CISCO NETWORKING ACADEMY PROGRAM

CCNP:

Implementing Secure Converged Wide-area Networks v5.0

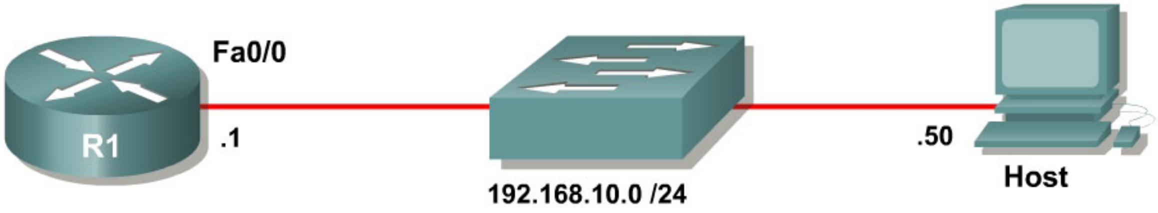
Student Lab Manual

Lab 3.1 Configuring SDM on a Router

Learning Objectives

- Prepare a router for access with Cisco Security Device Manager
- Install SDM onto a PC
- Install SDM onto a router through a Windows host

Topology Diagram



Scenario

In this lab, you will prepare a router for access via the Cisco Security Device Manager (SDM), using some basic commands, to allow connectivity from the SDM to the router. You will then install the SDM application locally on a host computer. Finally, you will install SDM onto the flash memory of a router.

Step 1: Lab Preparation

Start this lab by erasing any previous configurations and reloading your devices. Once your devices are reloaded, set the appropriate hostnames. Ensure that the switch is set up so that both the router and host are in the same VLAN. By default, all ports on the switch are assigned to VLAN 1.

Ensure that your PC meets the minimum requirements to support SDM. SDM can be run on a PC running any of the following operating systems:

- Microsoft Windows ME
- Microsoft Windows NT 4.0 Workstation with Service Pack 4
- Microsoft Windows XP Professional
- Microsoft Windows 2003 Server (Standard Edition)
- Microsoft Windows 2000 Professional with Service Pack 4

Note: Windows 2000 Advanced Server is not supported.

In addition, a web browser with SUN JRE 1.4 or later or an ActiveX controlled browser must be enabled.

Step 2: Prepare the Router for SDM

The Cisco SDM application uses the virtual terminal lines and HTTP server to manipulate the configuration of the device. Since a user must log in to access or change the configuration, some basic commands must be issued to allow remote access.

These are basic IOS commands and are not SDM-specific. However, without these commands, SDM will not be able to access the router, and will not work properly.

First, create a username and password on the router for SDM to use. This login will need to have a privilege level of 15 so that SDM can change configuration settings on the router. Make the password argument of this command the last argument on the line, since everything after the password argument will become part of the password. The username and password combination will be used later when accessing the router.

```
R1(config)# username ciscosdm privilege 15 password 0 ciscosdm
```

HTTP access to the router must be configured for SDM to work. If your image supports it (you will need to have an IOS image that supports crypto functionality), you should also enable secure HTTPS access using the **ip http secure-server** command. Enabling HTTPS generates some output about RSA encryption keys. This is normal. Also, make sure the HTTP server uses the local database for authentication purposes.

```
R1(config)# ip http server
R1(config)# ip http secure-server
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Jan 14 20:19:45.310: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Jan 14 20:19:46.406: %PKI-4-NOAUTOSAVE: Configuration was modified.  Issue
"write memory" to save new certificate
R1(config)# ip http authentication local
```

Finally, configure the virtual terminal lines of the router to authenticate using the local authentication database. Allow virtual terminal input through both telnet and SSH.

```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input telnet ssh
```

Based on your knowledge of SDM, why do you think that the router needs to have these non-SDM specific commands entered in?

Step 3: Configure Addressing

Now that the router has all of the commands necessary for remote access, connectivity will need to be established between the PC and the router. The first thing we will need to do is configure the Fast Ethernet interface on the router with the IP address shown in the diagram. If you have already configured the correct IP address, skip this step.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
```

Next, assign an IP address to the PC. If the PC already has an IP address in the same subnet as the router, you may skip this step. These steps may vary depending on your Windows version and theme.

First, access the PC Control Panel window and open the Network Connections management interface.

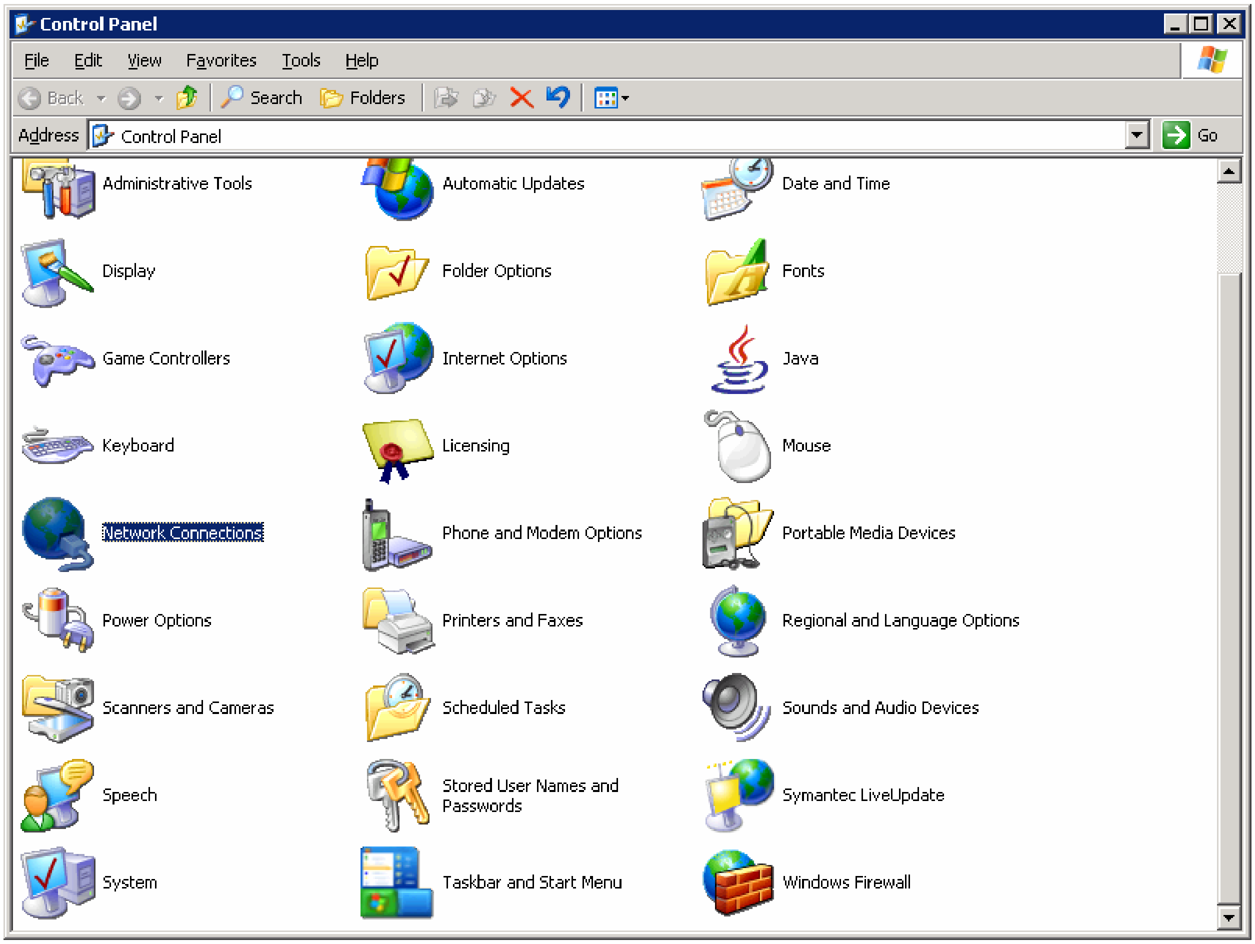
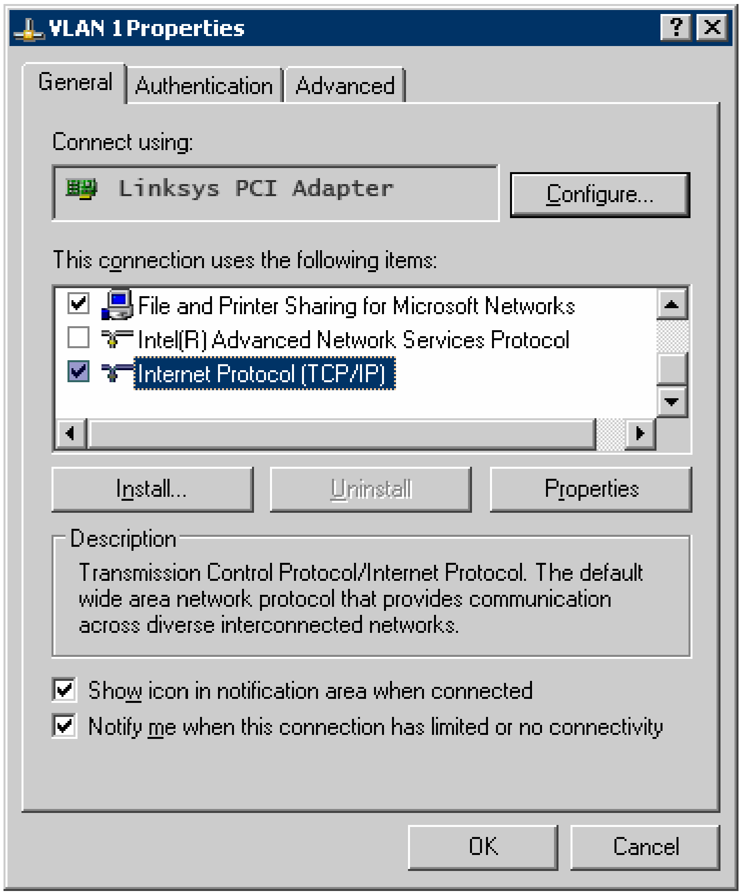


Figure 3-1: Microsoft Windows Control Panel

Right-click the LAN interface that connects to the Catalyst switch and click **Properties**. Choose **Internet Protocol (TCP/IP),** and then click the **Properties** button.



Connect using:

Linksys PCI Adapter

This connection uses the following items:

- ☑ File and Printer Sharing for Microsoft Networks
- ☐ Intel(R) Advanced Network Services Protocol
- ☑ Internet Protocol (TCP/IP)

Description

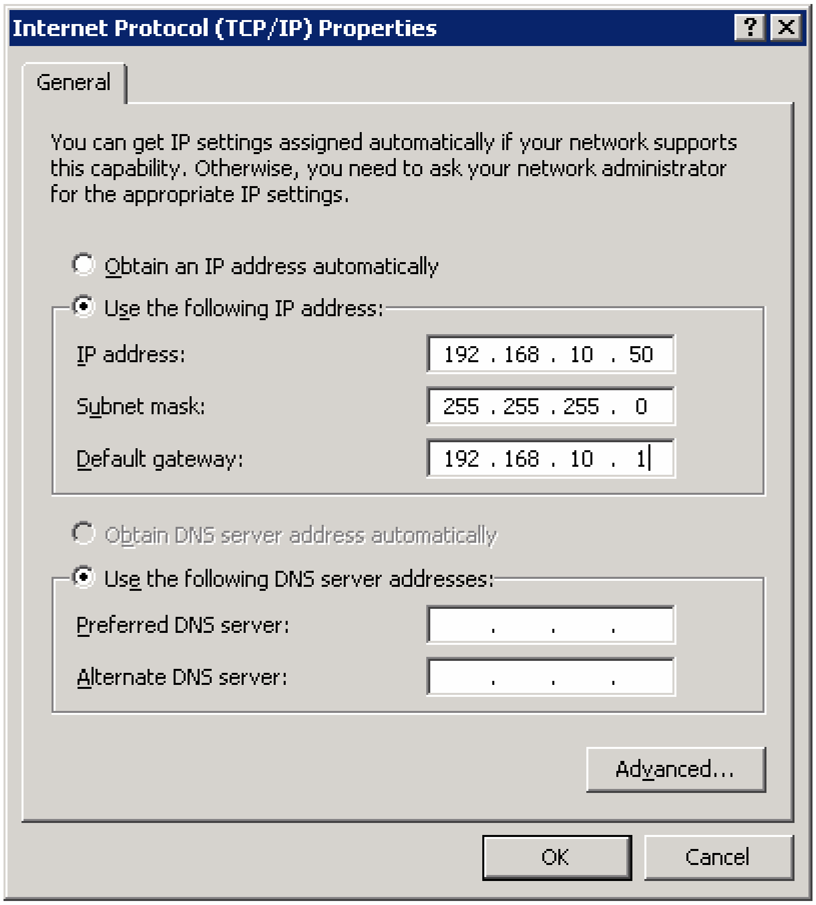
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☑ Show icon in notification area when connected

☑ Notify me when this connection has limited or no connectivity

Figure 3-2: Network Connection Properties

Finally, configure the IP address shown in the diagram on the interface.



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 10 . 50

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 10 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Figure 3-3: IP Properties

Click **OK** once to apply the TCP/IP settings and again to exit the configuration dialog box for the LAN interface. Open the Start Menu, and then click **Run**.... Issue the **cmd** command and press the [Return] key. At the Windows command-line prompt, ping the R1 Ethernet interface. You should receive responses. If you do not receive a response, troubleshoot by verifying the VLAN of the switchports and the IP address and subnet mask on each of the devices attached to the switch.

```
C:\Documents and Settings\Administrator> ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:

Reply from 192.168.10.1: bytes=32 time=1ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Step 4: Extract SDM on the Host

Now that the router is ready to be accessed from SDM and there is connectivity between the router and the PC, you can use SDM to configure the router.

You should start by extracting the SDM zip file to a directory on your hard drive. In this example, the directory used is “C:\sdm\,” although you can use any path you want. If your version of Windows has a built-in zip utility, you can use that to extract it, or if you don’t have it built in, you can use a third-party tool such as WinZip. To get to the built in Windows Extraction Wizard, right-click the SDM zip file and click **Extract All...**. If you decide to use a third-party tool, extract the file to the directory of your choice and skip to the next step.

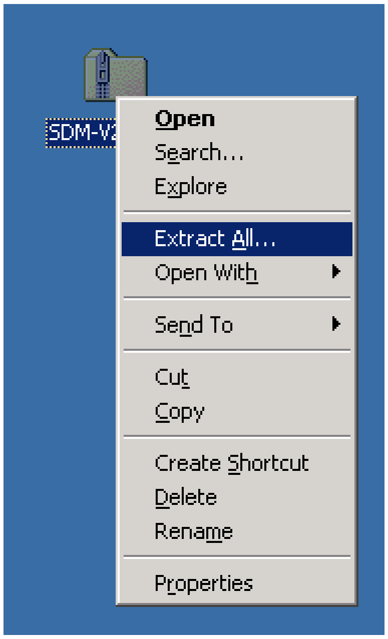
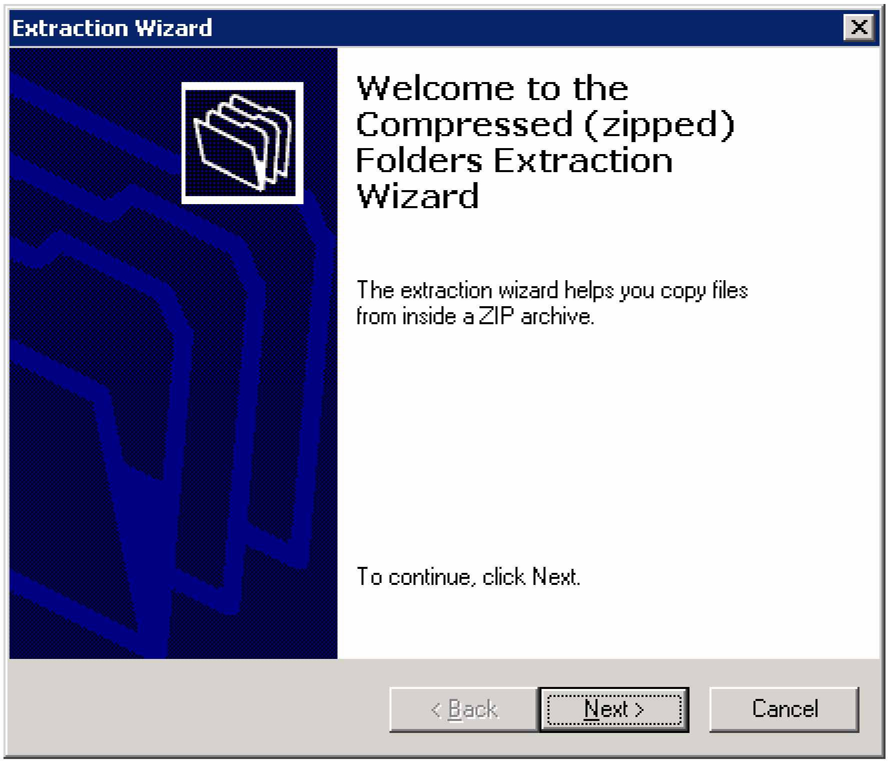


Figure 4-1: Zip File Menu

Once the extraction wizard has opened, click **Next** to get to the destination selection screen.



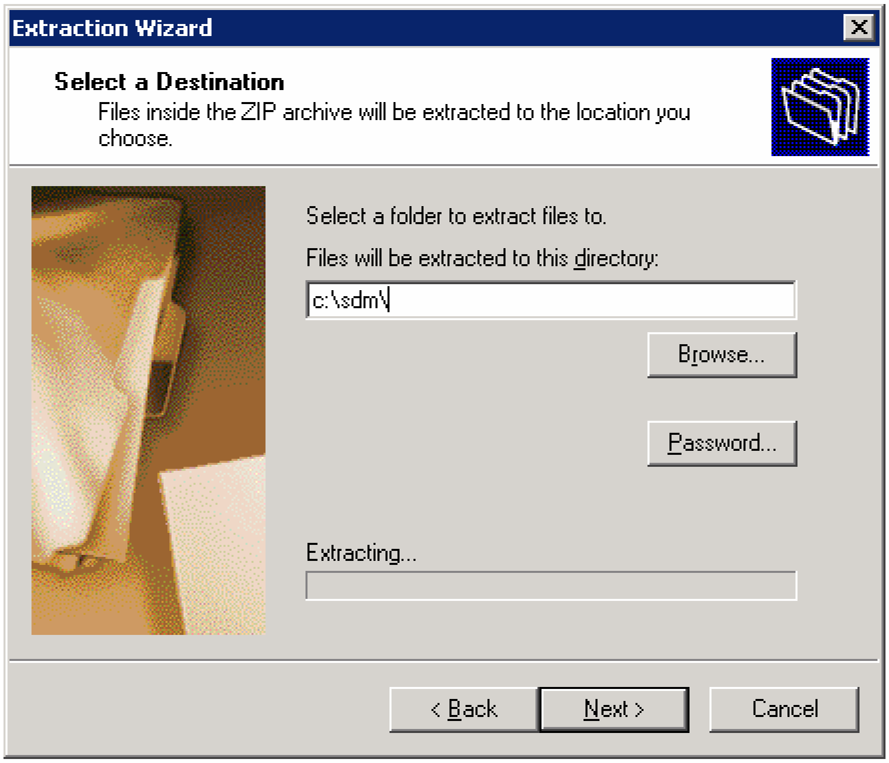
Welcome to the Compressed (zipped) Folders Extraction Wizard

The extraction wizard helps you copy files from inside a ZIP archive.

To continue, click Next.

Figure 4-2: Windows Extraction Wizard

Select the folder you want to use as the destination directory, and then click **Next**.



Extraction Wizard

Select a Destination

Files inside the ZIP archive will be extracted to the location you choose.

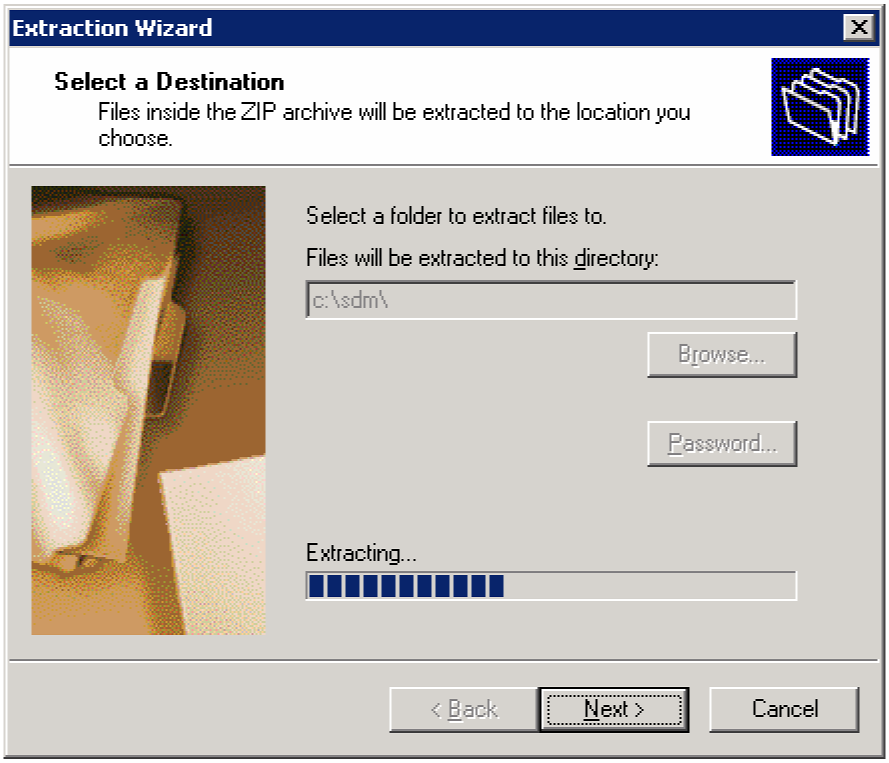
Select a folder to extract files to.

