send errors 2, #recv errors 0
local crypto endpt.: 10.1.1.2, remote crypto
endpt.: 10.1.1.1
path mtu 1500, media mtu 1500
current outbound spi: 188913E4
```

Within the command output, four areas detail the existing inbound and outbound SAs. The report includes sections that identify the two possible security protocols. In this example, however, there are no AH SAs, because only the ESP security protocol was used.

```
inbound esp sas:
spi: 0 x 18A503E4 (413467620)
transform: esp-des esp-md5-hmac,
in use settings ={Tunnel,}
slot: 0, conn id: 11, crypto map: VPN-TO-R2
sa timing: remaining key lifetime (k/sec):
(4607999/3460)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
outbound esp sas:
spi: 0 x 3530692(55772818)
transform: esp-des esp-md5-hmac,
in use settings ={Tunnel,}
slot: 0, conn id: 11, crypto map: VPN-TO-R2
sa timing: remaining key lifetime (k/sec):
(4607999/3460)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
```

Sample Problem Scenarios

This appendix contains the following sample problem scenarios:

Transform Set Mismatch; Tunnel Initiated From CE1

```
TRANSFORM SET ON CE1:
crypto ipsec transform-set myset esp-des esp-md5-hmac
TRANSFORM SET ON CE2:
crypto ipsec transform-set myset ah-md5-hmac esp-des
DEBUGS ON CE1:
00:27:07: IPSEC(sa_request): ,
(key eng. msg.) src= 10.15.58.10, dest= 10.15.58.38,
src_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
dest_proxy= 192.168.2.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
spi= 0x5AE6C721(1525073697), conn_id= 0, keysize= 0, flags= 0x4004
00:27:07: ISAKMP: received ke message (1/1)
00:27:07: ISAKMP: local port 500, remote port 500
00:27:07: ISAKMP (0:2): beginning Main Mode exchange
00:27:07: ISAKMP (0:2): sending packet to 10.15.58.38 (I) MM_NO_STATE
00:27:07: ISAKMP (0:2): received packet from 10.15.58.38 (I) MM_NO_STATE
00:27:07: ISAKMP (0:2): processing SA payload. message ID = 0
00:27:07: ISAKMP (0:2): found peer pre-shared key matching 10.15.58.38
00:27:07: ISAKMP (0:2): Checking ISAKMP transform 1 against priority 10 policy
00:27:07: ISAKMP: encryption DES-CBC
00:27:07: ISAKMP: hash MD5
00:27:07: ISAKMP: default group 1
00:27:07: ISAKMP: . auth pre-share
00:27:07: ISAKMP (0:2): atts are acceptable. Next payload is 0
00:27:07: CryptoEngine0: generate alg parameter
00:27:07: CRYPTO_ENGINE: Dh phase 1 status: 0
00:27:07: CRYPTO_ENGINE: Dh phase 1 status: 0
```

```
00:27:07: ISAKMP (0:2): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
00:27:07: ISAKMP (0:2): sending packet to 10.15.58.38 (I) MM_SA_SETUP
00:27:07: ISAKMP (0:2): received packet from 10.15.58.38 (I) MM_SA_SETUP
00:27:07: ISAKMP (0:2): processing KE payload. message ID = 0
00:27:07: CryptoEngine0: generate alg parameter
00:27:07: ISAKMP (0:2): processing NONCE payload. message ID = 0
00:27:07: ISAKMP (0:2): found peer pre-shared key matching 10.15.58.38
00:27:07: CryptoEngine0: create ISAKMP SKEYID for conn id 2
00:27:07: ISAKMP (0:2): SKEYID state generated
00:27:07: ISAKMP (0:2): processing vendor id payload
00:27:07: ISAKMP (0:2): speaking to another IOS box!
00:27:07: ISAKMP (2): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
00:27:07: ISAKMP (2): Total payload length: 12
00:27:07: CryptoEngine0: generate hmac context for conn id 2
00:27:07: ISAKMP (0:2): sending packet to 10.15.58.38 (I) MM_KEY_EXCH
00:27:07: ISAKMP (0:2): received packet from 10.15.58.38 (I) MM_KEY_EXCH
00:27:07: ISAKMP (0:2): processing ID payload. message ID = 0
00:27:07: ISAKMP (0:2): processing HASH payload. message ID = 0
00:27:07: CryptoEngine0: generate hmac context for conn id 2
00:27:07: ISAKMP (0:2): SA has been authenticated with 10.15.58.38
00:27:07: ISAKMP (0:2): beginning Quick Mode exchange, M-ID of 1536042321
00:27:07: CryptoEngine0: generate hmac context for conn id 2
00:27:07: ISAKMP (0:2): sendin.g packet to 10.15.58.38 (I) QM_IDLE
00:27:07: CryptoEngine0: clear dh number for conn id 1
00:27:07: ISAKMP (0:2): received packet from 10.15.58.38 (I) QM_IDLE
00:27:07: CryptoEngine0: generate hmac context for conn id 2
```

The following messages indicate that the proposal was rejected.