Files will be extracted to this directory:

c:\sdm\

Figure 4-3: Destination Selection Dialog

The files are extracted. It may take a few seconds for the extraction to finish.



Extraction Wizard

Select a Destination

Files inside the ZIP archive will be extracted to the location you choose.

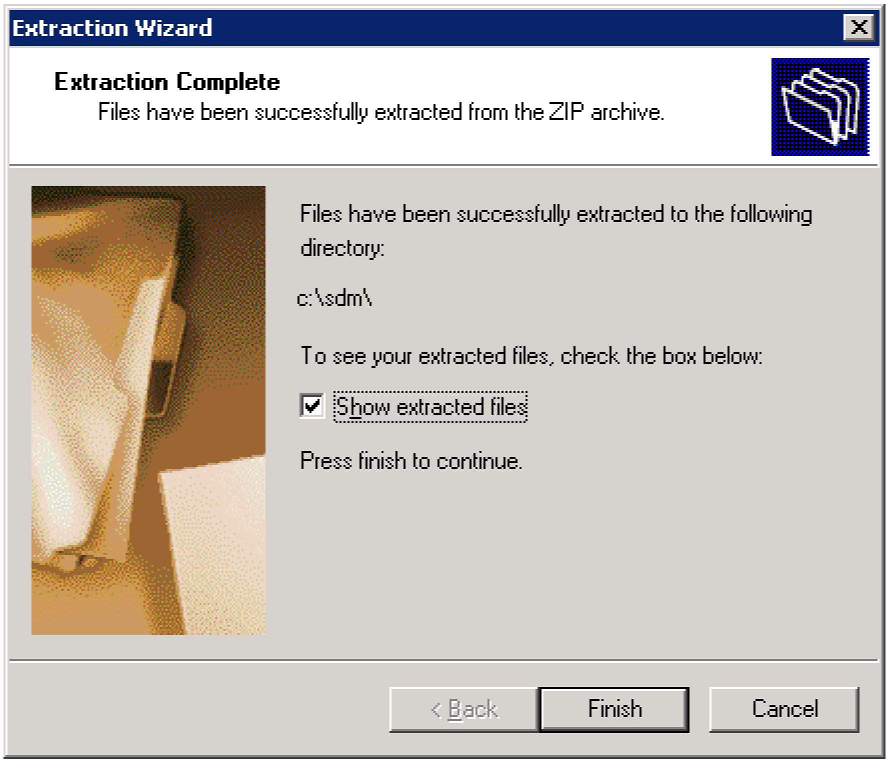
Select a folder to extract files to.

Files will be extracted to this directory:

c:\sdm\

Figure 4-4: Windows Extraction Wizard

Afterwards, you are prompted to decide if you want to show the extracted files. Check this option if it is not already checked, and then click **Finish**.



Files have been successfully extracted from the ZIP archive.

Files have been successfully extracted to the following directory:

c:\sdm\

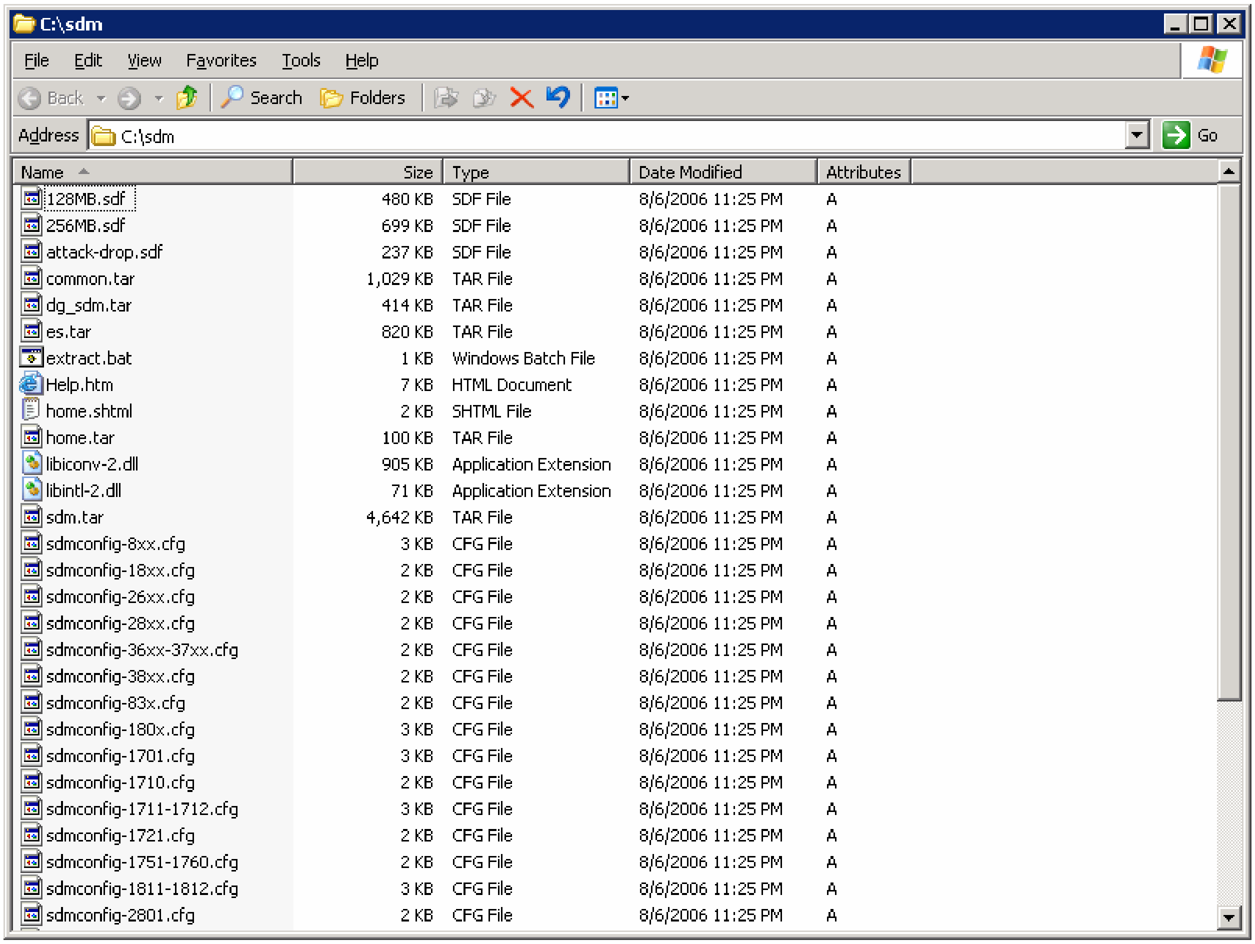
To see your extracted files, check the box below:

☑ Show extracted files

Press finish to continue.

Figure 4-5: Final Extraction Wizard Dialog

After you have extracted the file, open the directory to which the file was extracted. The files in this directory may look different depending on the version of SDM you have.



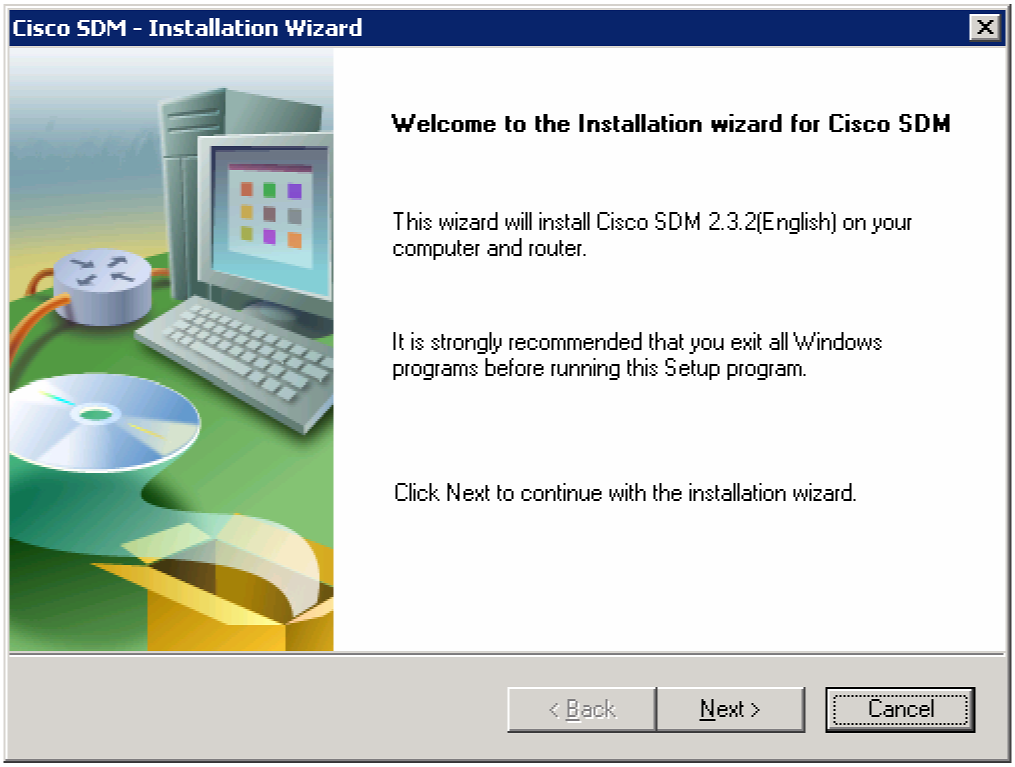
Name	Size	Type	Date Modified	Attributes
128MB.sdf	480 KB	SDF File	8/6/2006 11:25 PM	A
256MB.sdf	699 KB	SDF File	8/6/2006 11:25 PM	A
attack-drop.sdf	237 KB	SDF File	8/6/2006 11:25 PM	A
common.tar	1,029 KB	TAR File	8/6/2006 11:25 PM	A
dg_sdm.tar	414 KB	TAR File	8/6/2006 11:25 PM	A
es.tar	820 KB	TAR File	8/6/2006 11:25 PM	A
extract.bat	1 KB	Windows Batch File	8/6/2006 11:25 PM	A
Help.htm	7 KB	HTML Document	8/6/2006 11:25 PM	A
home.shtml	2 KB	SHTML File	8/6/2006 11:25 PM	A
home.tar	100 KB	TAR File	8/6/2006 11:25 PM	A
libiconv-2.dll	905 KB	Application Extension	8/6/2006 11:25 PM	A
libintl-2.dll	71 KB	Application Extension	8/6/2006 11:25 PM	A
sdm.tar	4,642 KB	TAR File	8/6/2006 11:25 PM	A
sdmconfig-8xx.cfg	3 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-18xx.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-26xx.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-28xx.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-36xx-37xx.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-38xx.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-83x.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-180x.cfg	3 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-1701.cfg	3 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-1710.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-1711-1712.cfg	3 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-1721.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-1751-1760.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-1811-1812.cfg	3 KB	CFG File	8/6/2006 11:25 PM	A
sdmconfig-2801.cfg	2 KB	CFG File	8/6/2006 11:25 PM	A

Figure 4-6: Directory of SDM Extraction

You are almost ready to use SDM to configure the router. The last step is installing the SDM application on the PC.

Step 5: Install SDM on the PC

Double-click the **setup.exe** executable program to open the installation wizard. Once the installation wizard screen opens, click **Next**.



Cisco SDM - Installation Wizard

Welcome to the Installation wizard for Cisco SDM

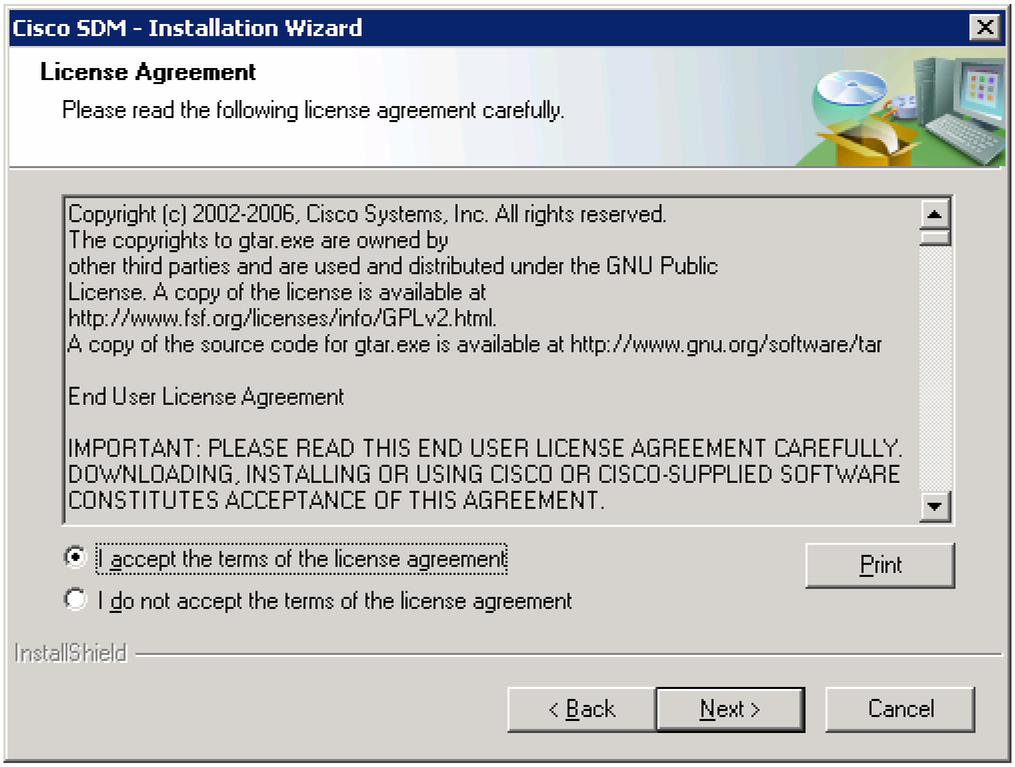
This wizard will install Cisco SDM 2.3.2(English) on your computer and router.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Next to continue with the installation wizard.

Figure 5-1: Welcome Screen for SDM Installation Wizard

Accept the terms of the license agreement, and then click **Next**.



Cisco SDM - Installation Wizard

License Agreement

Please read the following license agreement carefully.

Copyright (c) 2002-2006, Cisco Systems, Inc. All rights reserved.
The copyrights to gtar.exe are owned by
other third parties and are used and distributed under the GNU Public
License. A copy of the license is available at
http://www.fsf.org/licenses/info/GPLv2.html.
A copy of the source code for gtar.exe is available at http://www.gnu.org/software/tar

End User License Agreement

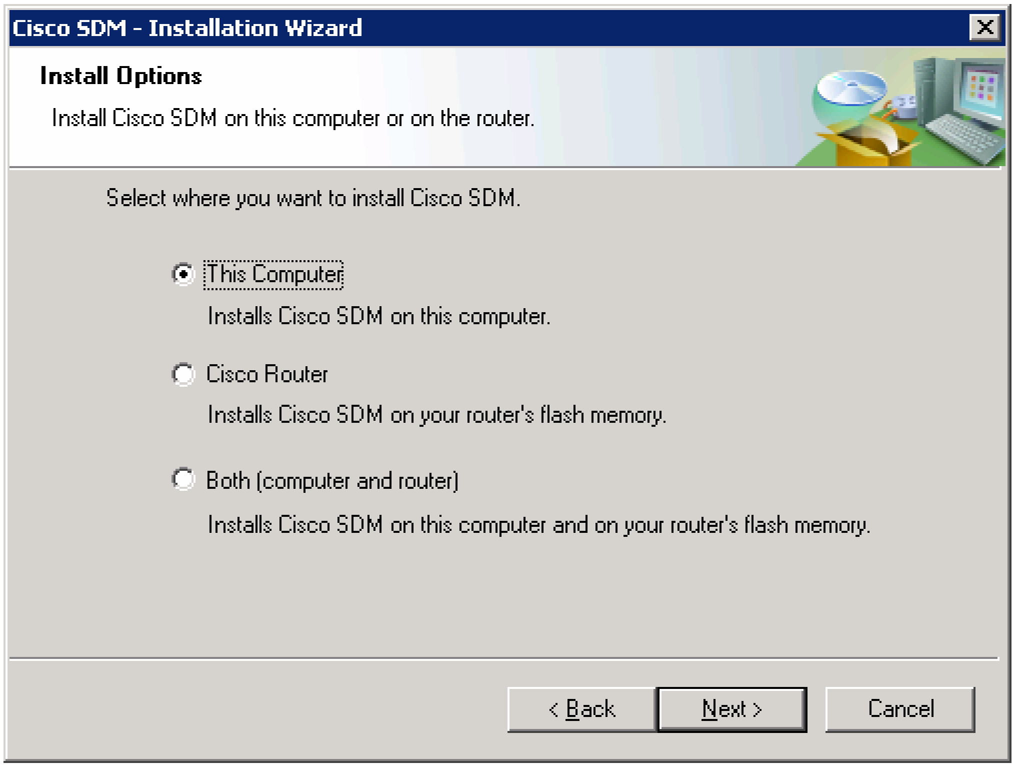
IMPORTANT: PLEASE READ THIS END USER LICENSE AGREEMENT CAREFULLY.
DOWNLOADING, INSTALLING OR USING CISCO OR CISCO-SUPPLIED SOFTWARE
CONSTITUTES ACCEPTANCE OF THIS AGREEMENT.

● I accept the terms of the license agreement

○ I do not accept the terms of the license agreement

Figure 5-2: SDM License Agreement

The next screen prompts you to choose from three options where you want to install SDM.



Cisco SDM - Installation Wizard

Install Options

Install Cisco SDM on this computer or on the router.

Select where you want to install Cisco SDM.

● This Computer

Installs Cisco SDM on this computer.

○ Cisco Router

Installs Cisco SDM on your router's flash memory.

○ Both (computer and router)

Installs Cisco SDM on this computer and on your router's flash memory.

Figure 5-3: Installation Location Options

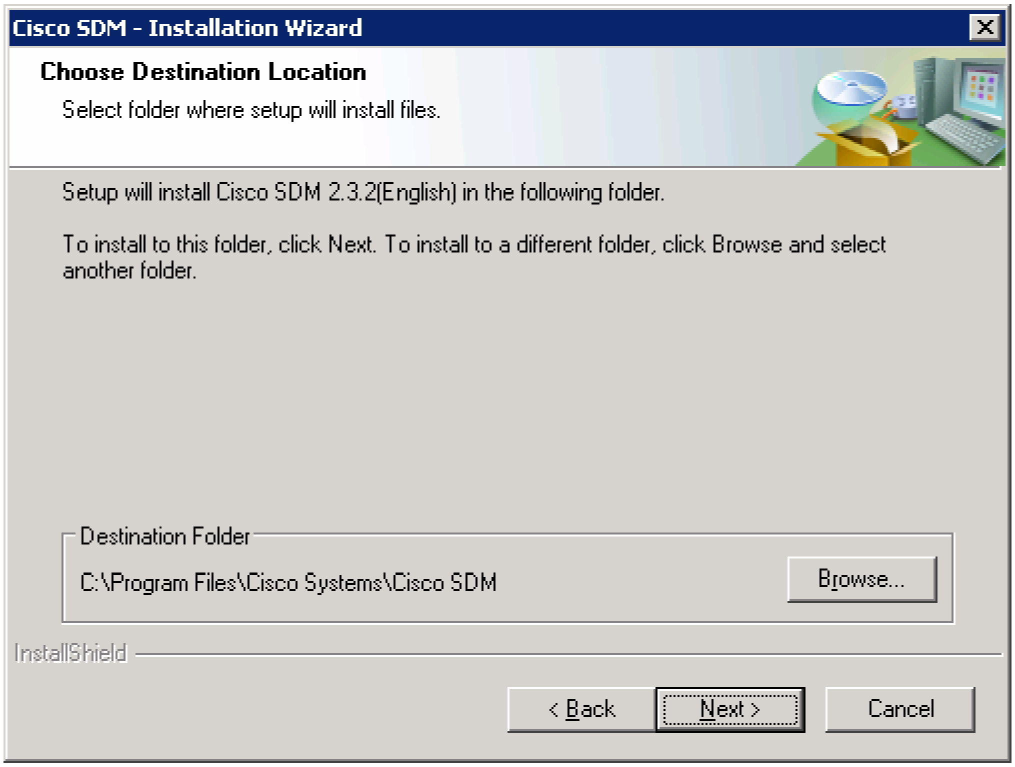
When installing SDM, you can install the application on the computer and not place it on the flash memory of the router, or you can install it on the router without affecting the computer, or you can install it to both. Both installation types are very similar. This lab explains how to install SDM on your computer and on the Cisco router. It is not necessary to explain how to install it on both because that is self-evident once you have learned how to install to one or the other. If you do not want to install SDM to your computer, skip to step 7.

What are the advantages and disadvantages of installing SDM on the computer only?

What are the advantages and disadvantages of installing SDM on the router only?

What are the advantages and disadvantages of installing SDM on both the router and PC?

For now, click **This computer**, and then click **Next**. Use the default destination folder and click **Next** again.



Cisco SDM - Installation Wizard

Choose Destination Location

Select folder where setup will install files.

Setup will install Cisco SDM 2.3.2(English) in the following folder.

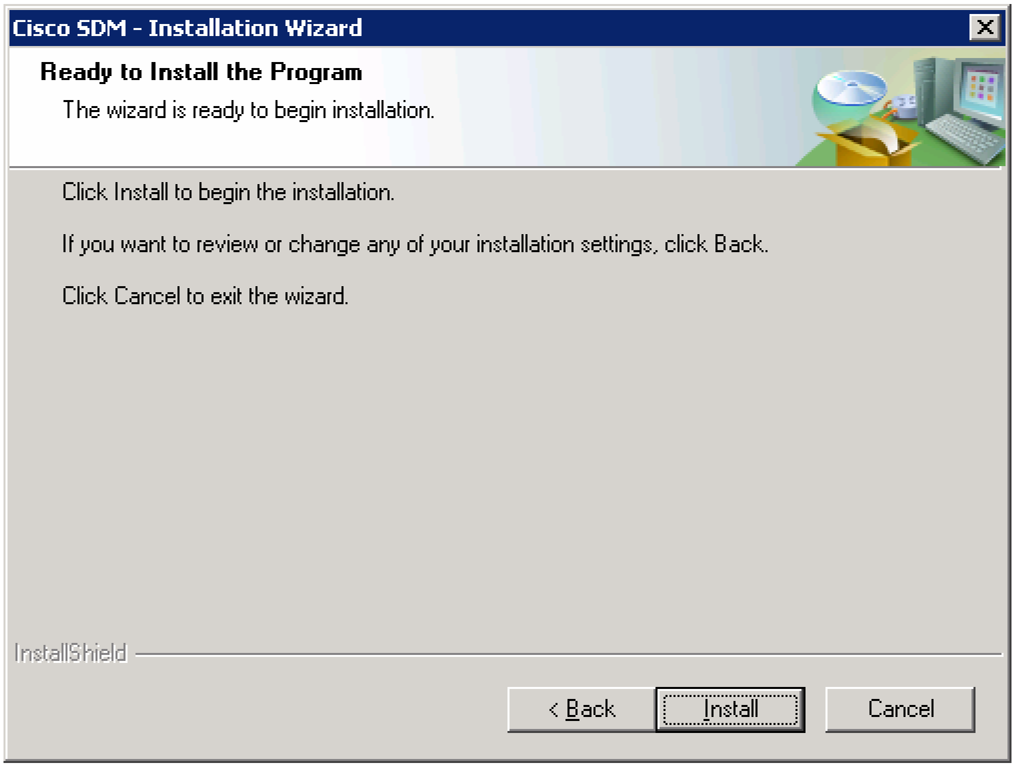
To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\Program Files\Cisco Systems\Cisco SDM

Figure 5-4: Local Installation Location Dialog

Click **Install** to begin the installation.



Ready to Install the Program

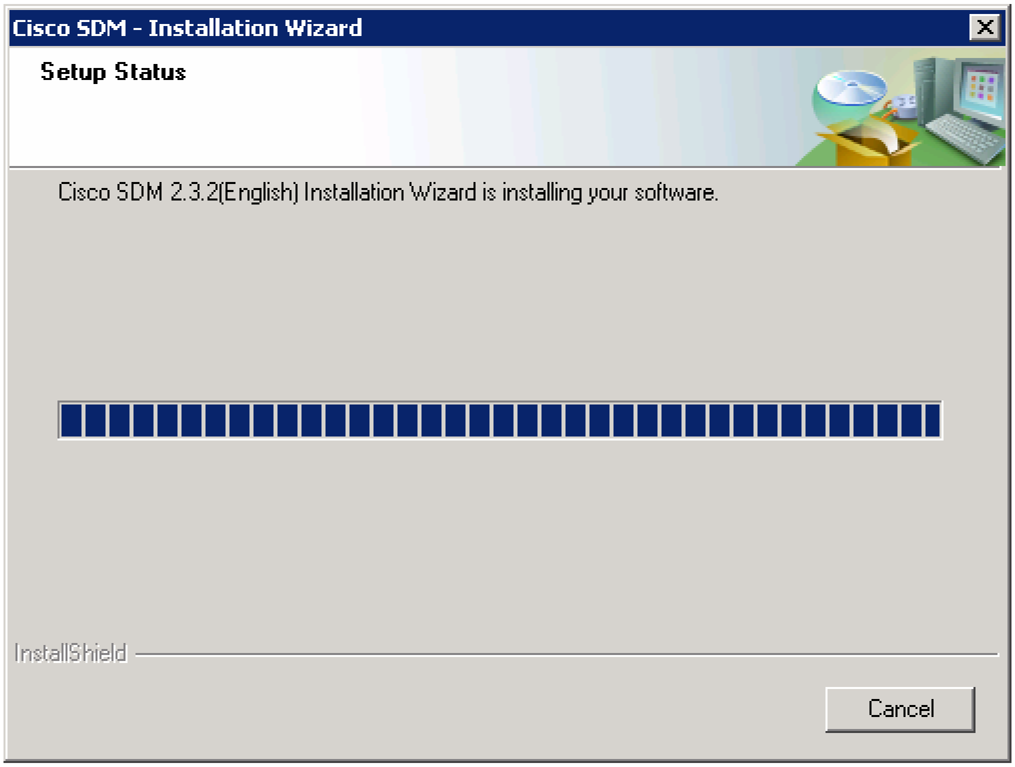
The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back.

Click Cancel to exit the wizard.

Figure 5-5: Installation Prompt

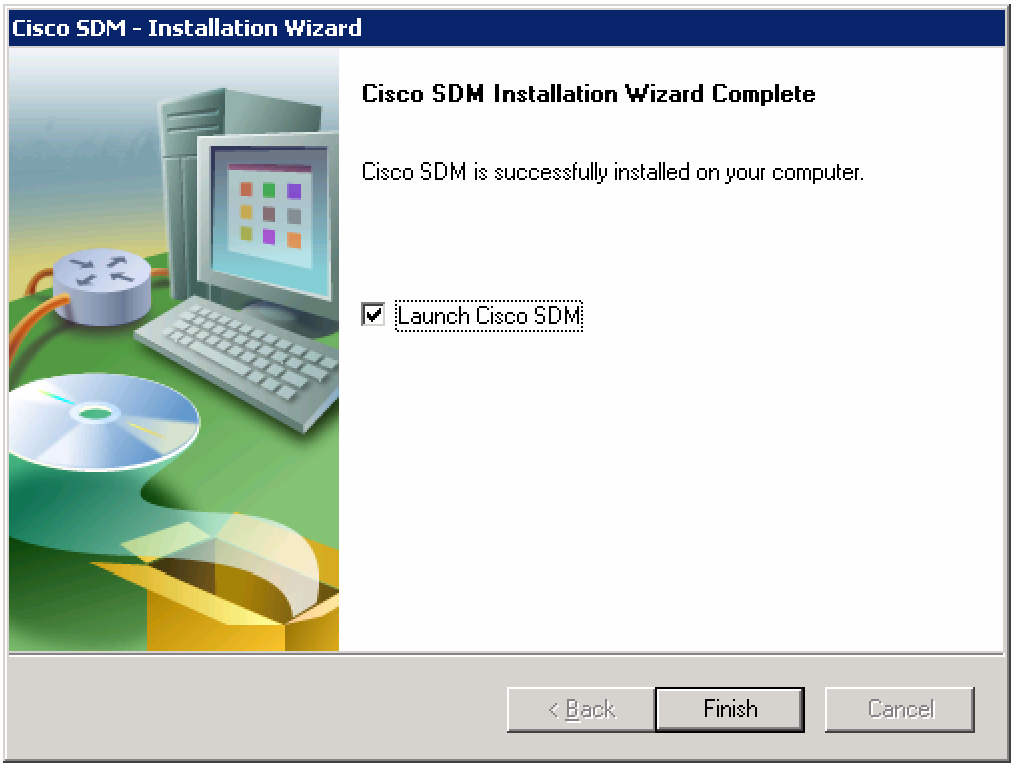


Setup Status

Cisco SDM 2.3.2(English) Installation Wizard is installing your software.

Figure 5-6: Installation Progress Information

The software installs, and then you are prompted with a final dialog box to launch SDM. Check the **Launch Cisco SDM** box, and then click **Finish**.



Cisco SDM - Installation Wizard

Cisco SDM Installation Wizard Complete

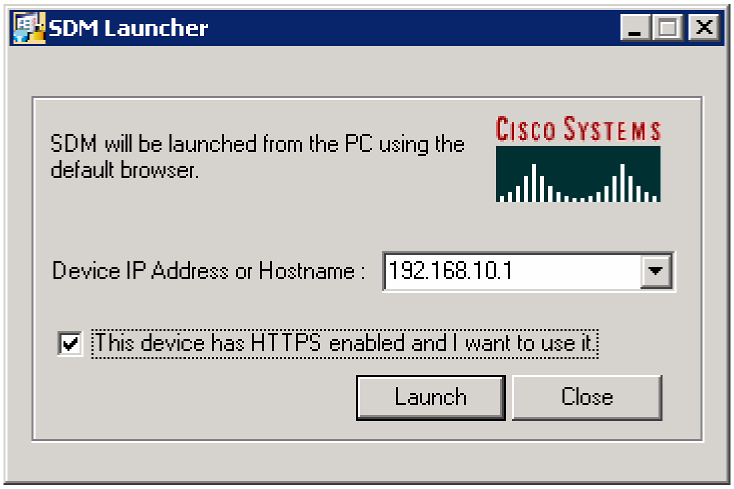
Cisco SDM is successfully installed on your computer.

☑ Launch Cisco SDM

Figure 5-7: Final Installation Wizard Report

Step 6: Run SDM from the PC

SDM should start up from the installer when you have completed step 5 if you checked the Launch Cisco SDM option. If you did not, or you are running SDM without just installing it, click the icon on the desktop labeled **Cisco SDM**. The SDM Launcher dialog box will open. Type the IP address of the router shown in the diagram as a Device IP Address. Check **This device has HTTPS enabled and I want to use it** if you enabled the HTTP secure server in step 2. Then click the **Launch** button.



SDM will be launched from the PC using the default browser.

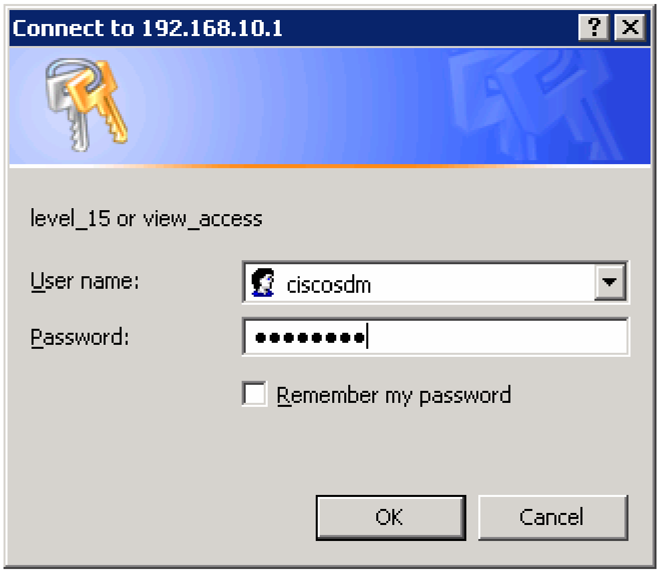
Device IP Address or Hostname : 192.168.10.1

☑ This device has HTTPS enabled and I want to use it.

Figure 6-1: SDM Launcher Window

Click **Yes** when the security warning appears. Note that Internet Explorer may block SDM at first, and you will need to allow it or adjust your Internet Explorer security settings accordingly to use it. Depending on the version of Internet Explorer you are running, one of these settings is especially important for running SDM locally, and it is on the Tools menu, under Internet Options.... Click the **Advanced** tab, and under the Security heading, check **Allow active content to be run in files on My Computer** if it is not already checked.

Enter in the username and password you created in step 2.



level_15 or view_access

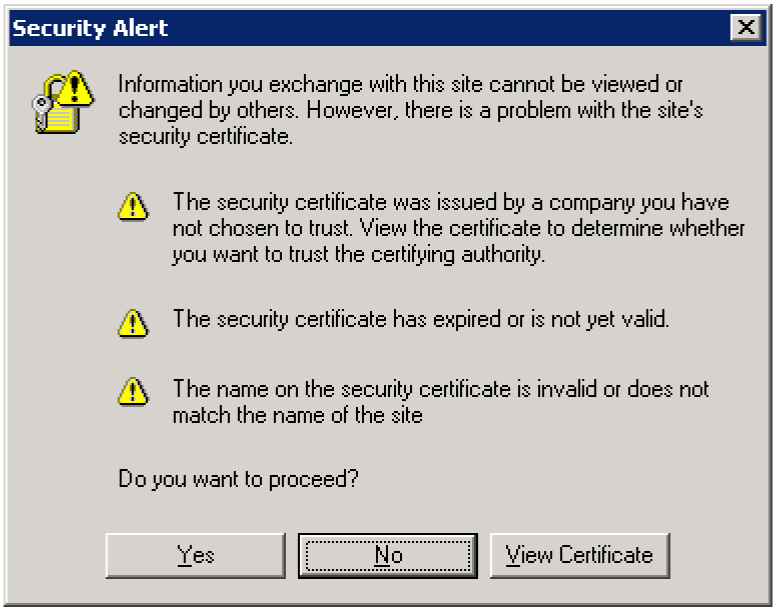
User name: ciscosdm

Password: ●●●●●●●●●

☐ Remember my password

Figure 6-2: HTTP Authentication Screen

You may be prompted to accept a certificate from this router. Accept the certificate to proceed. After this, give the username and password for the router and click **Yes**.



Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

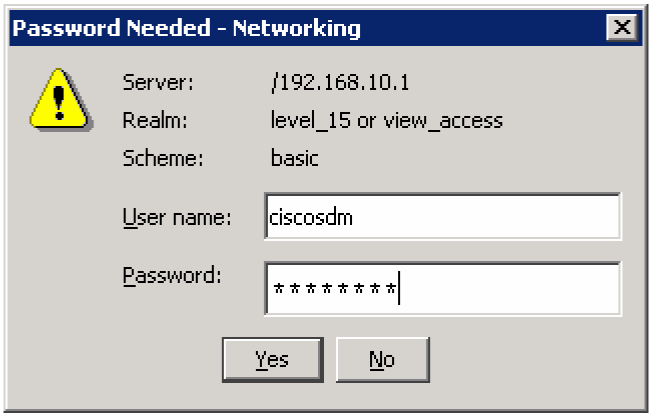
The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

The security certificate has expired or is not yet valid.

The name on the security certificate is invalid or does not match the name of the site

Do you want to proceed?

Figure 6-3: Internet Explorer Security Alert Prompt



Password Needed - Networking

Server: /192.168.10.1

Realm: level_15 or view_access

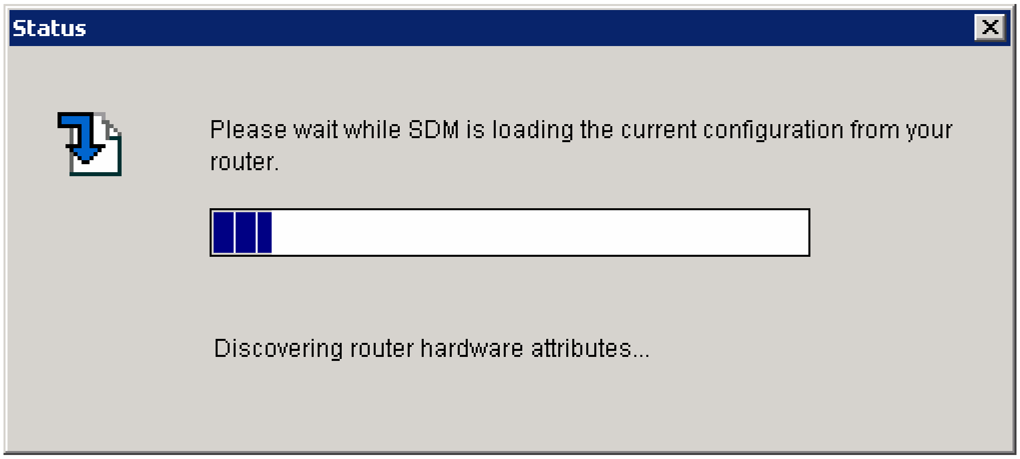
Scheme: basic

User name: ciscosdm

Password: ********

Figure 6-4: SDM Authentication Dialog

SDM reads the configuration off the router.

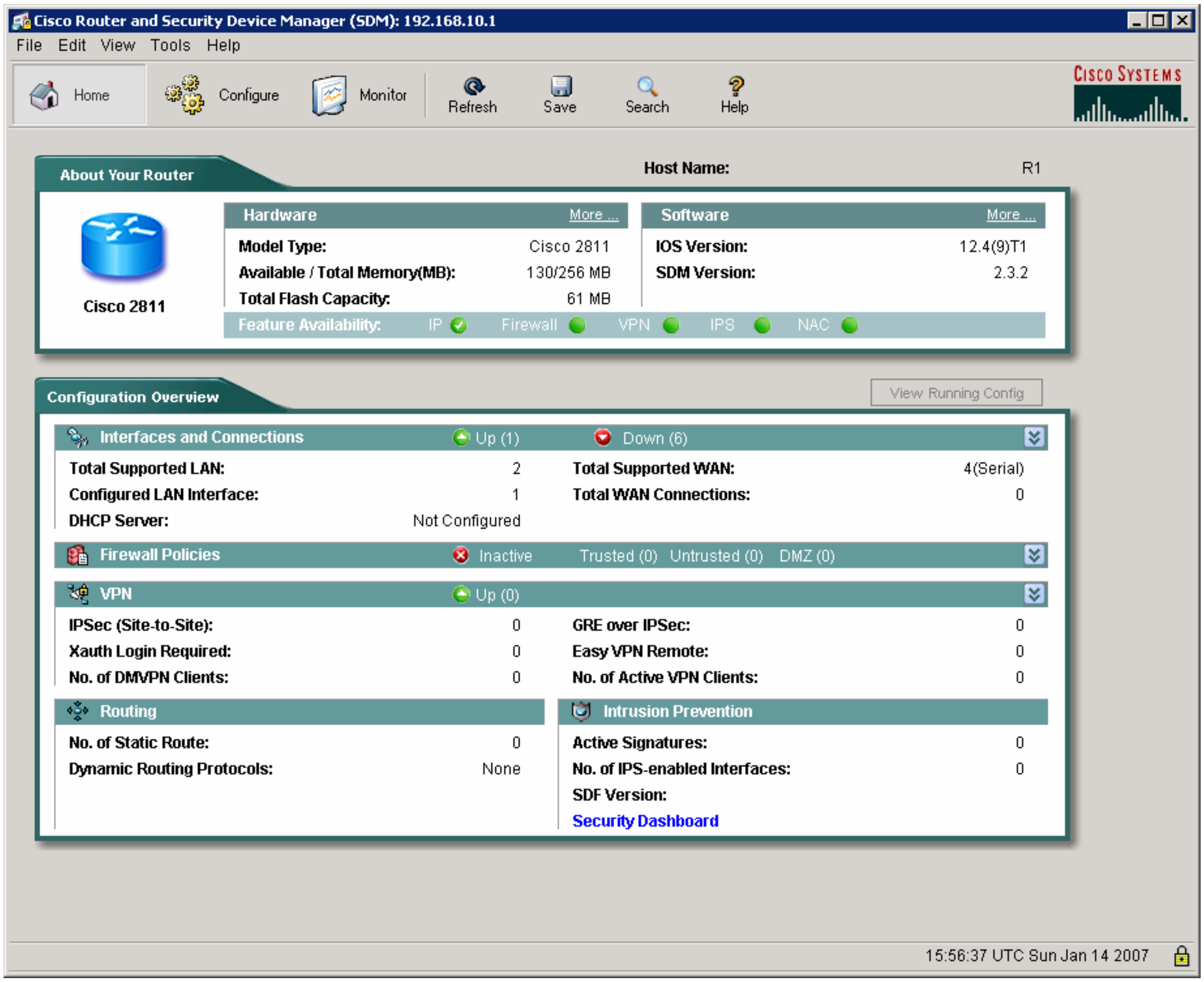


Please wait while SDM is loading the current configuration from your router.