```
00:27:07: ISAKMP (0:2): processing HASH payload. message ID = 403023752
00:27:07: ISAKMP (0:2): processing NOTIFY PROPOSAL_NOT_CHOSEN protocol 0
spi 0, message ID = 403023752
00:27:07: ISAKMP (0:2): deleting node 403023752 error FALSE reason "informational (in)
state
1"
00:27:07: IPSEC(key_engine): got a queue event...
00:27:07: IPSEC(key_engine_delete_sas): rec'd delete notify from ISAKMP
00:27:07: IPSEC(key_engine_delete_sas): delete all SAs shared with 10.15.58.38 ...
Success rate is 0 percent (0/5)
ce1#
DEBUGS ON CE2:
00:27:09: ISAKMP (0:0): received packet from 10.15.58.10 (N) NEW SA
00:27:09: ISAKMP: local port 500, remote port 500
00:27:09: ISAKMP (0:2): processing SA payload. message ID = 0
00:27:09: ISAKMP (0:2): found peer pre-shared key matching 10.15.58.10
00:27:09: ISAKMP (0:2): Checking ISAKMP transform 1 against priority 10 policy
00:27:09: ISAKMP: encryption DES-CBC
00:27:09: ISAKMP: hash MD5
00:27:09: ISAKMP: default group 1
00:27:09: ISAKMP: auth pre-share
00:27:09: ISAKMP (0:2): atts are acceptable. Next payload is 0
00:27:09: CryptoEngine0: generate alg parameter
00:27:10: CRYPTO_ENGINE: Dh phase 1 status: 0
00:27:10: CRYPTO_ENGINE: Dh phase 1 status: 0
00:27:10: ISAKMP (0:2): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
00:27:10: ISAKMP (0:2): sending packet to 10.15.58.10 (R) MM_SA_SETUP
00:27:10: ISAKMP (0:2): received packet from 10.15.58.10 (R) MM_SA_SETUP
```

```
00:27:10: ISAKMP (0:2): processing KE payload. message ID = 0
00:27:10: CryptoEngine0: generate alg parameter
00:27:10: ISAKMP (0:2): processing NONCE payload. message ID = 0
00:27:10: ISAKMP (0:2): found peer pre-shared key matching 10.15.58.10
00:27:10: CryptoEngine0: create ISAKMP SKEYID for conn id 2
00:27:10: ISAKMP (0:2): SKEYID state generated
00:27:10: ISAKMP (0:2): processing vendor id payload
00:27:10: ISAKMP (0:2): speaking to another IOS box!
00:27:10: ISAKMP (0:2): sending packet to 10.15.58.10 (R) MM_KEY_EXCH
00:27:10: ISAKMP (0:2): received packet from 10.15.58.10 (R) MM_KEY_EXCH
00:27:10: ISAKMP (0:2): processing ID payload. message ID = 0
00:27:10: ISAKMP (0:2): processing HASH payload. message ID = 0
00:27:10: CryptoEngine0: generate hmac context for conn id 2
00:27:10: ISAKMP (0:2): SA has been authenticated with 10.15.58.10
00:27:10: ISAKMP (2): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
00:27:10: ISAKMP (2): Total payload length: 12
00:27:10: CryptoEngine0: generate hmac context for conn id 2
00:27:10: CryptoEngine0: clear dh number for conn id 1
00:27:10: ISAKMP (0:2): sending packet to 10.15.58.10 (R) QM_IDLE
00:27:10: ISAKMP (0:2): received packet from 10.15.58.10 (R) QM_IDLE
00:27:10: CryptoEngine0: generate hmac context for conn id 2
00:27:10: ISAKMP (0:2): processing HASH payload. message ID = 1536042321
00:27:10: ISAKMP (0:2): processing SA payload. message ID = 1536042321
00:27:10: ISAKMP (0:2): Checking IPSec proposal 1
00:27:10: ISAKMP: transform 1, ESP_DES
00:27:10: ISAKMP: attributes in transform:
00:27:10: ISAKMP: encaps is 1
00:27:10: ISAKMP: SA life type in seconds
00:27:10: ISAKMP: SA life duration (basic) of 3600
00:27:10: ISAKMP: SA life type in kilobytes
00:27:10: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
00:27:10: ISAKMP: authenticator is HMAC-MD5
```

The output from CE2 indicates the reason for sending PROPOSAL_NOT_CHOSEN. (The transform proposal did not match the proposal defined.)