Discovering router hardware attributes...

Figure 6-5: SDM Load Progress Indicator

If everything was configured correctly in step 2, you will be able to access the SDM dashboard. If your configuration here looks correct, it means you have successfully configured and connected to SDM. Your information may vary depending upon which version of SDM you are running.



About Your Router

Host Name: R1

Cisco 2811

Hardware	More ...	Software	More ...
Model Type:	Cisco 2811	IOS Version:	12.4(9)T1
Available / Total Memory(MB):	130/256 MB	SDM Version:	2.3.2
Total Flash Capacity:	61 MB		

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Configuration Overview

Interfaces and Connections — Up (1) — Down (6)

Total Supported LAN:	2	Total Supported WAN:	4(Serial)
Configured LAN Interface:	1	Total WAN Connections:	0
DHCP Server:	Not Configured		

Firewall Policies — Inactive — Trusted (0) Untrusted (0) DMZ (0)

VPN — Up (0)

IPSec (Site-to-Site):	0	GRE over IPSec:	0
Xauth Login Required:	0	Easy VPN Remote:	0
No. of DMVPN Clients:	0	No. of Active VPN Clients:	0

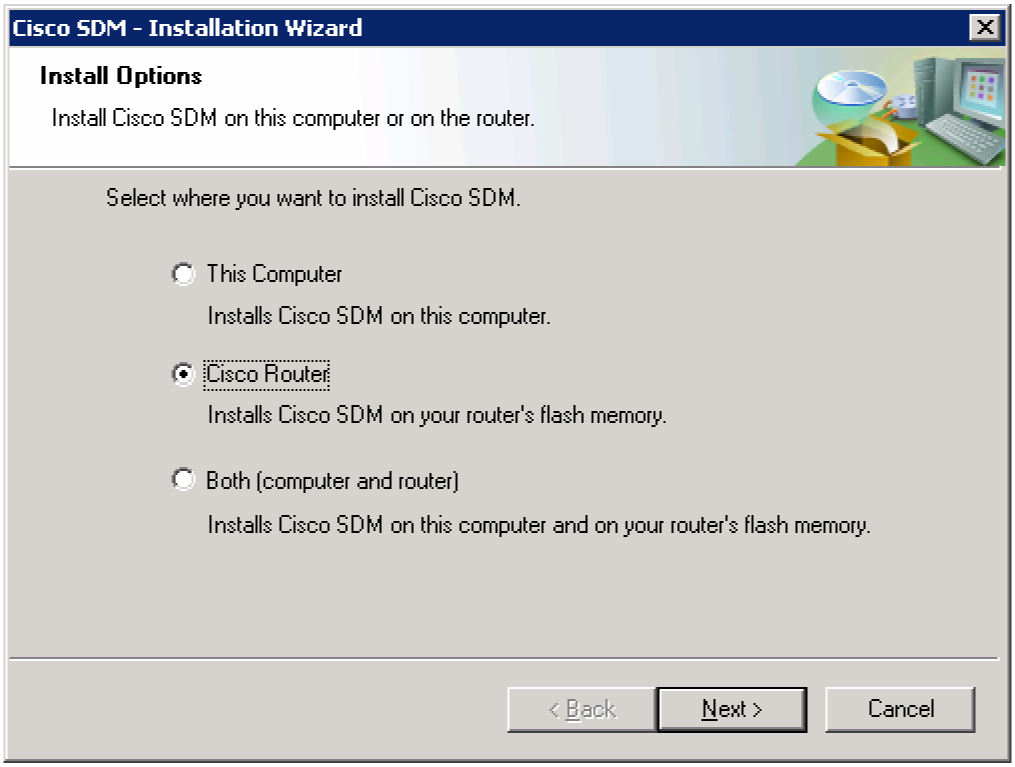
Routing		Intrusion Prevention	
No. of Static Route:	0	Active Signatures:	0
Dynamic Routing Protocols:	None	No. of IPS-enabled Interfaces:	0
		SDF Version:	

15:56:37 UTC Sun Jan 14 2007

Figure 6-6: SDM Dashboard

Step 7: Install SDM to the Router

Follow step 6 until the prompt shown in the following figure appears.. When this window appears, click **Cisco Router** to install SDM to your router's flash memory. If you don't want to install SDM to your router's flash memory, or do not have the available space on the flash drive, then do not attempt to install SDM to the router.



Cisco SDM - Installation Wizard

Install Options

Install Cisco SDM on this computer or on the router.

Select where you want to install Cisco SDM.

○ This Computer

Installs Cisco SDM on this computer.

● Cisco Router

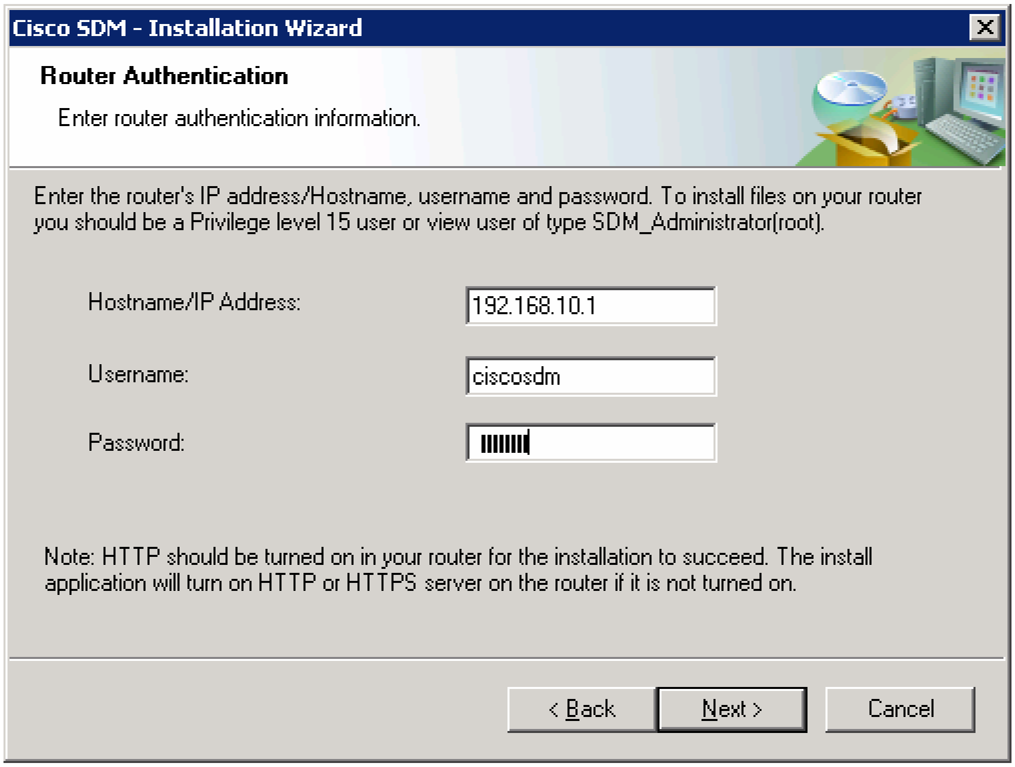
Installs Cisco SDM on your router's flash memory.

○ Both (computer and router)

Installs Cisco SDM on this computer and on your router's flash memory.

Figure 7-1: Installation Location Options

Enter your router's information so that the installer can remotely access and install SDM to the router.



Cisco SDM - Installation Wizard

Router Authentication

Enter router authentication information.

Enter the router's IP address/Hostname, username and password. To install files on your router you should be a Privilege level 15 user or view user of type SDM_Administrator(root).

Hostname/IP Address: 192.168.10.1

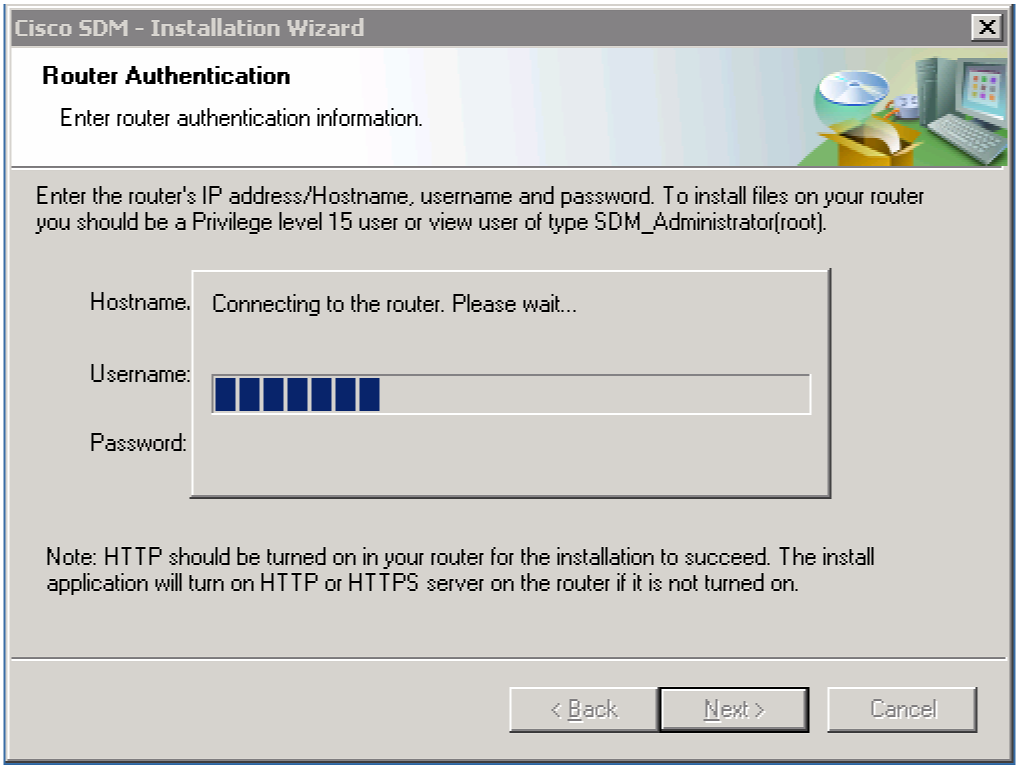
Username: ciscosdm

Password: ||||||||

Note: HTTP should be turned on in your router for the installation to succeed. The install application will turn on HTTP or HTTPS server on the router if it is not turned on.

Figure 7-2: Router Authentication Dialog

Cisco SDM connects to the router. You may notice some messages being logged to the console. This is normal.



Cisco SDM - Installation Wizard

Router Authentication

Enter router authentication information.

Enter the router's IP address/Hostname, username and password. To install files on your router you should be a Privilege level 15 user or view user of type SDM_Administrator(root).

Hostname:

Username:

Password:

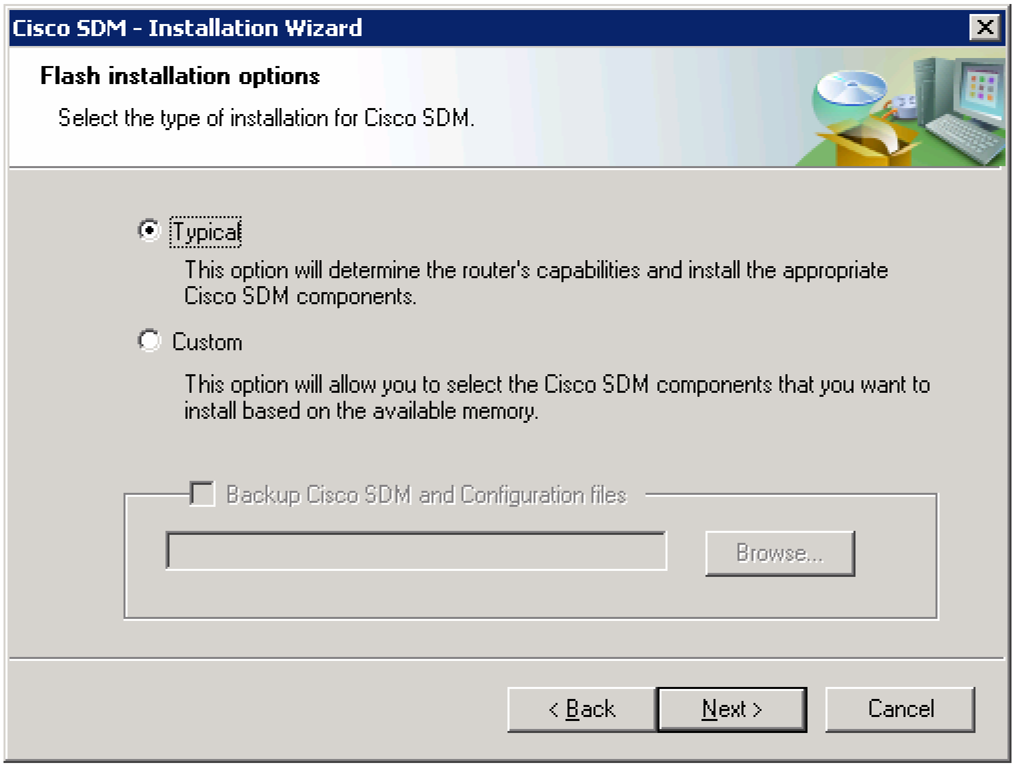
Connecting to the router. Please wait...

Note: HTTP should be turned on in your router for the installation to succeed. The install application will turn on HTTP or HTTPS server on the router if it is not turned on.

Figure 7-3: Router Connection Indicator

```
Jan 14 16:15:26.367: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:15:30.943: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:15:36.227: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:15:39.211: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:15:44.583: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
```

As shown in the following figure, choose **Typical** as your installation type, and then click **Next**.



Cisco SDM - Installation Wizard

Flash installation options

Select the type of installation for Cisco SDM.

● Typical

This option will determine the router's capabilities and install the appropriate Cisco SDM components.

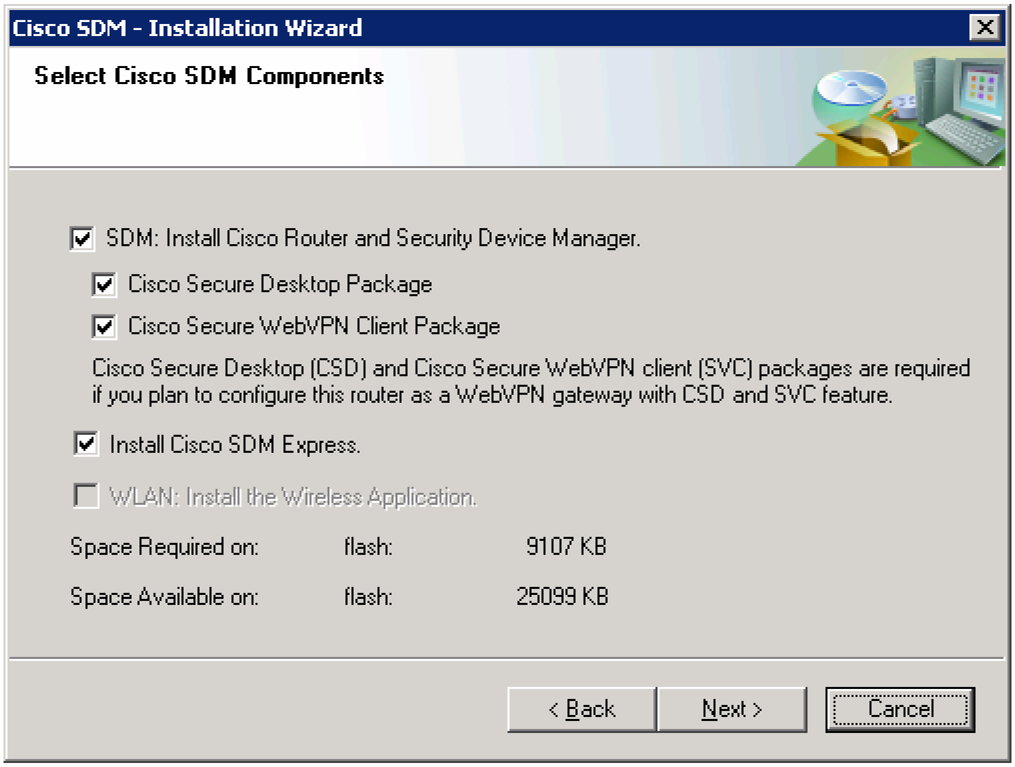
○ Custom

This option will allow you to select the Cisco SDM components that you want to install based on the available memory.

□ Backup Cisco SDM and Configuration files

Figure 7-4: SDM Installation Options, Step 1

Leave the default installation options checked and click **Next**.



Select Cisco SDM Components

☑ SDM: Install Cisco Router and Security Device Manager.

☑ Cisco Secure Desktop Package

☑ Cisco Secure WebVPN Client Package

Cisco Secure Desktop (CSD) and Cisco Secure WebVPN client (SVC) packages are required if you plan to configure this router as a WebVPN gateway with CSD and SVC feature.

☑ Install Cisco SDM Express.

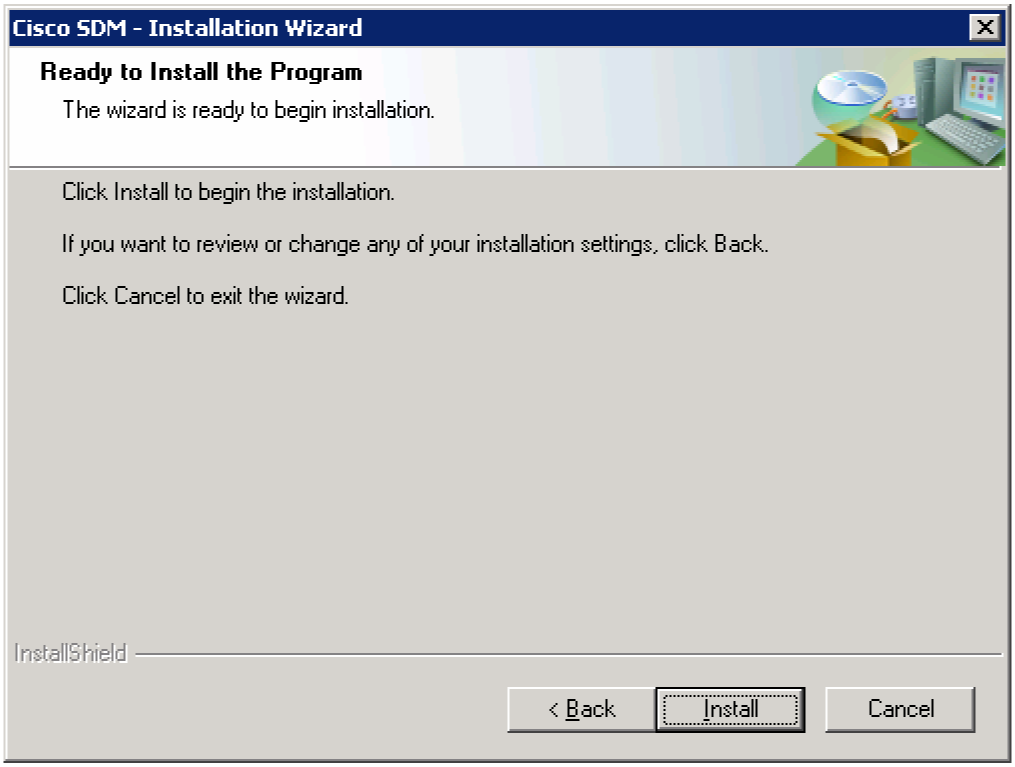
□ WLAN: Install the Wireless Application.

Space Required on: flash: 9107 KB

Space Available on: flash: 25099 KB

Figure 7-5: SDM Installation Options, Step 2

Finally, click **Install** for the installation process to begin. During the installation, more messages may be logged to the console. This installation process takes a little while (look at the timestamps in the console output below to estimate the duration on a Cisco 2811). The time will vary by router model.


Cisco SDM - Installation Wizard

Ready to Install the Program

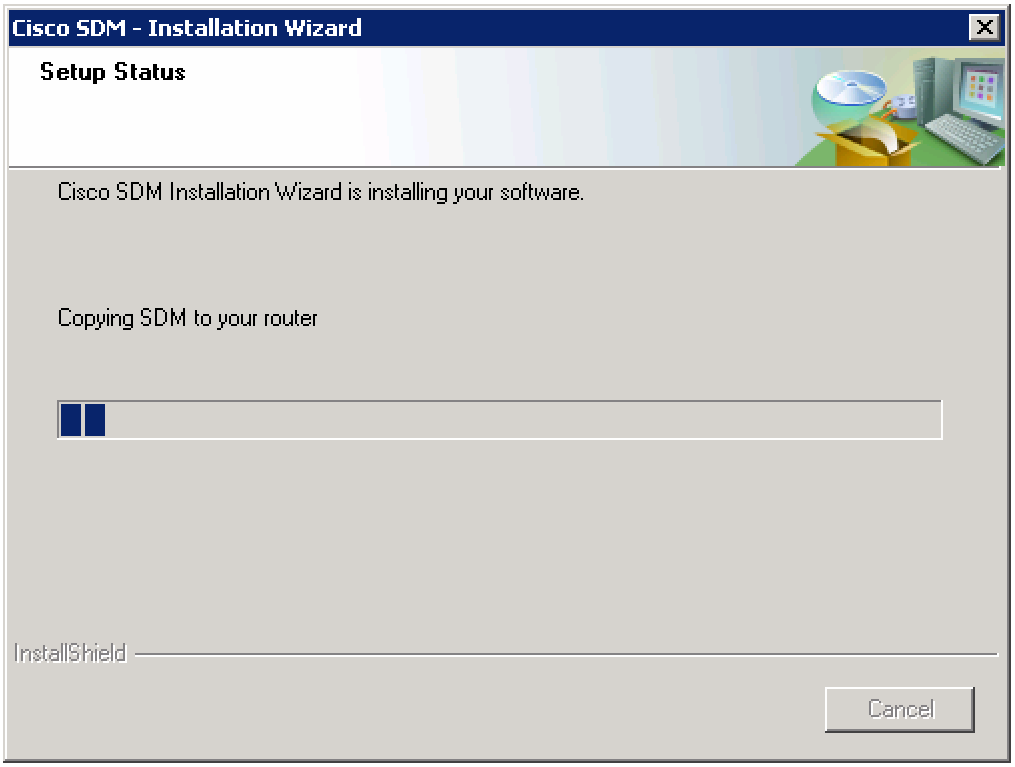
The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back.

Click Cancel to exit the wizard.

Figure 7-6: Confirmation Prompt


Cisco SDM - Installation Wizard

Setup Status

Cisco SDM Installation Wizard is installing your software.

Copying SDM to your router

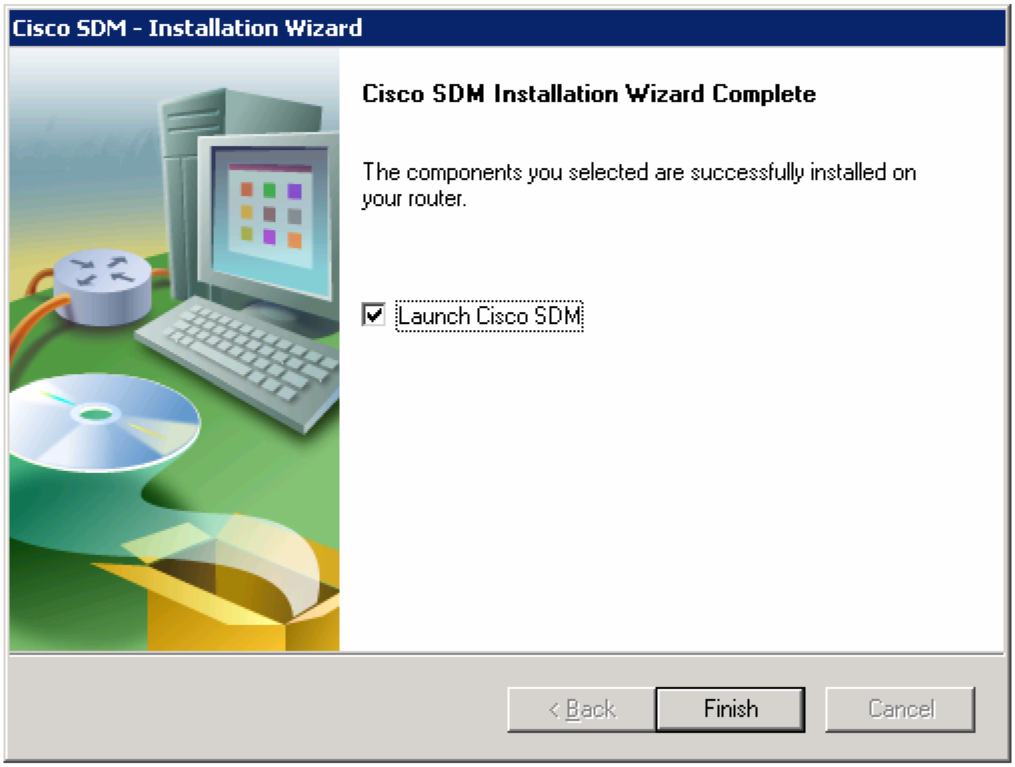
Figure 7-7: Installation Progress Indicator

```
Jan 14 16:19:40.795: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:19:43.855: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:19:49.483: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:25:57.823: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:26:02.331: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:27:42.279: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:27:46.767: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:28:11.403: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:28:15.795: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
Jan 14 16:29:04.391: %SYS-5-CONFIG_I: Configured from console by ciscosdm on
vty0 (192.168.10.50)
```

At the end of the installation, you are prompted to launch SDM on the router. Before you do this, go onto the console and issue the **show flash:** command. Notice all the files that SDM installed to flash. Before the installation, the only file listed was the first file, the IOS image.

```
R1# show flash:

CompactFlash directory:
File  Length   Name/status
  1   38523272 c2800nm-advipservicesk9-mz.124-9.T1.bin
  2   1038     home.shtml
  3   1823     sdmconfig-2811.cfg
  4   102400   home.tar
  5   491213   128MB.sdf
  6   1053184  common.tar
  7   4753408  sdm.tar
  8   1684577  securedesktop-ios-3.1.1.27-k9.pkg
  9   398305   sslclient-win-1.1.0.154.pkg
 10   839680   es.tar
[47849552 bytes used, 16375724 available, 64225276 total]
62720K bytes of ATA CompactFlash (Read/Write)
```

As shown in the following figure, make sure that the **Launch Cisco SDM** option is checked, and then click the **Finish** button to launch SDM.



Cisco SDM - Installation Wizard

Cisco SDM Installation Wizard Complete

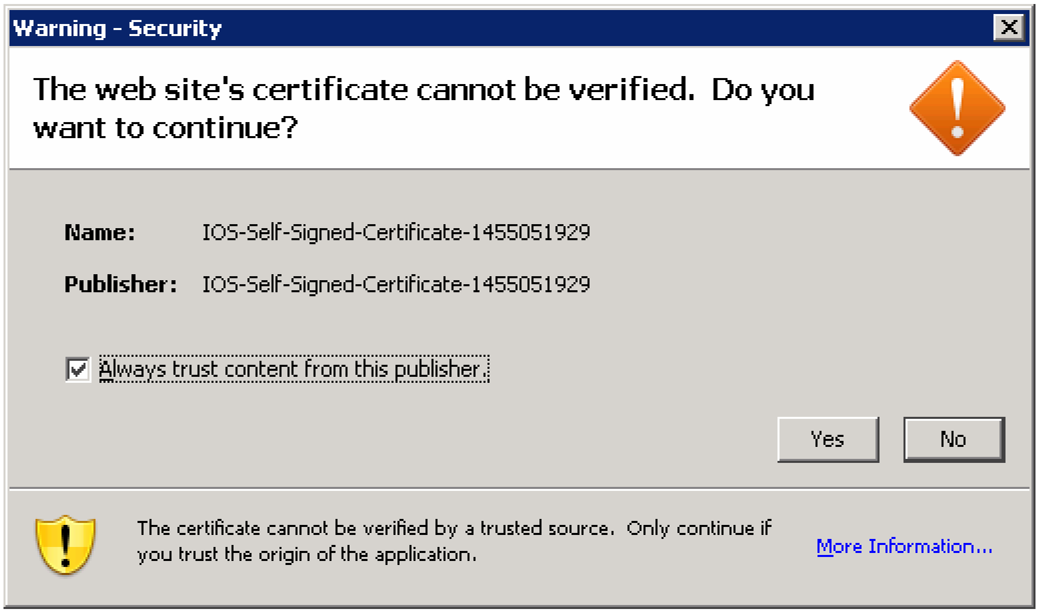
The components you selected are successfully installed on your router.

☑ Launch Cisco SDM

Figure 7-8: Final SDM Installation Dialog

Step 8: Run SDM from the Router

SDM should start up from the installer when you have completed the previous step if you checked the **Launch Cisco SDM** option. If you did not, or you are running SDM without installing it, open up Internet Explorer and navigate to the URL "https://<IP address>/" or "http://<IP address>/" depending on whether you enabled the HTTP secure server in step 2. When you are prompted to accept the certificate, click **Yes**.



Warning - Security

The web site's certificate cannot be verified. Do you want to continue?

Name: IOS-Self-Signed-Certificate-1455051929

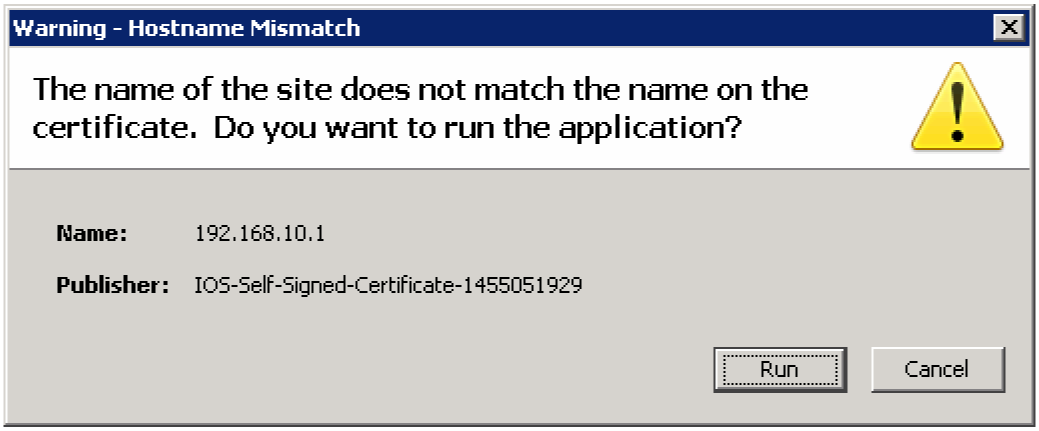
Publisher: IOS-Self-Signed-Certificate-1455051929

☑ Always trust content from this publisher.

The certificate cannot be verified by a trusted source. Only continue if you trust the origin of the application.

Figure 8-1: Internet Explorer Certificate Confirmation

Ignore the security warnings and click **Run**.



Warning - Hostname Mismatch

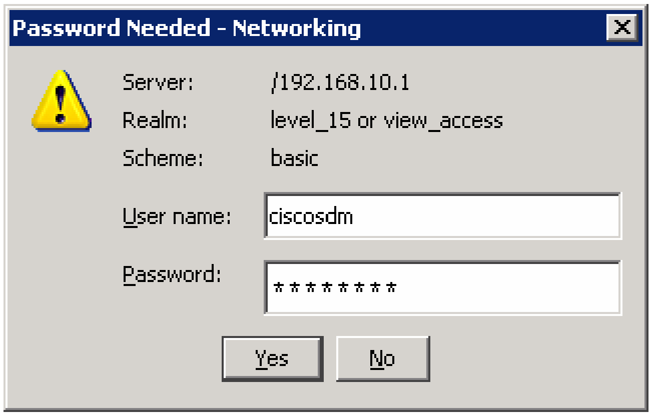
The name of the site does not match the name on the certificate. Do you want to run the application?

Name: 192.168.10.1

Publisher: IOS-Self-Signed-Certificate-1455051929

Figure 8-2: Internet Explorer Security Confirmation

Enter in the username and password you configured in step 2.



Password Needed - Networking

Server: /192.168.10.1

Realm: level_15 or view_access

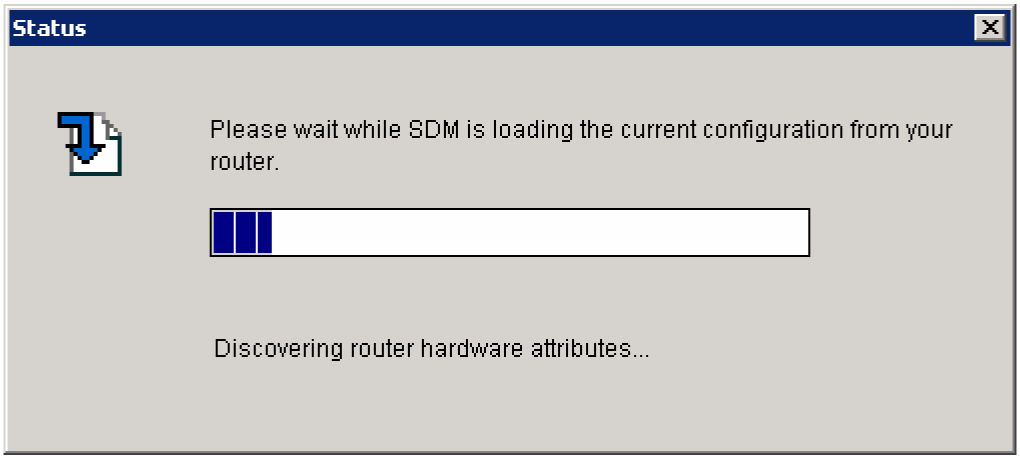
Scheme: basic

User name: ciscosdm

Password: ********

Figure 8-3: SDM Authentication Dialog

SDM will read the configuration off the router.



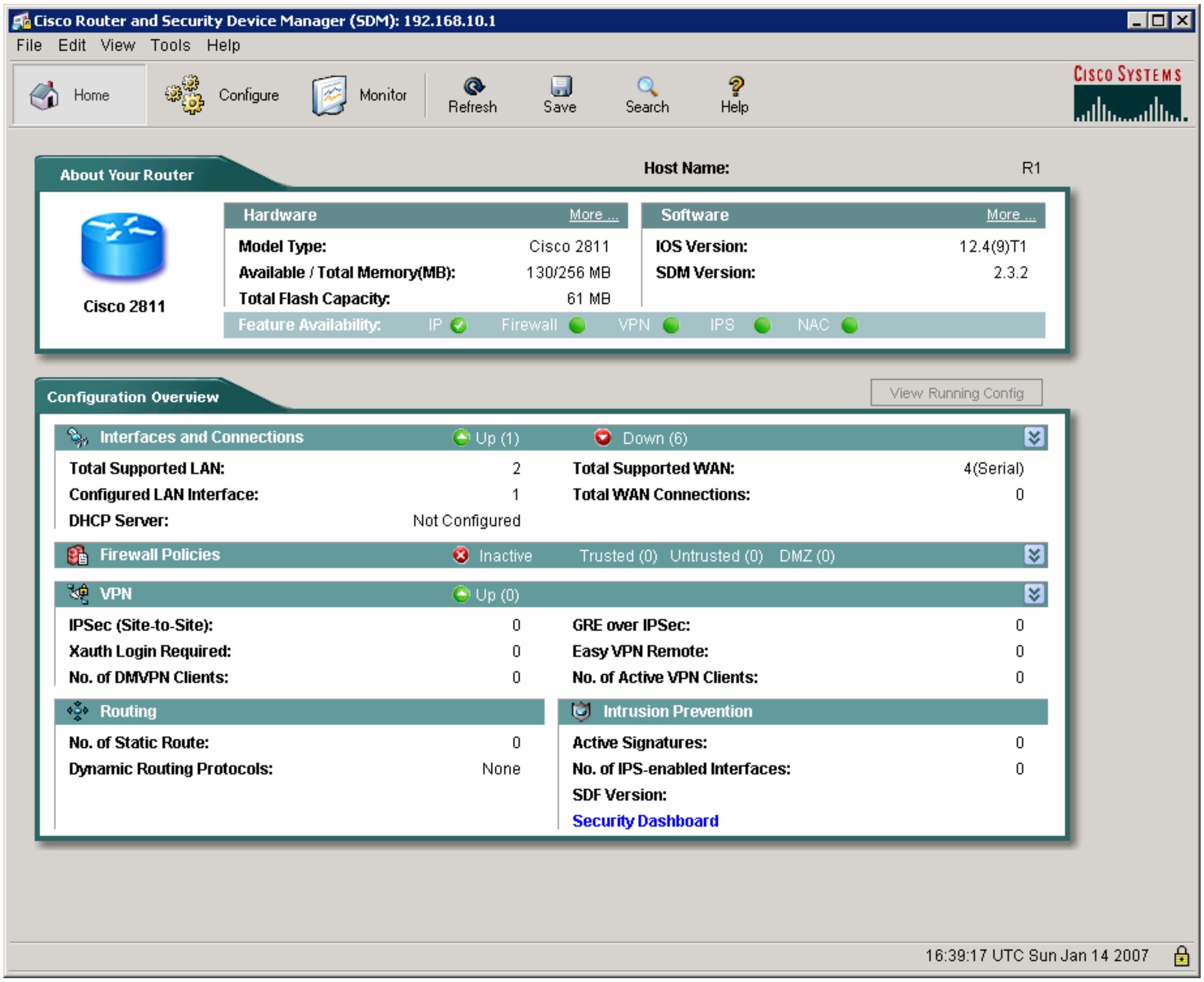
Status

Please wait while SDM is loading the current configuration from your router.

Discovering router hardware attributes...

Figure 8-4: SDM Load Progress Indicator

Once SDM is finished loading the current configuration of your router, the SDM homepage appears. If your configuration here looks correct, it means you have successfully configured and connected to SDM. What you see may differ from what appears in the following figure depending upon router model number, IOS version, and so forth.


Cisco Router and Security Device Manager (SDM): 192.168.10.1
File Edit View Tools Help
Home
Configure
Monitor
Refresh
Save
Search
Help
Cisco Systems

About Your Router

Host Name: R1

Cisco 2811

Hardware	More ...
Model Type:	Cisco 2811
Available / Total Memory(MB):	130/256 MB
Total Flash Capacity:	61 MB

Software	More ...
IOS Version:	12.4(9)T1
SDM Version:	2.3.2

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Configuration Overview

View Running Config

Interfaces and Connections Up (1) Down (6)

Total Supported LAN:	2	Total Supported WAN:	4(Serial)
Configured LAN Interface:	1	Total WAN Connections:	0
DHCP Server:	Not Configured		

Firewall Policies Inactive Trusted (0) Untrusted (0) DMZ (0)

VPN Up (0)

IPSec (Site-to-Site):	0	GRE over IPSec:	0
Xauth Login Required:	0	Easy VPN Remote:	0
No. of DMVPN Clients:	0	No. of Active VPN Clients:	0

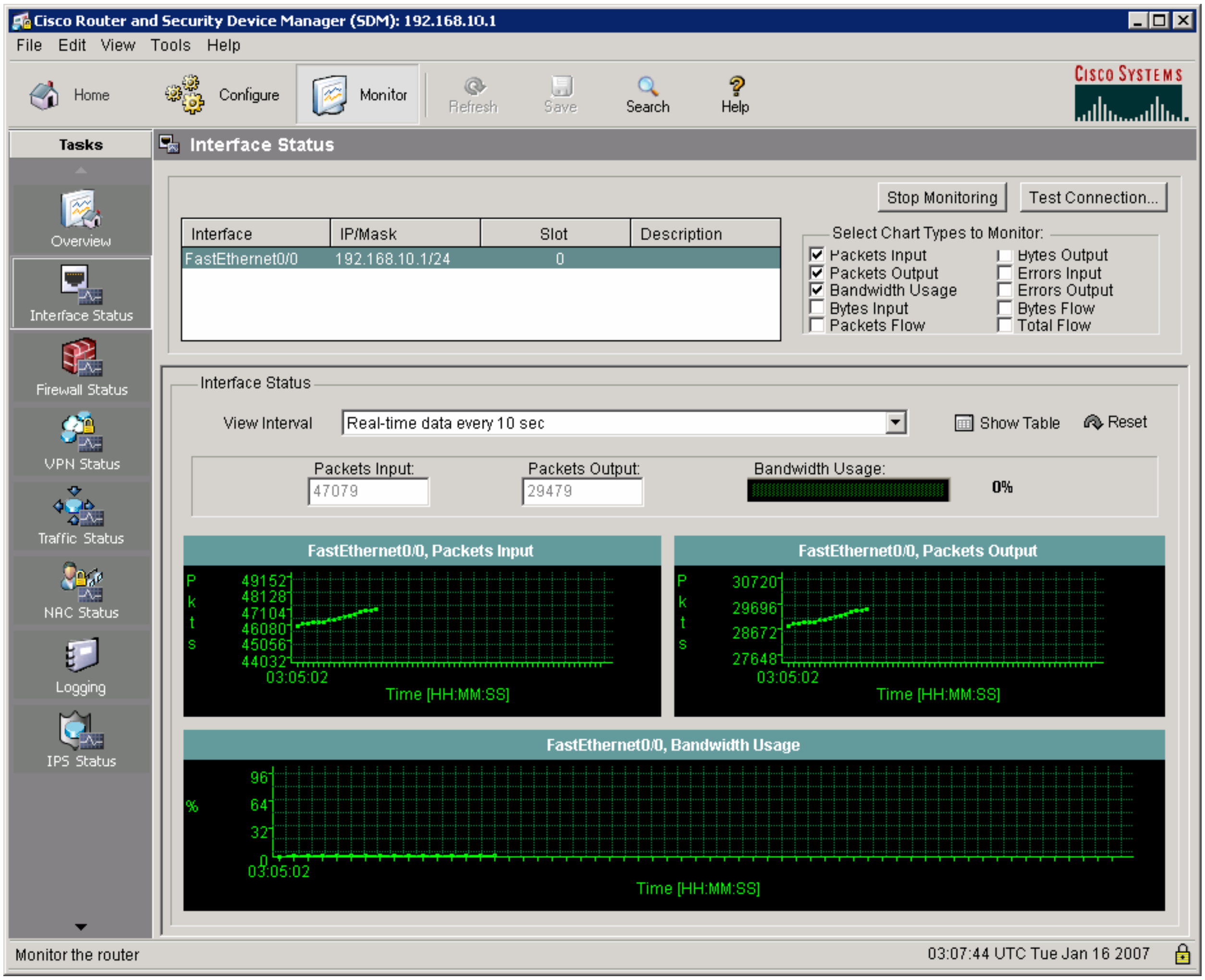
Routing		Intrusion Prevention	
No. of Static Route:	0	Active Signatures:	0
Dynamic Routing Protocols:	None	No. of IPS-enabled Interfaces:	0
		SDF Version:	
		Security Dashboard	

16:39:17 UTC Sun Jan 14 2007

Figure 8-5: SDM Dashboard

Step 9: Monitor an Interface in SDM

In SDM, you can look at an interface to verify that SDM is working and communicating with the router properly. To do this, click the **Monitor** tab at the top, and then click **Interface Status** on the left sidebar. You should see the graphs start to populate when FastEthernet0/0 is selected.