```
00:27:10: validate proposal 0
00:27:10: IPSEC(validate_proposal): transform proposal (prot 3, trans 2, hmac_alg 1) not
supported
00:27:10: ISAKMP (0:2): atts not acceptable. Next payload is 0
00:27:10: ISAKMP (0:2): phase 2 SA not acceptable!
00:27:10: CryptoEngine0: generate hmac context for conn id 2
00:27:10: ISAKMP (0:2): sending packet to 10.15.58.10 (R) QM_IDLE
00:27:10: ISAKMP (0:2): purging node 403023752
00:27:10: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Quick mode failed with peer at
10.15.58.10
00:27:10: ISAKMP (0:2): deleting node 1536042321 error FALSE reason
"IKMP_NO_ERR_NO_TRANS"
```

Access-list Mismatch; Tunnel Initiated From CE1

```
ACCESS-LIST ON CE1:
Extended IP access list 100
permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
ACCESS-LIST ON CE2:
```

```
Extended IP access list 100
permit ip 192.168.2.0 0.0.0.255 host 192.168.1.1
```

Debug on CE1:

```
00:39:17: IPSEC(sa_request): ,
(key eng. msg.) src= 10.15.58.10, dest= 10.15.58.38,
src_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
dest_proxy= 192.168.2.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
spi= 0x86F18EB(141498603), conn_id= 0, keysize= 0, flags= 0x4004
00:39:17: ISAKMP: received ke message (1/1)
00:39:17: ISAKMP: local port 500, remote port 500
00:39:17: ISAKMP (0:1): beginning Main Mode exchange
00:39:17: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_NO_STATE
00:39:17: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_NO_STATE
00:39:17: ISAKMP (0:1): processing SA payload. message ID = 0
00:39:17: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.38
00:39:17: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 10 policy
00:39:17: ISAKMP: encryption DES-CBC
00:39:17: ISAKMP: hash MD5
00:39:17: ISAKMP: default group 1
00:39:17: ISAKMP: auth pre-share
00:39:17: ISAKMP (0:1): atts are acceptable. Next payload is 0
00:39:17: CryptoEngine0: generate alg parameter
00:39:18: CRYPTO_ENGINE: Dh phase 1 status: 0
00:39:18: CRYPTO_ENGINE: Dh phase 1 status: 0
00:39:18: ISAKMP (0:1): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
00:39:18: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_SA_SETUP
00:39:18: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_SA_SETUP
00:39:18: ISAKMP (0:1): processing KE payload. message ID = 0
00:39:18: CryptoEngine0: generate alg parameter
00:39:18: ISAKMP (0:1): processing NONCE payload. message ID = 0
00:39:18: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.38
00:39:18: CryptoEngine0: create ISAKMP SKEYID for conn id 1
00:39:18: ISAKMP (0:1): SKEYID state generated
00:39:18: ISAKMP (0:1): processing vendor i.d payload
00:39:18: ISAKMP (0:1): speaking to another IOS box!
00:39:18: ISAKMP (1): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
00:39:18: ISAKMP (1): Total payload length: 12
00:39:18: CryptoEngine0: generate hmac context for conn id 1
00:39:18: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_KEY_EXCH
00:39:18: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_KEY_EXCH
00:39:18: ISAKMP (0:1): processing ID payload. message ID = 0
00:39:18: ISAKMP (0:1): processing HASH payload. message ID = 0
00:39:18: CryptoEngine0: generate hmac context for conn id 1
00:39:18: ISAKMP (0:1): SA has been authenticated with 10.15.58.38
00:39:18: ISAKMP (0:1): beginning Quick Mode exchange, M-ID of -537199713
00:39:18: CryptoEngine0: generate hmac context for conn id 1
00:39:18: ISAKMP (0:1): sending packet to 10.15.58.38 (I) QM_IDLE
00:39:18: CryptoEngine0: clear dh number for conn id 1
00:39:18: ISAKMP (0:1): received packet from 10.15.58.38 (I) QM_IDLE
00:39:18: CryptoEngine0: generate hmac context for conn id 1
```

The initiator side (CE1) produced the error message, but you must look at the receiving side (CE2) for details:

```
00:39:18: ISAKMP (0:1): processing HASH payload. message ID = 400363004
00:39:18: ISAKMP (0:1): processing NOTIFY PROPOSAL_NOT_CHOSEN protocol 3
spi 141498603, message ID = 400363004
00:39:18: ISAKMP (0:1): deleting spi 141498603 message ID = -537199713
00:39:18: ISAKMP (0:1): deleting node -537199713 error TRUE reason "delete_lar
val"
00:39:18: ISAKMP (0:1): deleting node 400363004 error FALSE reason "informational (in)
state 1"....
```

Debug on CE2:

```
00:39:20: ISAKMP (0:0): received packet from 10.15.58.10 (N) NEW SA
00:39:20: ISAKMP: local port 500, remote port 500
00:39:20: ISAKMP (0:1): processing SA payload. message ID = 0
00:39:20: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.10
00:39:20: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 10 policy
00:39:20: ISAKMP: encryption DES-CBC
00:39:20: ISAKMP: hash MD5
00:39:20: ISAKMP: default group 1
00:39:20: ISAKMP: auth pre-share
00:39:20: ISAKMP (0:1): atts are acceptable. Next payload is 0
00:39:20: CryptoEngine0: generate alg parameter
00:39:20: CRYPTO_ENGINE: Dh phase 1 status: 0
00:39:20: CRYPTO_ENGINE: Dh phase 1 status: 0
00:39:20: ISAKMP (0:1): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
00:39:20: ISAKMP (0:1): sending packet to 10.15.58.10 (R) MM_SA_SETUP
00:39:20: ISAKMP (0:1): received packet from 10.15.58.10 (R) MM_SA_SETUP
00:39:20: ISAKMP (0:1): processing KE payload. message ID = 0
00:39:20: CryptoEngine0: generate alg parameter
00:39:21: ISAKMP (0:1): processing NONCE payload. message ID = 0
00:39:21: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.10
00:39:21: CryptoEngine0: create ISAKMP SKEYID for conn id 1
00:39:21: ISAKMP (0:1): SKEYID state generated
00:39:21: ISAKMP (0:1): processing vendor id payload
00:39:21: ISAKMP (0:1): speaking to another IOS box!
00:39:21: ISAKMP (0:1): sending packet to 10.15.58.10 (R) MM_KEY_EXCH
00:39:21: ISAKMP (0:1): received packet from 10.15.58.10 (R) MM_KEY_EXCH
00:39:21: ISAKMP (0:1): processing ID payload. message ID = 0
00:39:21: ISAKMP (0:1): processing HASH payload. message ID = 0
00:39:21: CryptoEngine0: generate hmac context for conn id 1
00:39:21: ISAKMP (0:1): SA has been authenticated with 10.15.58.10
00:39:21: ISAKMP (1): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
00:39:21: ISAKMP (1): Total payload length: 12
00:39:21: CryptoEngine0: generate hmac context for conn id 1
00:39:21: CryptoEngine0: clear dh number for conn id 1
00:39:21: ISAKMP (0:1): sending packet to 10.15.58.10 (R) QM_IDLE
00:39:21: ISAKMP (0:1): received packet from 10.15.58.10 (R) QM_IDLE
00:39:21: CryptoEngine0: generate hmac context for conn id 1
00:39:21: ISAKMP (0:1): processing HASH payload. message ID = -537199713
00:39:21: ISAKMP (0:1): processing SA payload. message ID = -537199713
00:39:21: ISAKMP (0:1): Checking IPSec proposal 1
00:39:21: ISAKMP: transform 1, ESP_DES
00:39:21: ISAKMP: attributes in transform:
00:39:21: ISAKMP: encaps is 1
00:39:21: ISAKMP: SA life type in seconds
```

```
00:39:21: ISAKMP: SA life duration (basic) of 3600
00:39:21: ISAKMP: SA life type in kilobytes
00:39:21: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
00:39:21: ISAKMP: authenticator is HMAC-MD5
00:39:21: validate proposal 0
00:39:21: ISAKMP (0:1): atts are acceptable.
```

In the following output, the errror message "proxy identities not supported" indicates that the access-lists did not match on both sides.

```
00:39:21: IPSEC(validate_proposal_request): proposal part #1,
(key eng. msg.) dest= 10.15.58.38, src= 10.15.58.10,
dest_proxy= 192.168.2.0/255.255.255.0/0/0 (type=4),
src_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-des esp-md5-hmac,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
00:39:21: validate proposal request 0
00:39:21: IPSEC(validate_transform_proposal): proxy identities not supported
00:39:21: ISAKMP (0:1): IPSec policy invalidated proposal
00:39:21: ISAKMP (0:1): phase 2 SA not acceptable!
00:39:21: CryptoEngine0: generate hmac context for conn id 1
00:39:21: ISAKMP (0:1): sending packet to 10.15.58.10 (R) QM_IDLE
00:39:21: ISAKMP (0:1): purging node 400363004
00:39:21: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Quick mode failed with peer at
10.15.58.10
00:39:21: ISAKMP (0:1): deleting node -537199713 error FALSE reason
"IKMP_NO_ERR_NO_TRANS"
CE2#
```

Key Mismatch; Tunnel Initiated From CE1

```
KEY ON CE1:
crypto isakmp key cisco123 address 10.15.58.38
KEY ON CE2:
crypto isakmp key mykey address 10.15.58.10
DEBUG ON CE1:
00:57:36: IPSEC(sa_request): ,
(key eng. msg.) src= 10.15.58.10, dest= 10.15.58.38,
src_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
dest_proxy= 192.168.2.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
spi= 0x278A45B5(663373237), conn_id= 0, keysize= 0, flags= 0x4004
00:57:36: ISAKMP: received ke message (1/1)
00:57:36: ISAKMP: local port 500, remote port 500
00:57:36: ISAKMP (0:1): beginning Main Mode exchange
00:57:36: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_NO_STATE
00:57:36: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_NO_STATE
00:57:36: ISAKMP (0:1): processing SA payload. message ID = 0
00:57:36: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.38
00:57:36: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 10 policy
00:57:36: ISAKMP: encryption DES-CBC
00:57:36: ISAKMP: hash MD5
00:57:36: ISAKMP: default group 1
00:57:36: ISAKMP: .
Success rate is 0 percent (0/1)
ce1# auth pre-share
00:57:36: ISAKMP (0:1): atts are acceptable. Next payload is 0
00:57:36: CryptoEngine0: generate alg parameter
00:57:36: CRYPTO_ENGINE: Dh phase 1 status: 0
```