Interface	IP/Mask	Slot	Description
FastEthernet0/0	192.168.10.1/24	0	

Select Chart Types to Monitor:

☑ Packets Input	□ Bytes Output
☑ Packets Output	□ Errors Input
☑ Bandwidth Usage	□ Errors Output
□ Bytes Input	□ Bytes Flow
□ Packets Flow	□ Total Flow

Interface Status

View Interval: Real-time data every 10 sec

Packets Input: 47079

Packets Output: 29479

Bandwidth Usage: 0%

Monitor the router

03:07:44 UTC Tue Jan 16 2007

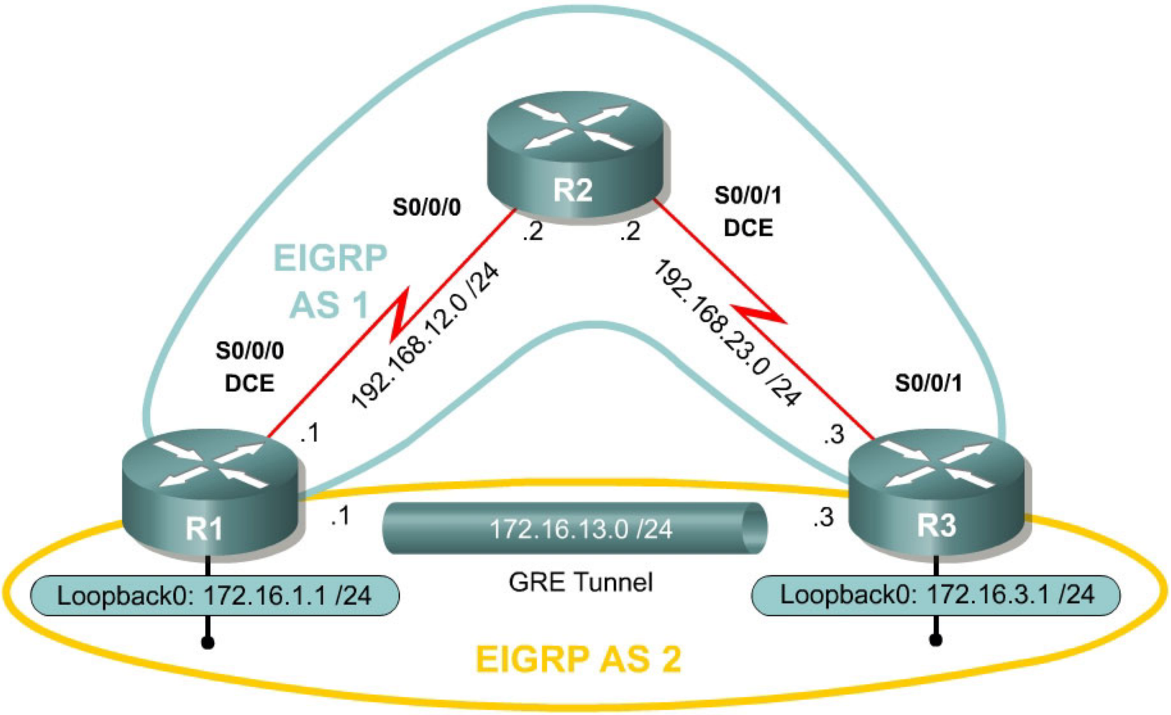
Figure 9-1: SDM Dashboard

Lab 3.2 Configuring a Basic GRE Tunnel

Learning Objectives

- Configure a GRE tunnel
- Configure EIGRP on a router
- Configure and test routing over the tunnel interfaces

Topology Diagram



Scenario

This lab is designed as an introduction to tunnels. In later labs you will configure more advanced tunnels using encryption, but this lab shows the basic mechanics of GRE tunnels.

Step 1: Configure Loopbacks and Physical Interfaces

Configure the loopback interfaces with the addresses shown in the diagram. Also configure the serial interfaces shown in the diagram. Do not forget to set the clockrates on the appropriate interfaces and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command. Do not set up the tunnel interface until the next step.

```
R1(config)# interface loopback 0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface serial 0/0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# no shutdown

R2(config)# interface serial 0/0/0
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3(config)# interface loopback 0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Configure EIGRP AS 1

Configure EIGRP AS 1 for the major networks 192.168.12.0/24 and 192.168.23.0/24. Do not include the networks in the diagram falling in the 172.16.0.0/16 range. The Class C networks will serve as the transit networks for the tunnel network. Make sure you disable EIGRP automatic summarization.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 192.168.12.0

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 192.168.12.0
R2(config-router)# network 192.168.23.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
R3(config-router)# network 192.168.23.0
```

Verify that R1 and R3 can see the remote transit network with the **show ip route** command. If they cannot see the remote transit network, troubleshoot. R2 will not learn any new routes because it is directly connected to both networks.

```
R1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, Serial0/0/0
     172.16.0.0/24 is subnetted, 1 subnets
```

```
C       172.16.1.0 is directly connected, Loopback0
D    192.168.23.0/24 [90/2681856] via 192.168.12.2, 00:00:15, Serial0/0/0

R2# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, Serial0/0/0
C    192.168.23.0/24 is directly connected, Serial0/0/1

R3# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

D    192.168.12.0/24 [90/2681856] via 192.168.23.2, 00:00:36, Serial0/0/1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.3.0 is directly connected, Loopback0
C    192.168.23.0/24 is directly connected, Serial0/0/1
```

Step 3: Configure a GRE Tunnel

Tunnels allow connectivity between remote areas of a network to communicate via a common network protocol and link independent of the native network protocol or routing protocol of their interconnection. For instance, consider a company with two locations in which each of the sites connects directly to the Internet with a static IP address. In order to allow private connections between the two sites, you could easily configure a tunnel between the two remote IP addresses so that private and/or encrypted communications could be sent between the two sites.

In this scenario, router R2 represents the agency providing connectivity between the two sites. R1 and R3 represent the remote sites. A tunnel will allow R1 and R3 to have a virtual private network (VPN) with each other and route between them. This type of VPN built on GRE encapsulation is not encrypted by default, but can be encrypted through simple configuration techniques.

When this configuration is complete, R2 does not need to be informed of the private networks behind R1 or R3, but simply passes IP data traffic between them based on the IP addresses on the packets it is sent. Since tunneled traffic is encapsulated within another IP header in this situation, R2 makes routing decisions based on the outermost IP header only. By running a routing protocol over the tunnel between the two sites, you can ensure that remote sites dynamically learn which remote IP networks are accessible to them.

In this lab, you use a tunnel to establish a VPN between the routers, and then route traffic between the remote sites using the tunnel interface. You will be using a base configuration without any encryption, although we will use encryption in later labs. In a production network, you would not want to send private network information through the public internet unencrypted because traffic sniffers along the way are easily able to read unencrypted data traffic.

A tunnel is a logical interface that acts as a logical connection between two endpoints. It is similar to a loopback interface in that it is a *virtual* interface created in software, but not represented by a hardware device. It is different than a loopback interface, however, in that more than one router is involved. You must configure each of the routers at the endpoints of a tunnel with a tunnel interface. GRE stands for generic routing encapsulation, and it is the simplest type of tunnel you can configure.

From global configuration mode, issue the **interface tunnel** *number* command. For simplicity, use tunnel number 0 on both routers. Next, configure an IP address with **ip address** *address* ***subnet-mask***, just like you would do on any other interface. This IP address is used inside the tunnel, part of the private network between R1 and R3.

Finally, assign a source and destination address for the tunnel with **tunnel source** *address* and **tunnel destination** *address*, respectively. Source can also be specified by interface. These addresses specify the endpoints of the router. GRE traffic will be encapsulated out of the serial address and deencapsulated on the remote destination serial address. We do not need to configure a tunnel mode because the default tunnel mode is GRE.

```
R1(config)# int tunnel0
R1(config-if)# tunnel source serial0/0/0
R1(config-if)# tunnel destination 192.168.23.3
R1(config-if)# ip address 172.16.13.1 255.255.255.0

R3(config)# int tunnel0
R3(config-if)# tunnel source serial0/0/1
R3(config-if)# tunnel destination 192.168.12.1
R3(config-if)# ip address 172.16.13.3 255.255.255.0
```

Verify that you can **ping** across the tunnel to the other side. If you can do this, you have successfully set up the tunnel.

```
R1# ping 172.16.13.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms
```

```
R3# ping 172.16.13.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/72 ms
```

1. What is the GRE header source address of the packet?

2. What is the GRE header destination address of the packet?

3. What is the source address of the packet encapsulated in the GRE tunnel?

4. What is the destination address of the packet encapsulated in the GRE tunnel?

5. Are these packets encrypted using the commands you entered?

Step 4: Routing EIGRP AS 2 over the Tunnel

Now that you have the tunnel set up, you can set up dynamic routing protocols over it. When the next hop address of a destination network is through the tunnel, the packet is encapsulated in an IP packet as described in the previous step.

Configure EIGRP AS 2 to route the entire 172.16.0.0 major network over the tunnel, but disable automatic summarization. Remember that R2 is not participating in this routing process so it will not need to be configured.

```
R1(config)# router eigrp 2
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.0.0

R3(config)# router eigrp 2
R3(config-router)# no auto-summary
R3(config-router)# network 172.16.0.0
```

You should see EIGRP neighbors come up with their messages logged to the console. Now issue the **show ip eigrp neighbors 2** command on R1 and R3. The ‘2’ at the end of the command string specifies the AS number. If you omit this, you will get neighbor tables for both EIGRP processes.

```
R1# show ip eigrp neighbors 2
IP-EIGRP neighbors for process 2
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.13.3             Tu0               10 00:01:14  100  5000  0  3

R3# show ip eigrp neighbors 2
IP-EIGRP neighbors for process 2
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.13.1             Tu0               13 00:02:47 1608  5000  0  2
```

Notice that the neighbor adjacencies are formed over the tunnel interface, even though no physical connection between the two routers exists. If you issue the **show ip route** command on the three routers, you see that R1 and R3 see each others loopbacks. Even though R2 is in the physical path, it has no knowledge of the loopback networks.

```
R1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, Serial0/0/0
     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.13.0 is directly connected, Tunnel0
C       172.16.1.0 is directly connected, Loopback0
D       172.16.3.0 [90/297372416] via 172.16.13.3, 00:04:23, Tunnel0
D    192.168.23.0/24 [90/2681856] via 192.168.12.2, 03:06:16, Serial0/0/0

R2# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, Serial0/0/0
C    192.168.23.0/24 is directly connected, Serial0/0/1

R3# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

D    192.168.12.0/24 [90/2681856] via 192.168.23.2, 03:06:54, Serial0/0/1
     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.13.0 is directly connected, Tunnel0
D       172.16.1.0 [90/297372416] via 172.16.13.1, 00:05:12, Tunnel0
C       172.16.3.0 is directly connected, Loopback0
C    192.168.23.0/24 is directly connected, Serial0/0/1
```

You will also be able to **ping** the remote loopback addresses from R1 and R3. R2 will not be able to ping either, because no route to the 172.16.0.0 network exists in its routing table.

```
R1# ping 172.16.3.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/68 ms

R2# ping 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2# ping 172.16.3.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R3# ping 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/68 ms
```

Why can’t R2 ping 172.16.1.1 or 172.16.3.1?

Appendix A: TCL Script Output

```
tclsh

foreach address {
```

```
172.16.1.1
172.16.3.1
172.16.13.1
172.16.13.3
192.168.12.1
192.168.12.2
192.168.23.2
192.168.23.3
} {
ping $address }

R1# tclsh
R1(tcl)#
R1(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#172.16.3.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#192.168.23.3
+>(tcl)#} {
+>(tcl)#ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
R1(tcl)#tclquit

R2#tclsh
R2(tcl)#
R2(tcl)#foreach address {
+>(tcl)#172.16.1.1
```

```
+>(tcl)#172.16.3.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#192.168.23.3
+>(tcl)#} {
+>(tcl)#ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/56/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/36 ms
R2(tcl)#tclquit

R3#tclsh
R3(tcl)#
R3(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#172.16.3.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#192.168.23.3
+>(tcl)#} {
+>(tcl)#ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/64 ms
R3(tcl)#tclquit
```

Final Configurations

```
R1# show run
hostname R1
!
interface Tunnel0
 ip address 172.16.13.1 255.255.255.0
 tunnel source Serial0/0/0
 tunnel destination 192.168.23.3
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.12.1 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 192.168.12.0
 no auto-summary
!
router eigrp 2
 network 172.16.0.0
 no auto-summary
!
end

R2# show run
hostname R2
!
interface Serial0/0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
```

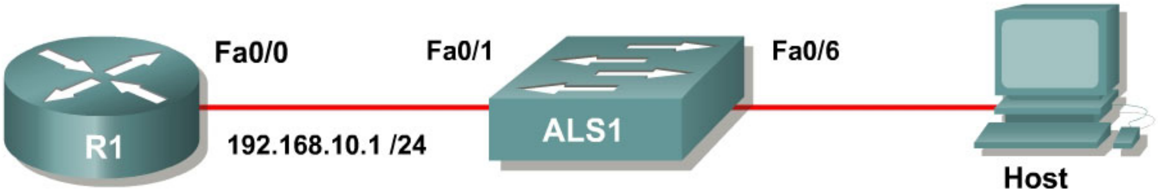
```
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 192.168.12.0
 network 192.168.23.0
 no auto-summary
!
end

R3# show run
hostname R3
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Tunnel0
 ip address 172.16.13.3 255.255.255.0
 tunnel source Serial0/0/1
 tunnel destination 192.168.12.1
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 no shutdown
!
router eigrp 1
 network 192.168.23.0
 no auto-summary
!
router eigrp 2
 network 172.16.0.0
 no auto-summary
!
end
```

Lab 3.3 Configuring Wireshark and SPAN

Learning Objectives

- Install Wireshark on a host PC
- Configure a switch to use the SPAN monitoring tool.

Topology Diagram



Scenario

In this lab, you will configure a switch to mirror traffic from a certain port out to a destination port for analyzing. In addition, you will configure Wireshark on a host PC to monitor the mirrored traffic flow.

Wireshark is a packet sniffing application that can read and analyze the incoming packets. Because it is useful for troubleshooting and verification, Wireshark is used in many of the labs in this course.

Step 1: Configure the Router

Configure the R1 FastEthernet0/0 interface with the IP address shown in the diagram. On the switch ALS1, place all the ports in VLAN 1. Configure EIGRP AS 1 with the 192.168.10.0 network in order to generate traffic on the wire.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# router eigrp 1
R1(config-router)# network 192.168.10.0
```

What kind of packets would you expect R1 to send toward ALS1’s Fast Ethernet interface?

Step 2: Install Wireshark and WinPcap

Run the Wireshark installer executable file. If you do not have the installer, download it from http://www.wireshark.org. Once the installation wizard opens, click **Next**:

- Click **I Agree** to agree to the Wireshark license agreement.
- Use the default settings and click **Next**.
- On the Select Additional Tasks click **Next**.
- Use the default installation directory, and then click **Next**.
- Check the option to install WinPcap, because you will need it for Wireshark to work. If you plan to have non-administrators use Wireshark, choose the Services option also.
- The WinPcap installation wizard will now run.
- During the install, the WinPcap installer starts up if you selected the launch option earlier (see Figure 2-4). Agree to the license agreement by clicking the **I Agree** button.

After clicking **I Agree**, WinPcap installs.

- Click **Finish** to complete the WinPcap installation, and the Wireshark installer will continue.
- After the Wireshark installer has finished, click **Next** to go to the final screen.
- Check **Run Wireshark** if you want to run the program now, and then click **Finish**.

Step 3: Configure SPAN on a Switch

On the Catalyst switch, you need to configure Switched Port Analyzer (SPAN) to mirror traffic going in and out of the router port to the host port. To do this, use the **monitor session** *number* **source interface** *interface-type interface-number* command. This specifies the source interface that is the interface to be monitored. The destination interface is specified in a similar way using the **monitor session** *number* **destination interface** *interface-type interface-number* command. You must use the same session number in both lines, indicating that they are the same monitoring session.

```
ALS1(config)# monitor session 1 source interface fastethernet0/1
ALS1(config)# monitor session 1 destination interface fastethernet0/6
```

It is important to note that when an interface is a SPAN destination interface, the switch will not forward any frames at OSI Layer 2 or Layer 3 aside from those captured from the SPAN session. Thus, the destination port does not participate in Dynamic Trunk Protocol (DTP), VLAN Trunking Protocol (VTP), Cisco Discovery Protocol (CDP), Spanning-tree Protocol (STP), or EtherChannel negotiation protocols such as PAgP (Port Aggregation Protocol) or LACP (Link Aggregation Protocol). The only traffic sent out of the destination interface is the traffic from the SPAN session.

Verify the configuration using the **show monitor** command.

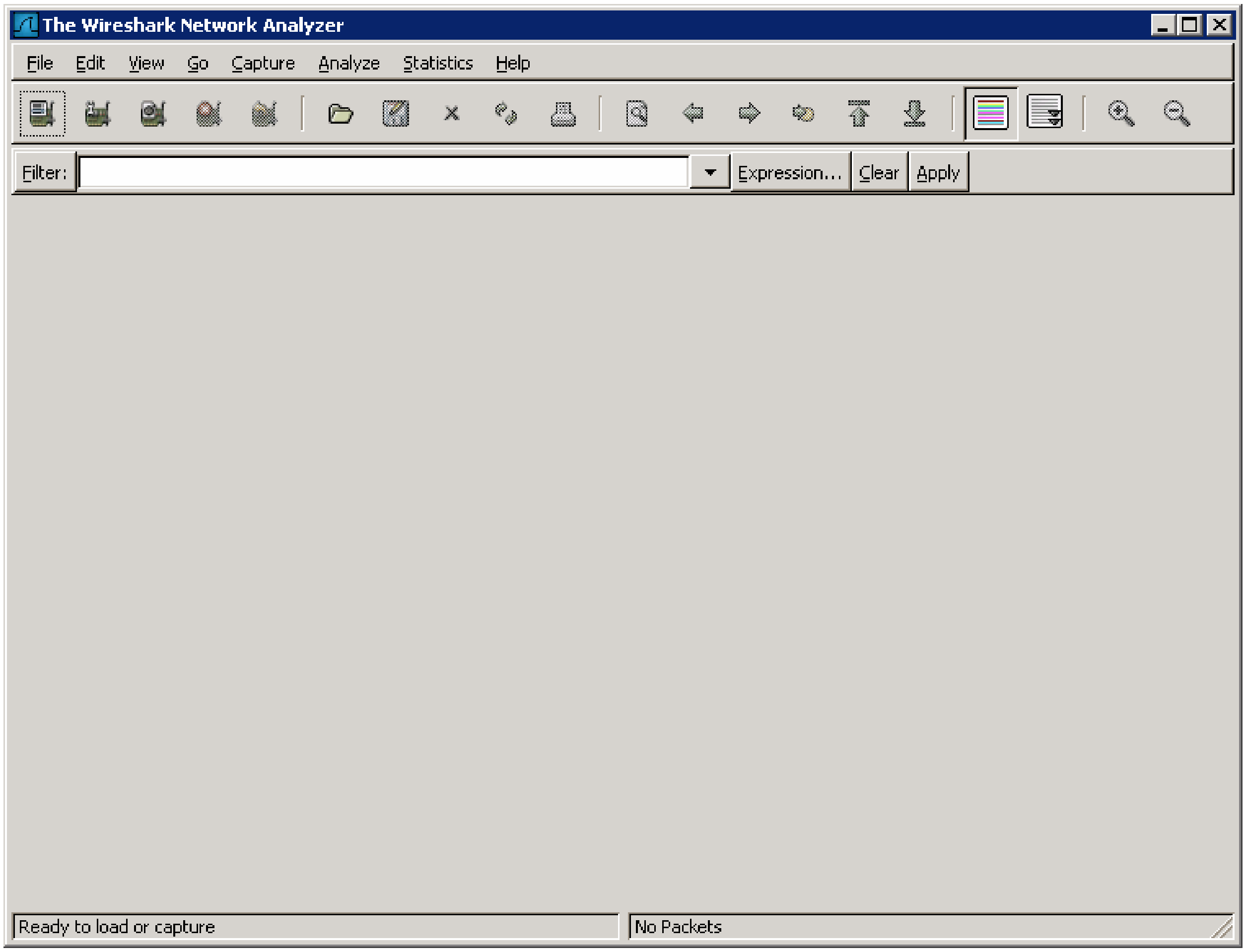
```
ALS1# show monitor
Session 1
---------
Type               : Local Session
Source Ports       :
    Both           : Fa0/1
Destination Ports : Fa0/6
    Encapsulation : Native
          Ingress : Disabled
```

If you had not implemented the following command, would the host still receive the EIGRP hello packets? Explain.

```
ALS1(config)# monitor session 1 destination interface fastethernet0/6
```

Step 4: Sniff Packets using Wireshark

Now that the switch is sending SPAN packets to the host, you can show packets generated from R1 in Wireshark. To do this, open Wireshark. It opens with an empty Wireshark window.

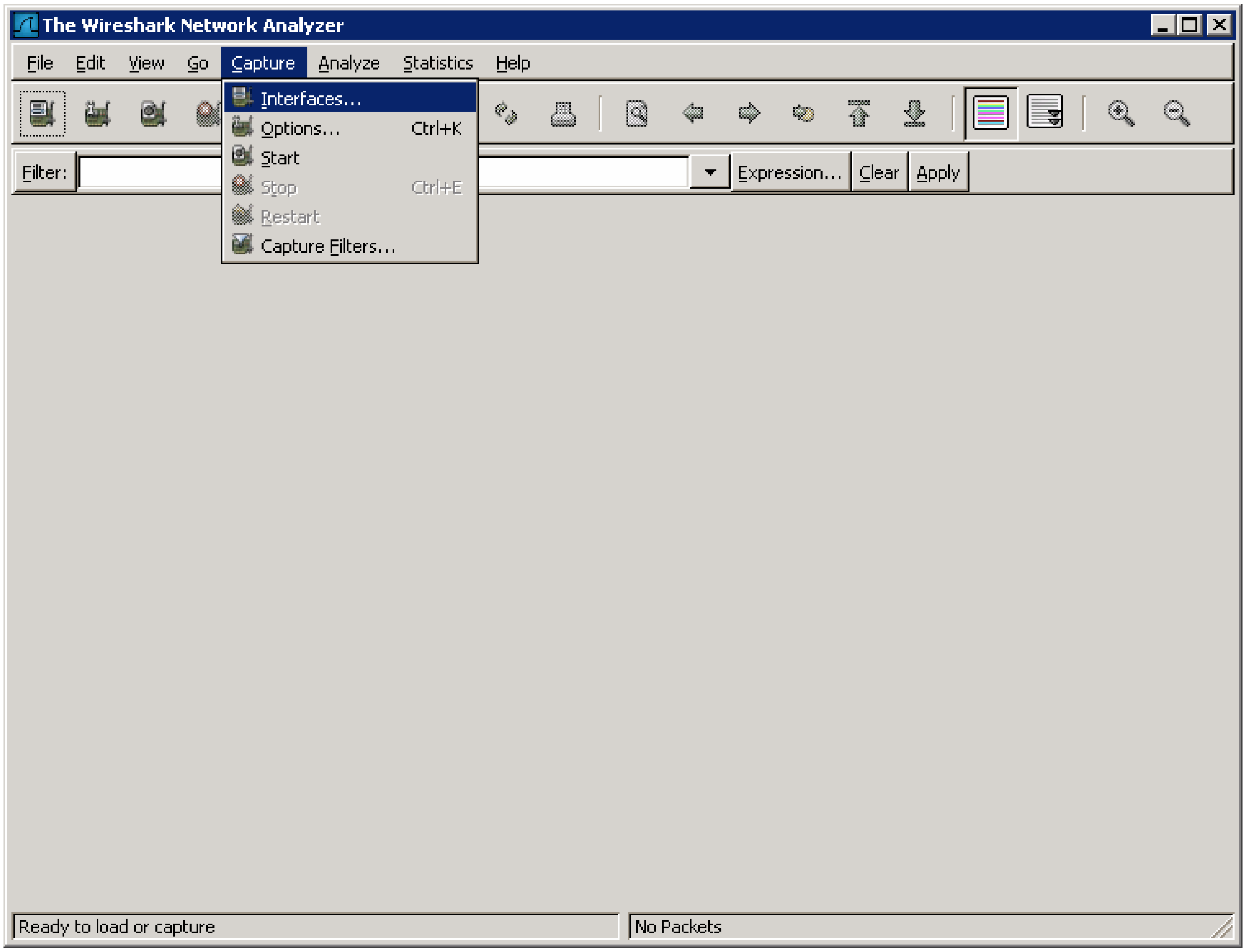


Ready to load or capture

No Packets

Figure 4-1: Wireshark Application Window

Click **Capture** on the toolbar, and then click **Interfaces...**.

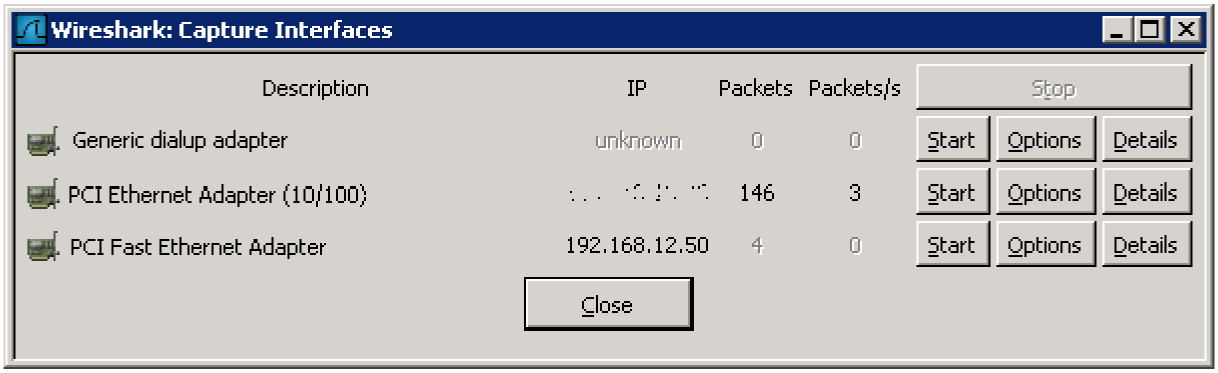

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Help
Interfaces...
Options... Ctrl+K
Start
Stop Ctrl+E
Restart
Capture Filters...
Filter:
Expression... Clear Apply

Ready to load or capture

No Packets

Figure 4-2: Capture Menu

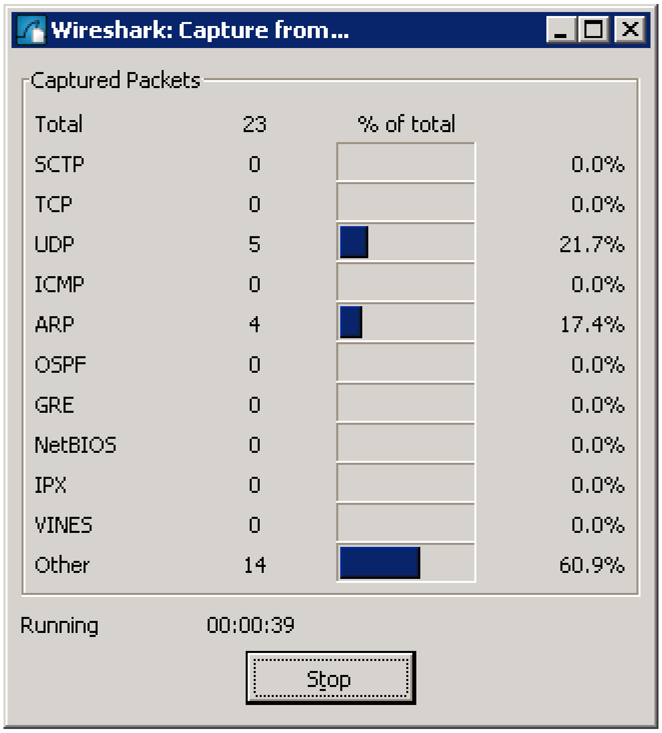
Choose the interface on the PC that is connected to the SPAN destination port and click **Start** for that interface. The IP on the host does not necessarily need to be in the same subnet as the traffic you are sniffing.



Description	IP	Packets	Packets/s			
Generic dialup adapter	unknown	0	0	Start	Options	Details
PCI Ethernet Adapter (10/100)	[illegible]	146	3	Start	Options	Details
PCI Fast Ethernet Adapter	192.168.12.50	4	0	Start	Options	Details

Figure 4-3: Interface List

Once you have sniffed a decent amount of traffic (~30 seconds), click **Stop**. EIGRP packets are classified as **Other** in this list.



Captured Packets

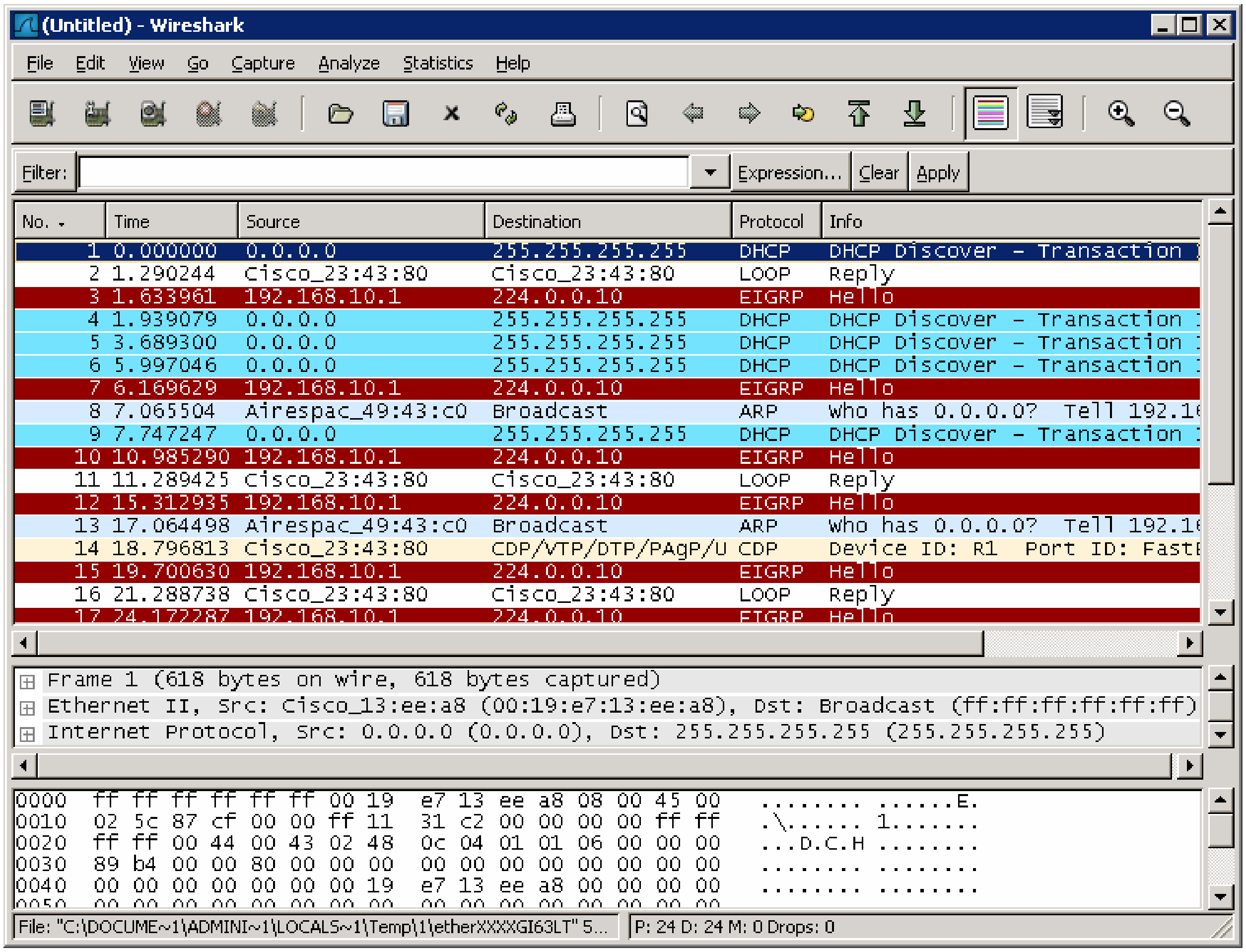
Total	23	% of total
SCTP	0	0.0%
TCP	0	0.0%
UDP	5	21.7%
ICMP	0	0.0%
ARP	4	17.4%
OSPF	0	0.0%
GRE	0	0.0%
NetBIOS	0	0.0%
IPX	0	0.0%
VINES	0	0.0%
Other	14	60.9%

Running 00:00:39

Figure 4-4: Capture Summary Window

Why are EIGRP packets not classified in any of these protocols?

Wireshark lists all captured packets. In addition, deeper packet information and a raw readout of the packet are available for the selected packet (see Figure 4-5.) Explore the detailed information available for each packet. Note the EIGRP hello multicasts are sent to the host via the SPAN session.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. .  Time       Source             Destination        Protocol  Info
     1 0.000000   0.0.0.0            255.255.255.255    DHCP      DHCP Discover - Transaction I
     2 1.290244   Cisco_23:43:80     Cisco_23:43:80     LOOP      Reply
     3 1.633961   192.168.10.1       224.0.0.10         EIGRP     Hello
     4 1.939079   0.0.0.0            255.255.255.255    DHCP      DHCP Discover - Transaction I
     5 3.689300   0.0.0.0            255.255.255.255    DHCP      DHCP Discover - Transaction I
     6 5.997046   0.0.0.0            255.255.255.255    DHCP      DHCP Discover - Transaction I
     7 6.169629   192.168.10.1       224.0.0.10         EIGRP     Hello
     8 7.065504   Airespac_49:43:c0  Broadcast          ARP       Who has 0.0.0.0?  Tell 192.16
     9 7.747247   0.0.0.0            255.255.255.255    DHCP      DHCP Discover - Transaction I
    10 10.985290  192.168.10.1       224.0.0.10         EIGRP     Hello
    11 11.289425  Cisco_23:43:80     Cisco_23:43:80     LOOP      Reply
    12 15.312935  192.168.10.1       224.0.0.10         EIGRP     Hello
    13 17.064498  Airespac_49:43:c0  Broadcast          ARP       Who has 0.0.0.0?  Tell 192.16
    14 18.796813  Cisco_23:43:80     CDP/VTP/DTP/PAgP/U CDP       Device ID: R1  Port ID: FastE
    15 19.700630  192.168.10.1       224.0.0.10         EIGRP     Hello
    16 21.288738  Cisco_23:43:80     Cisco_23:43:80     LOOP      Reply
    17 24.172287  192.168.10.1       224.0.0.10         EIGRP     Hello
```

```
⊞ Frame 1 (618 bytes on wire, 618 bytes captured)
⊞ Ethernet II, Src: Cisco_13:ee:a8 (00:19:e7:13:ee:a8), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
```

```
0000  ff ff ff ff ff ff 00 19  e7 13 ee a8 08 00 45 00   ........ ......E.
0010  02 5c 87 cf 00 00 ff 11  31 c2 00 00 00 00 ff ff   .\...... 1.......
0020  ff ff 00 44 00 43 02 48  0c 04 01 01 06 00 00 00   ...D.C.H ........
0030  89 b4 00 00 80 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 19  e7 13 ee a8 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\etherXXXXGI63LT" 5... | P: 24 D: 24 M: 0 Drops: 0

Figure 4-5: Wireshark Packet Detail Window

Final Configurations

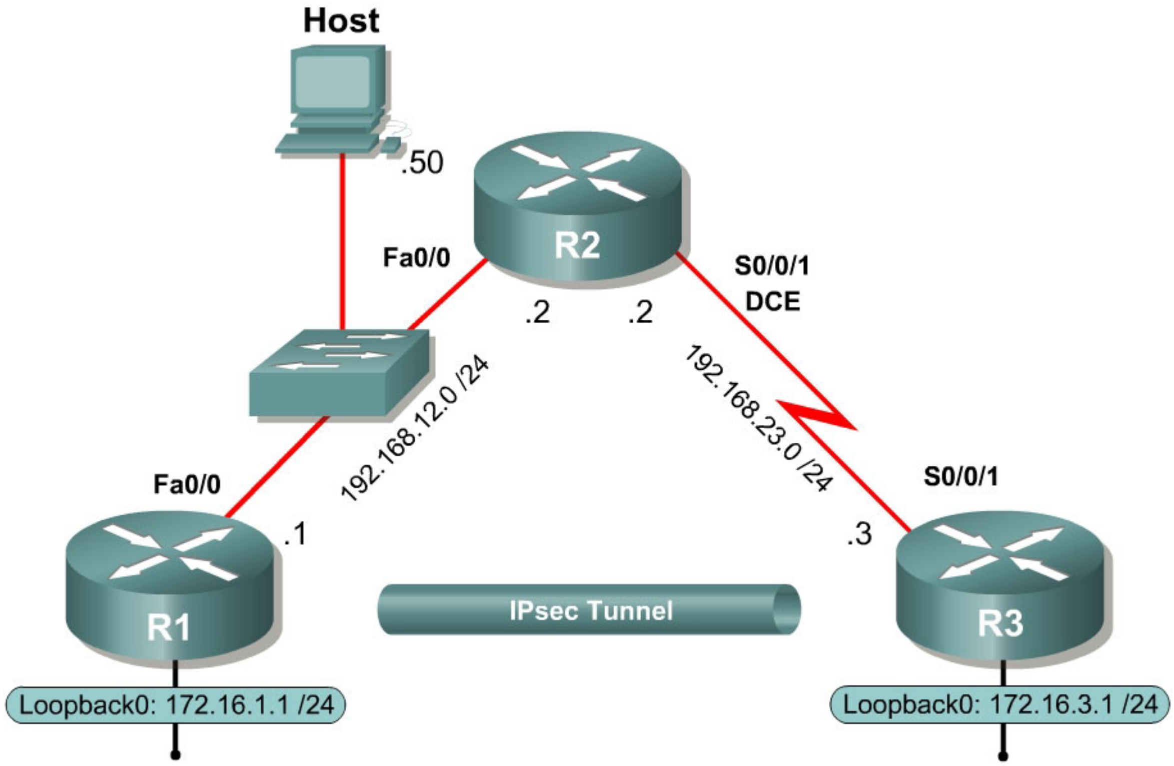
```
R1# show run
!
hostname R1
!
interface fastethernet0/0
 ip address 192.168.10.1 255.255.255.0
!
router eigrp 1
 network 192.168.10.0
!
end

ALS1# show run
!
hostname ALS1
!
monitor session 1 source interface fastethernet0/1
monitor session 1 destination interface fastethernet0/6
!
end
```

Lab 3.4 Configuring Site-to-Site IPsec VPNs with SDM

Learning Objectives

- Configure EIGRP on the routers
- Create a site-to-site IPsec VPN using SDM
- Verify IPsec operation

Topology Diagram



Scenario

In this lab, you will configure a site-to-site IPsec VPN. Once you have configured the VPN, the traffic between the loopback interfaces on R1 and R3 will be encrypted.

You will use the Cisco Security Device Manager (SDM) for this lab exercise. Lab 3.5 involves the same function as this exercise, but implemented via the command-line interface. Ensure that you are running Cisco IOS 12.4(6)T with Advanced IP services.

Step 1: Configure Addressing

Configure the loopback interfaces with the addresses shown in the diagram and configure the serial interfaces shown in the diagram. Set the clock rates on the appropriate interfaces and issue the **no shutdown** command on all physical connections. Verify that you have connectivity across local subnets using the **ping** command.