```
00:57:36: CRYPTO_ENGINE: Dh phase 1 status: 0
00:57:36: ISAKMP (0:1): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
00:57:36: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_SA_SETUP
00:57:36: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_SA_SETUP
00:57:36: ISAKMP (0:1): processing KE payload. message ID = 0
00:57:36: CryptoEngine0: generate alg parameter
00:57:37: ISAKMP (0:1): processing NONCE payload. message ID = 0
00:57:37: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.38
00:57:37: CryptoEngine0: create ISAKMP SKEYID for conn id 1
00:57:37: ISAKMP (0:1): SKEYID state generated
00:57:37: ISAKMP (0:1): processing vendor id payload
00:57:37: ISAKMP (0:1): speaking to another IOS box!
00:57:37: ISAKMP (1): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
00:57:37: ISAKMP (1): Total payload length: 12
00:57:37: CryptoEngine0: generate hmac context for conn id 1
00:57:37: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_KEY_EXCH
00:57:37: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_KEY_EXCH
```

The following message indicates that the packet could not be verified because the keys do not match.

```
00:57:37: ISAKMP: reserved not zero on NOTIFY payload!
00:57:37: %CRYPTO-4-IKMP_BAD_MESSAGE: IKE message from 10.15.58.38 failed its sanity
check or is malformed
DEBUGS OB CE2:
CE2#
00:54:40: ISAKMP (0:0): received packet from 10.15.58.10 (N) NEW SA
00:54:40: ISAKMP: local port 500, remote port 500
00:54:40: ISAKMP (0:1): processing SA payload. message ID = 0
00:54:40: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.10
00:54:40: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 10 policy
00:54:40: ISAKMP: encryption DES-CBC
00:54:40: ISAKMP: hash MD5
00:54:40: ISAKMP: default group 1
00:54:40: ISAKMP: auth pre-share
00:54:40: ISAKMP (0:1): atts are acceptable. Next payload is 0
00:54:40: CryptoEngine0: generate alg parameter
00:54:40: CRYPTO_ENGINE: Dh phase 1 status: 0
00:54:40: CRYPTO_ENGINE: Dh phase 1 status: 0
00:54:40: ISAKMP (0:1): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
00:54:40: ISAKMP (0:1): sending packet to 10.15.58.10 (R) MM_SA_SETUP
00:54:40: ISAKMP (0:1): received packet from 10.15.58.10 (R) MM_SA_SETUP
00:54:40: ISAKMP (0:1): processing KE payload. message ID = 0
00:54:40: CryptoEngine0: generate alg parameter
00:54:40: ISAKMP (0:1): processing NONCE payload. message ID = 0
00:54:40: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.10
00:54:40: CryptoEngine0: create ISAKMP SKEYID for conn id 1
00:54:40: ISAKMP (0:1): SKEYID state generated
00:54:40: ISAKMP (0:1): processing vendor id payload
00:54:40: ISAKMP (0:1): speaking to another IOS box!
00:54:40: ISAKMP (0:1): sending packet to 10.15.58.10 (R) MM_KEY_EXCH
```

A keyed mismatch problem on the receiving side provides these messages:

```
00:54:40: ISAKMP (0:1): received packet from 10.15.58.10 (R) MM_KEY_EXCH
00:54:40: ISAKMP: reserved not zero on ID payload!
00:54:40: %CRYPTO-4-IKMP_BAD_MESSAGE: IKE message from 10.15.58.10 failed its sanity
check or is malformed
```

```
00:54:40: ISAKMP (0:1): incrementing error counter on sa: PAYLOAD_MALFORMED
```

ISAKMP Policy Mismatch; Tunnel Initiated From CE1

```
POLICY ON CE1:
Protection suite of priority 10
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Message Digest 5
authentication method: Pre-Shared Key
Diffie-Hellman group: #1 (768 bit)
lifetime: 86400 seconds, no volume limit
POLICY ON CE2:
Protection suite of priority 10
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Secure Hash Standard
authentication method: Pre-Shared Key
Diffie-Hellman group: #1 (768 bit)
lifetime: 86400 seconds, no volume limit
DEBUG ON CE1:
01:21:24: IPSEC(sa_request): ,
(key eng. msg.) src= 10.15.58.10, dest= 10.15.58.38,
src_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
dest_proxy= 192.168.2.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
spi= 0x3C8A0F6E(1015680878), conn_id= 0, keysize= 0, flags= 0x4004
01:21:24: ISAKMP: received ke message (1/1)
01:21:24: ISAKMP: local port 500, remote port 500
01:21:24: ISAKMP (0:1): beginning Main Mode exchange
01:21:24: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_NO_STATE
01:21:24: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_NO_STATE
01:21:24: ISAKMP (0:1): Notify has no hash. Rejected.
```

The output that follows from CE1 indicates that the Phase 1 exchange failed:

```
01:21:24: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Informational mode failed with peer
at 10.15.58.38 .
DEBUG ON CE2:
CE2#
01:21:27: ISAKMP (0:0): received packet from 10.15.58.10 (N) NEW SA
01:21:27: ISAKMP: local port 500, remote port 500
01:21:27: ISAKMP (0:1): processing SA payload. message ID = 0
01:21:27: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.10
01:21:27: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 10 policy
01:21:27: ISAKMP: encryption DES-CBC
01:21:27: ISAKMP: hash MD5
01:21:27: ISAKMP: default group 1
01:21:27: ISAKMP: auth pre-share
01:21:27: ISAKMP (0:1): atts are not acceptable. Next payload is 0
01:21:27: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 65535 policy
01:21:27: ISAKMP: encryption DES-CBC
01:21:27: ISAKMP: hash MD5
01:21:27: ISAKMP: default group 1
01:21:27: ISAKMP: auth pre-share
```

The messages below indicate that the Phase 1 negotiation failed. The received attributes (above) should be checked against the defined policy. The same debugs appear if encryption type (des/3des) or authentication type (preshare/RSA) are mismatched.

```
01:21:27: ISAKMP (0:1): atts are not acceptable. Next payload is 0
01:21:27: ISAKMP (0:1): no offers accepted!
```

```
01:21:27: ISAKMP (0:1): phase 1 SA not acceptable!
01:21:27: ISAKMP (0:1): incrementing error counter on sa: construct_fail_ag_init
01:21:27: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main mode failed with peer at
10.15.58.10
01:21:27: ISAKMP (0:1): sending packet to 10.15.58.10 (R) MM_NO_STATE
```

Crypto Map Not Applied; Tunnel Initiated from CE1

Crypto Map Not Applied on CE1

The tunnel will not try to come up. Issuing sh access-list will indicate that the matches do not increase.

Crypto Map Not Applied on CE2

If the receiving side does not have a map applied, the initiator starts the tunnel but the tunnel does not come up.

Debug on CE1

```
01:33:45: IPSEC(sa_request): ,
(key eng. msg.) src= 10.15.58.10, dest= 10.15.58.38,
src_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
dest_proxy= 192.168.2.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 3600s and 4608000kb,
spi= 0xA9B9C3A3(2847523747), conn_id= 0, keysize= 0, flags= 0x4004
01:33:45: ISAKMP: received ke message (1/1)
01:33:45: ISAKMP: local port 500, remote port 500
01:33:45: ISAKMP (0:1): beginning Main Mode exchange
01:33:45: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_NO_STATE
01:33:45: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_NO_STATE
01:33:45: ISAKMP (0:1): processing SA payload. message ID = 0
01:33:45: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.38
01:33:45: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 10 policy
01:33:45: ISAKMP: encryption DES-CBC
01:33:45: ISAKMP: hash MD5
01:33:45: ISAKMP: default group 1
01:33:45: ISAKMP: .
Success rate is 0 percent (0/1)
ce1# auth pre-share
01:33:45: ISAKMP (0:1): atts are acceptable. Next payload is 0
01:33:45: CryptoEngine0: generate alg parameter
01:33:45: CRYPTO_ENGINE: Dh phase 1 status: 0
01:33:45: CRYPTO_ENGINE: Dh phase 1 status: 0
01:33:45: ISAKMP (0:1): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
01:33:45: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_SA_SETUP
01:33:45: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_SA_SETUP
01:33:45: ISAKMP (0:1): processing KE payload. message ID = 0
01:33:45: CryptoEngine0: generate alg parameter
01:33:45: ISAKMP (0:1): processing NONCE payload. message ID = 0
01:33:45: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.38
01:33:45: CryptoEngine0: create ISAKMP SKEYID for conn id 1
01:33:45: ISAKMP (0:1): SKEYID state generated
```

```
01:33:45: ISAKMP (0:1): processing vendor id payload
01:33:45: ISAKMP (0:1): speaking to another IOS box!
01:33:45: ISAKMP (1): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
01:33:45: ISAKMP (1): Total payload length: 12
01:33:45: CryptoEngine0: generate hmac context for conn id 1
01:33:45: ISAKMP (0:1): sending packet to 10.15.58.38 (I) MM_KEY_EXCH
01:33:45: ISAKMP (0:1): received packet from 10.15.58.38 (I) MM_KEY_EXCH
01:33:45: ISAKMP (0:1): processing ID payload. message ID = 0
01:33:45: ISAKMP (0:1): processing HASH payload. message ID = 0
01:33:45: CryptoEngine0: generate hmac context for conn id 1
01:33:45: ISAKMP (0:1): SA has been authenticated with 10.15.58.38
01:33:45: ISAKMP (0:1): beginning Quick Mode exchange, M-ID of -263684175
01:33:45: CryptoEngine0: generate hmac context for conn id 1
01:33:45: ISAKMP (0:1): sending packet to 10.15.58.38 (I) QM_IDLE
01:33:45: CryptoEngine0: clear dh number for conn id 1
01:33:45: ISAKMP (0:1): received packet from 10.15.58.38 (I) QM_IDLE
01:33:45: CryptoEngine0: generate hmac context for conn id 1
01:33:45: ISAKMP (0:1): processing HASH payload. message ID = 2084297188
```

An error from the receiving side appears on CE1. Look at the other side (CE2) to determine the problem.

```
01:33:45: ISAKMP (0:1): processing NOTIFY PROPOSAL_NOT_CHOSEN protocol 0
spi 0, message ID = 2084297188
01:33:45: ISAKMP (0:1): deleting node 2084297188 error FALSE reason "information"
DEBUGS FROM CE2:
01:33:47: ISAKMP (0:0): received packet from 10.15.58.10 (N) NEW SA
01:33:47: ISAKMP: local port 500, remote port 500
01:33:47: ISAKMP (0:1): processing SA payload. message ID = 0
01:33:47: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.10
01:33:47: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 10 policy
01:33:47: ISAKMP: encryption DES-CBC
01:33:47: ISAKMP: hash MD5
01:33:47: ISAKMP: default group 1
01:33:47: ISAKMP: auth pre-share
01:33:47: ISAKMP (0:1): atts are acceptable. Next payload is 0
01:33:47: CryptoEngine0: generate alg parameter
01:33:48: CRYPTO_ENGINE: Dh phase 1 status: 0
01:33:48: CRYPTO_ENGINE: Dh phase 1 status: 0
01:33:48: ISAKMP (0:1): SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
01:33:48: ISAKMP (0:1): sending packet to 10.15.58.10 (R) MM_SA_SETUP
01:33:48: ISAKMP (0:1): received packet from 10.15.58.10 (R) MM_SA_SETUP
01:33:48: ISAKMP (0:1): processing KE payload. message ID = 0
01:33:48: CryptoEngine0: generate alg parameter
01:33:48: ISAKMP (0:1): processing NONCE payload. message ID = 0
01:33:48: ISAKMP (0:1): found peer pre-shared key matching 10.15.58.10
01:33:48: CryptoEngine0: create ISAKMP SKEYID for conn id 1
01:33:48: ISAKMP (0:1): SKEYID state generated
01:33:48: ISAKMP (0:1): processing vendor id payload
01:33:48: ISAKMP (0:1): speaking to another IOS box!
01:33:48: ISAKMP (0:1): sending packet to 10.15.58.10 (R) MM_KEY_EXCH
01:33:48: ISAKMP (0:1): received packet from 10.15.58.10 (R) MM_KEY_EXCH
01:33:48: ISAKMP (0:1): processing ID payload. message ID = 0
01:33:48: ISAKMP (0:1): processing HASH payload. message ID = 0
01:33:48: CryptoEngine0: generate hmac context for conn id 1
01:33:48: ISAKMP (0:1): SA has been authenticated with 10.15.58.10
01:33:48: ISAKMP (1): ID payload
next-payload : 8
type : 1
```

```
protocol : 17
port : 500
length : 8
01:33:48: ISAKMP (1): Total payload length: 12
01:33:48: CryptoEngine0: generate hmac context for conn id 1
01:33:48: CryptoEngine0: clear dh number for conn id 1
01:33:48: ISAKMP (0:1): sending packet to 10.15.58.10 (R) QM_IDLE
01:33:48: ISAKMP (0:1): received packet from 10.15.58.10 (R) QM_IDLE
01:33:48: CryptoEngine0: generate hmac context for conn id 1
01:33:48: ISAKMP (0:1): processing HASH payload. message ID = -263684175
01:33:48: ISAKMP (0:1): processing SA payload. message ID = -263684175
01:33:48: ISAKMP (0:1): Checking IPSec proposal 1
01:33:48: ISAKMP: transform 1, ESP_DES
01:33:48: ISAKMP: attributes in transform:
01:33:48: ISAKMP: encaps is 1
01:33:48: ISAKMP: SA life type in seconds
01:33:48: ISAKMP: SA life duration (basic) of 3600
01:33:48: ISAKMP: SA life type in kilobytes
01:33:48: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
01:33:48: ISAKMP: authenticator is HMAC-MD5
01:33:48: validate proposal 0
```

Phase II failed. The message Invalid local address below means that the packet entered an interface where the map was not applied and thus Phase II could not be negotiated.

```
01:33:48: IPSEC(validate_proposal): invalid local address 10.15.58.38
01:33:48: ISAKMP (0:1): atts not acceptable. Next payload is 0
01:33:48: ISAKMP (0:1): phase 2 SA not acceptable!
01:33:48: CryptoEngine0: generate hmac context for conn id 1
01:33:48: ISAKMP (0:1): sending packet to 10.15.58.10 (R) QM_IDLE
01:33:48: ISAKMP (0:1): purging node 2084297188
01:33:48: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Quick mode failed with peer at
10.15.58.10
01:33:48: ISAKMP (0:1): deleting node -263684175 error FALSE reason
"IKMP_NO_ERR_NO_TRANS"
```

Missing SAs

CE2 has rebooted and no longer has SAs. CE1, however, still has SAs. If traffic is being sent from CE1, the following message appears on CE2:

```
01:51:48: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
destaddr=10.15.58.38, prot=50, spi=0x3ACB80F(61650959)
```

Transform and Proposal Mismatch

The following was entered on the Cisco 7100 router:

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

The following was entered on the Cisco 7200 router:

```
:
crypto ipsec transform-set isaTransform esp-3des esp-md5-hmac
7100-UUT#
1d03h: ISAKMP (0:0): received packet from 193.168.1.1 (N) NEW SA
1d03h: ISAKMP: local port 500, remote port 500 ---------------------- (THIS IS THE PORT
IKE USES)
1d03h: ISAKMP (0:1): processing SA payload. message ID = 0
```

```
1d03h: ISAKMP (0:1): found peer pre-shared key matching 193.168.1.1
1d03h: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 1 policy
1d03h: ISAKMP: encryption 3DES-CBC ---- THESE ARE THE ISAKMP TRANSFORMS+
PROPOSALS FOR MAIN MODE
1d03h: ISAKMP: hash SHA
1d03h: ISAKMP: default group 2
1d03h: ISAKMP: auth pre-share
1d03h: ISAKMP (0:1): atts are acceptable. Next payload is 0
1d03h: ISAKMP (0:1): SA is doing pre-shared key authentication using id type ID_IPV4_ADDR
1d03h: ISAKMP (0:1): sending packet to 193.168.1.1 (R) MM_SA_SETUP ------- MAIN MODE
SECURITY ASSOCIATION SETUP
1d03h: ISAKMP (0:1): received packet from 193.168.1.1 (R) MM_SA_SETUP
1d03h: ISAKMP (0:1): processing KE payload. message ID = 0
1d03h: ISAKMP (0:1): processing NONCE payload. message ID = 0
1d03h: ISAKMP (0:1): found peer pre-shared key matching 193.168.1.1
1d03h: ISAKMP (0:1): SKEYID state generated
1d03h: ISAKMP (0:1): processing vendor id payload
1d03h: ISAKMP (0:1): speaking to another IOS box!
1d03h: ISAKMP (0:1): sending packet to 193.168.1.1 (R) MM_KEY_EXCH ------- MAIN MODE KEY
EXCHANGE BEING DONE.
1d03h: ISAKMP (0:1): received packet from 193.168.1.1 (R) MM_KEY_EXCH
1d03h: ISAKMP (0:1): processing ID payload. message ID = 0
1d03h: ISAKMP (0:1): processing HASH payload. message ID = 0
1d03h: ISAKMP (0:1): SA has been authenticated with 193.168.1.1
1d03h: ISAKMP (1): ID payload
next-payload : 8
type : 1
protocol : 17
port : 500
length : 8
1d03h: ISAKMP (1): Total payload length: 12
1d03h: ISAKMP (0:1): sending packet to 193.168.1.1 (R) QM_IDLE ------ MAIN MODE IS DONE
AND MOVING TO QUICK MODE
1d03h: ISAKMP (0:1): received packet from 193.168.1.1 (R) QM_IDLE
1d03h: ISAKMP (0:1): processing HASH payload. message ID = -2037037709
1d03h: ISAKMP (0:1): processing SA payload. message ID = -2037037709
1d03h: ISAKMP (0:1): Checking IPSec proposal 1
1d03h: ISAKMP: transform 1, ESP_3DES --- THESE ARE THE TRANSFORMS AND PROPOSALS
SENT BY THE PEER
1d03h: ISAKMP: attributes in transform:
1d03h: ISAKMP: encaps is 1
1d03h: ISAKMP: SA life type in seconds
1d03h: ISAKMP: SA life duration (basic) of 3600
1d03h: ISAKMP: SA life type in kilobytes
1d03h: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
1d03h: ISAKMP: authenticator is HMAC-MD5 ------------- USING HASH-MD5
1d03h: ISAKMP (0:1): atts not acceptable. Next payload is 0 ------ THE PROPOSAL IS
REJECTED
1d03h: ISAKMP (0:1): phase 2 SA not acceptable! ------------------- QUCIK MODE OR PHASE 2
IS
REJECTED.
1d03h: ISAKMP (0:1): sending packet to 193.168.1.1 (R) QM_IDLE
1d03h: ISAKMP (0:1): purging node 2061107598
1d03h: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Quick mode failed with peer at
193.168.1.1
1d03h: ISAKMP (0:1): deleting node -2037037709 error FALSE reason
"IKMP_NO_ERR_NO_TRANS"
7100-UUT#
7100-UUT#
1d03h: ISAKMP (0:1): received packet from 193.168.1.1 (R) QM_IDLE
1d03h: ISAKMP (0:1): phase 2 packet is a duplicate of a previous packet.
1d03h: ISAKMP (0:1): retransmitting due to retransmit phase 2
1d03h: ISAKMP (0:1): ignoring retransmission,because phase2 node marked dead -2037037709
```

All debugging has been turned off. Issue the **show cry isa sa** command.

```
7100-UUT#show cry isa sa
dst src state conn-id slot
195.168.1.1 193.168.1.1 QM_IDLE 1 0
7100-UUT#show cry isa sa
dst src state conn-id slot
```

A

B

C

D

F

H

I

K

M

N

P

Q

R

S

T

U

W