```
R1(config)# interface loopback0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface fastethernet0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown

R2(config)# interface fastethernet0/0
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/0/1
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3(config)# interface loopback0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface serial0/0/1
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Configure EIGRP

In order to maintain connectivity between remote networks, configure EIGRP to route between all networks in the diagram. Add all connected subnets into the EIGRP autonomous system on every router. Disable automatic summarization.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.0.0
R1(config-router)# network 192.168.12.0

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 192.168.12.0
R2(config-router)# network 192.168.23.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
R3(config-router)# network 172.16.0.0
R3(config-router)# network 192.168.23.0
```

Verify that you have full IP connectivity at this point using the following TCL script.

```
tclsh

foreach address {
172.16.1.1
192.168.12.1
192.168.12.2
192.168.23.2
172.16.3.1
192.168.23.3
} { ping $address }

tclquit
```

Compare your output with the output shown in Appendix A. Troubleshoot as necessary.

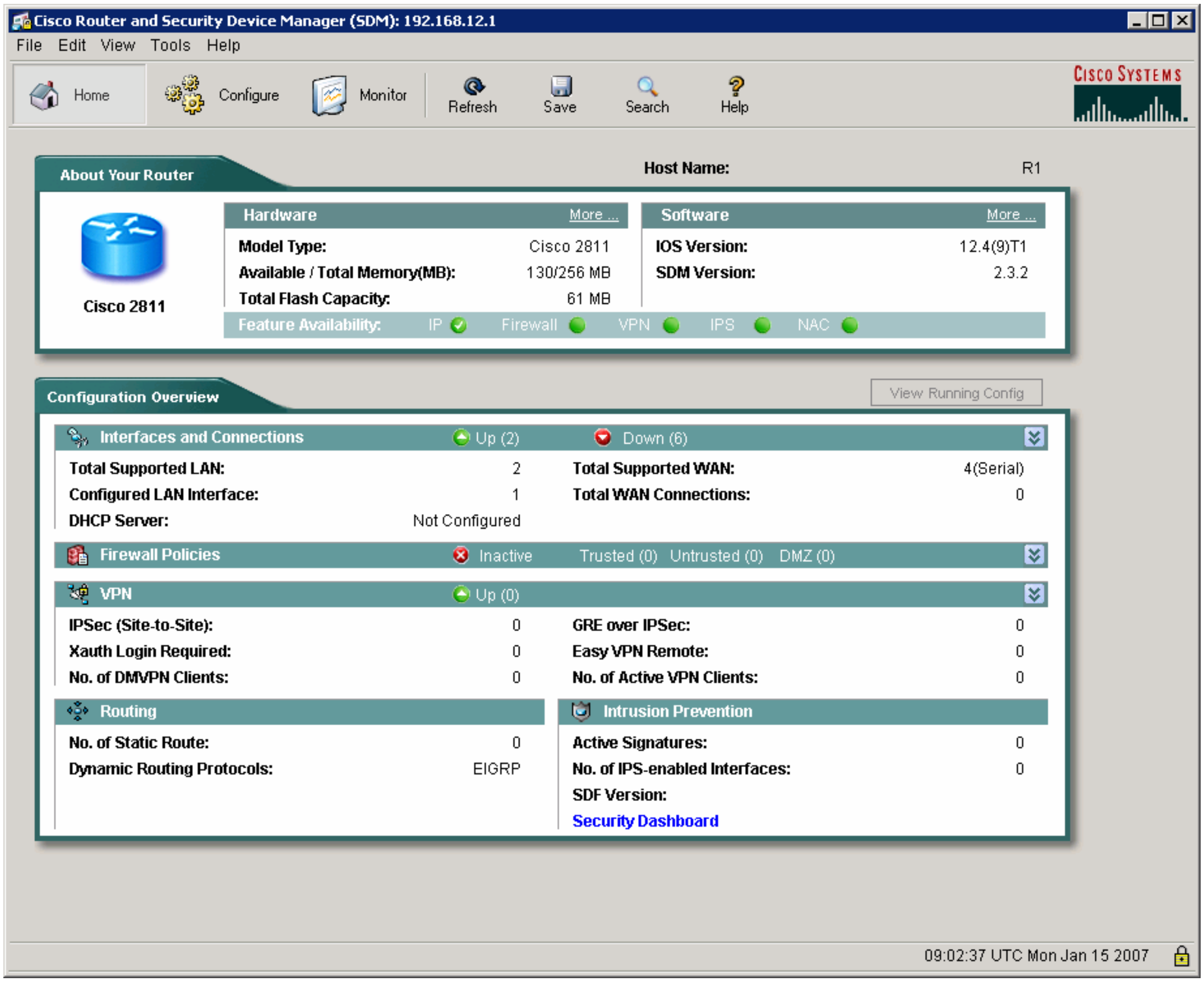
Step 3: Connect to the Routers via SDM

Configure the IP address shown in the diagram on the host PC and install SDM to either the router or the PC as shown in Lab 3.1. Ensure that the PC uses a default gateway to forward traffic to remote networks.

From the host, connect to the router using SDM. If you installed SDM application on the host, connect by launching the SDM application and connecting to 192.168.12.1. When you complete this step for R3, you will use 192.168.23.3 as the IP address.

The SDM home page is shown in the following figure. The page might be shown in an application window if it is installed on the host, or in an Internet Explorer window if it is being run from the router.

For information on how to configure SDM, refer to Lab 3.1: Configuring SDM on a Router.



Host Name: R1

Hardware	
Model Type:	Cisco 2811
Available / Total Memory(MB):	130/256 MB
Total Flash Capacity:	61 MB

Software	
IOS Version:	12.4(9)T1
SDM Version:	2.3.2

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Interfaces and Connections — Up (2) — Down (6)

Total Supported LAN:	2	Total Supported WAN:	4(Serial)
Configured LAN Interface:	1	Total WAN Connections:	0
DHCP Server:	Not Configured		

Firewall Policies — Inactive — Trusted (0) Untrusted (0) DMZ (0)

VPN — Up (0)

IPSec (Site-to-Site):	0	GRE over IPSec:	0
Xauth Login Required:	0	Easy VPN Remote:	0
No. of DMVPN Clients:	0	No. of Active VPN Clients:	0

Routing

No. of Static Route:	0
Dynamic Routing Protocols:	EIGRP

Intrusion Prevention

Active Signatures:	0
No. of IPS-enabled Interfaces:	0
SDF Version:	

Security Dashboard

09:02:37 UTC Mon Jan 15 2007

Figure 3-1: SDM Home Page

Step 4: Configure Site-to-Site IPsec VPN via SDM

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF). It provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices ("peers"), such as Cisco routers.

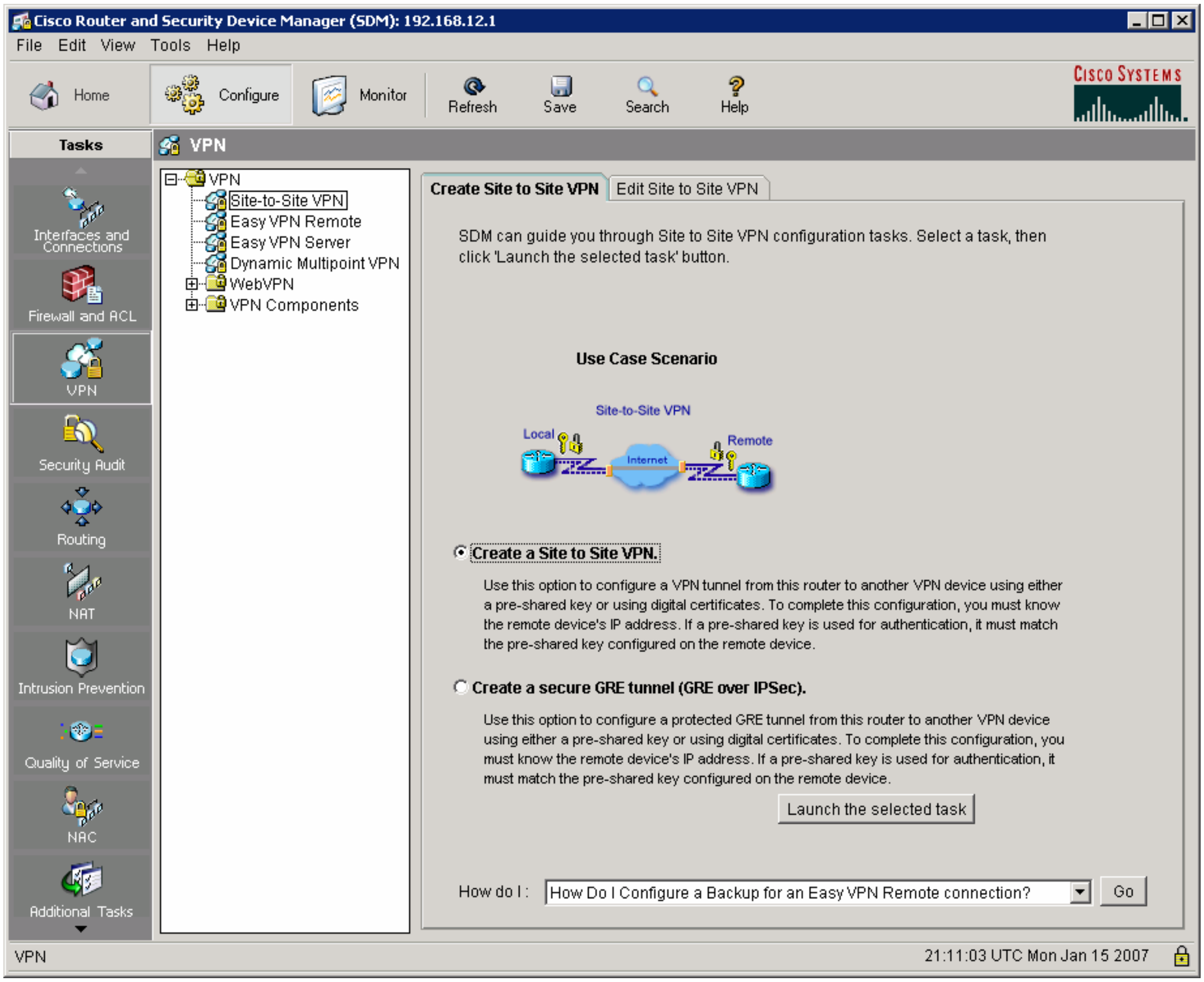
Since IPsec is a framework, it allows us to exchange security protocols as new technologies (including encryption algorithms) are developed.

There are two central configuration elements to the implementation of an IPsec VPN:

1. Implement Internet Key Exchange (IKE) parameters
2. Implement IPsec parameters

The exchange method employed by IKE is first used to pass and validate IKE policies between peers. Then, the peers exchange and match IPsec policies for the authentication and encryption of data traffic. The IKE policy controls the authentication, encryption algorithm, and key exchange method used for IKE proposals that are sent and received by the IPsec endpoints. The IPsec policy is used to encrypt data traffic sent through the VPN tunnel.

SDM contains a wizard that makes setting up site-to-site VPNs easier than using the command line interface. To access these settings, click the **Configure** heading at the top of the SDM window, below the menu bar. On the taskbar on the far left side of the window, choose **VPN**. In the VPN type list next to it, choose **Site-to-Site VPN**. After choosing the **Create a Site to Site VPN** tab in the main window, click **Launch the selected task** to begin the SDM Site-to-Site VPN wizard.



SDM can guide you through Site to Site VPN configuration tasks. Select a task, then click 'Launch the selected task' button.

● Create a Site to Site VPN.

Use this option to configure a VPN tunnel from this router to another VPN device using either a pre-shared key or using digital certificates. To complete this configuration, you must know the remote device's IP address. If a pre-shared key is used for authentication, it must match the pre-shared key configured on the remote device.

○ Create a secure GRE tunnel (GRE over IPSec).

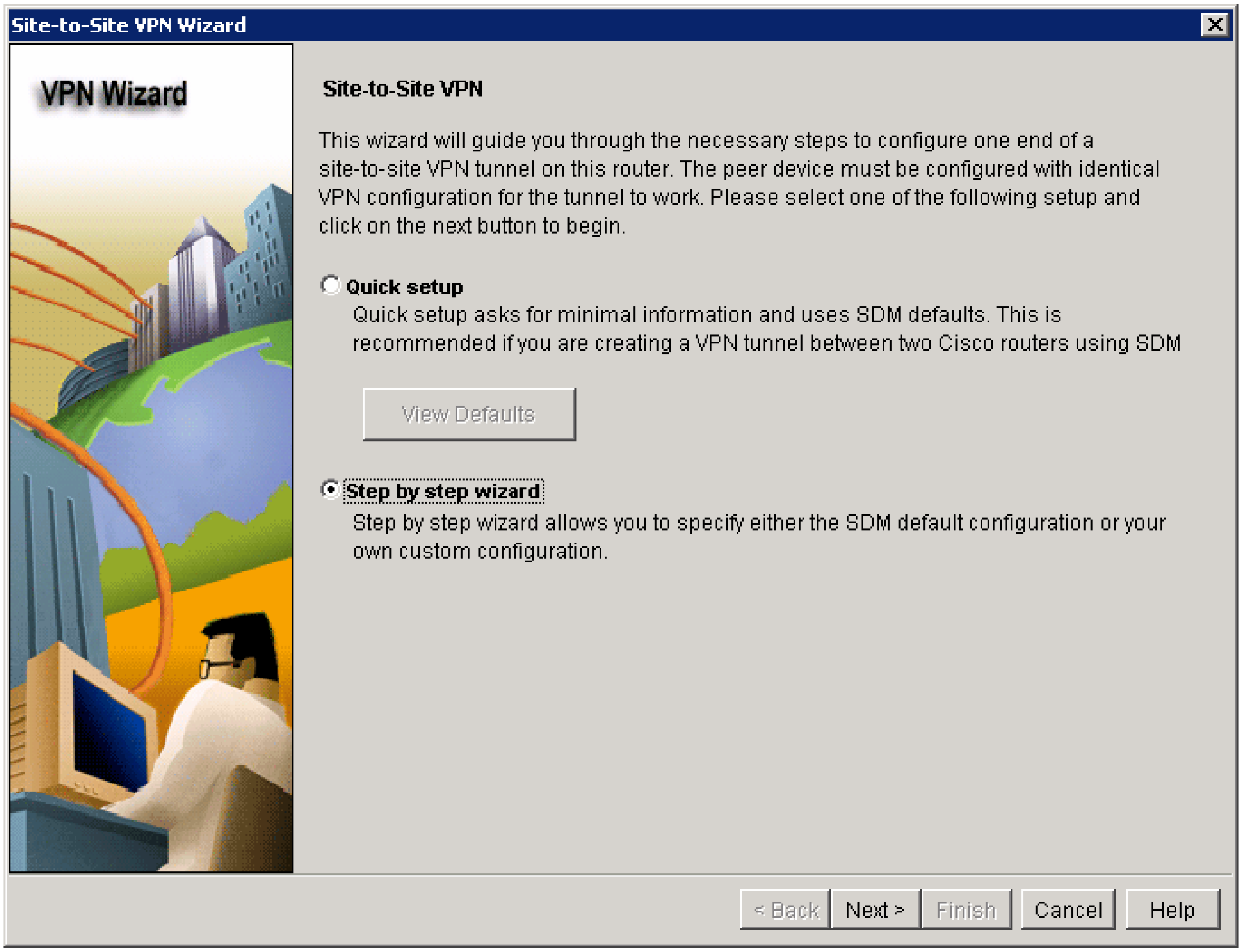
Use this option to configure a protected GRE tunnel from this router to another VPN device using either a pre-shared key or using digital certificates. To complete this configuration, you must know the remote device's IP address. If a pre-shared key is used for authentication, it must match the pre-shared key configured on the remote device.

How do I : How Do I Configure a Backup for an Easy VPN Remote connection?

21:11:03 UTC Mon Jan 15 2007

Figure 4-1: VPN Configuration Screen

At the next window, select **Step by step wizard,** and then click **Next**, so that you have more control over the VPN settings used. If you are in a hurry or don't care about specific VPN settings, you would use the **Quick setup** option.



Site-to-Site VPN Wizard

VPN Wizard

Site-to-Site VPN

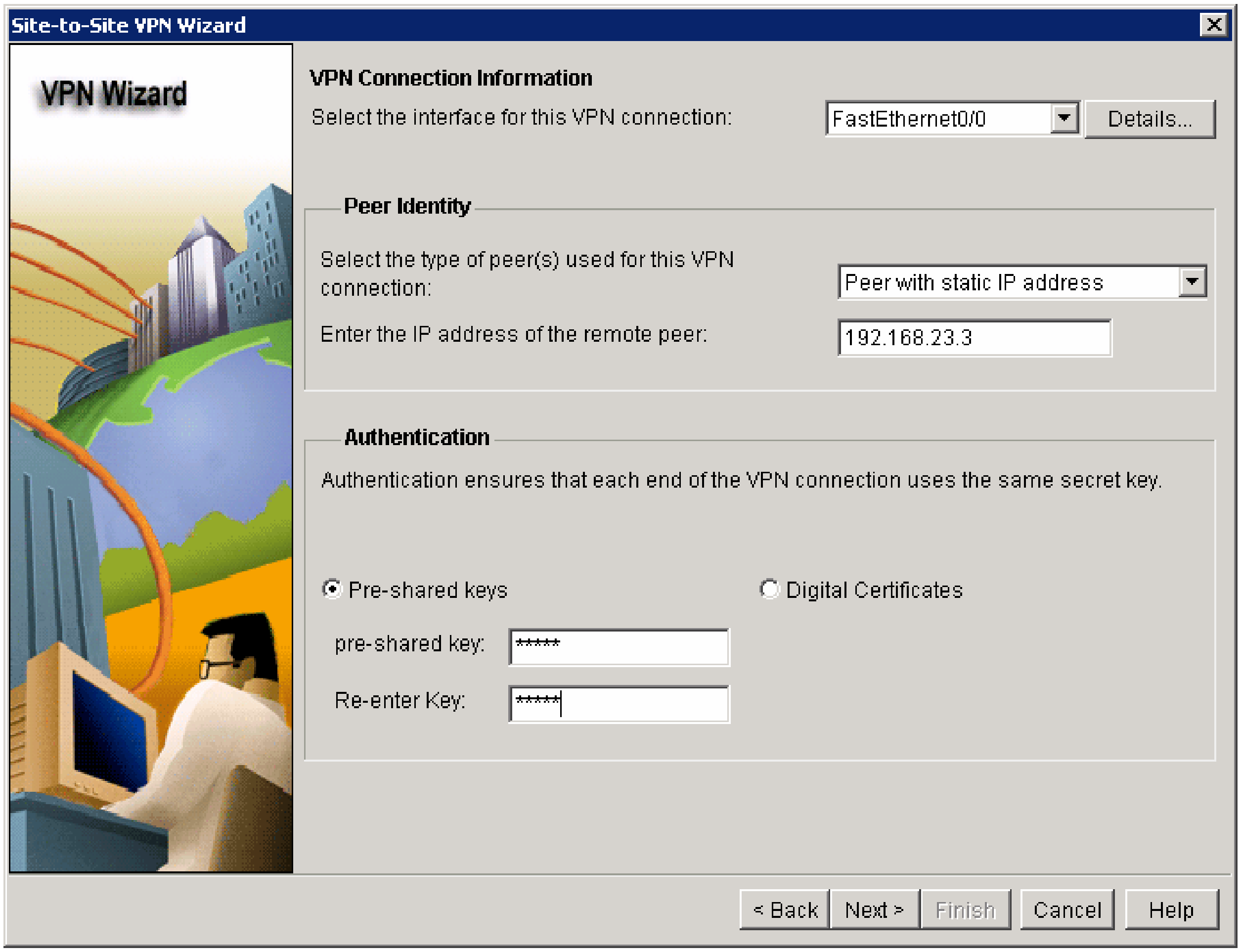
This wizard will guide you through the necessary steps to configure one end of a site-to-site VPN tunnel on this router. The peer device must be configured with identical VPN configuration for the tunnel to work. Please select one of the following setup and click on the next button to begin.

○ Quick setup
Quick setup asks for minimal information and uses SDM defaults. This is recommended if you are creating a VPN tunnel between two Cisco routers using SDM

● Step by step wizard
Step by step wizard allows you to specify either the SDM default configuration or your own custom configuration.

Figure 4-2: Site-to-Site VPN Wizard

At the next window, you can configure some of the basic site-to-site VPN settings. The interface option at the top indicates the outbound interface out of which R1 will send encrypted packets. In this lab topology, R1's outbound VPN interface is FastEthernet0/0. In the Peer Identity section, you select the peer type. Since you are using a static IP peer, you select that option and enter the IP address of the VPN destination. For authentication, click **Pre-shared keys**, and enter a VPN key. This key is what protects the VPN and keeps it secure, so in the real world you would want a secure key. Since this is just a lab, use "cisco" as your VPN key. You could also set up digital certificates as a more scalable solution. Digital certificates would require a more advanced set up, which is beyond the scope of this lab and the CCNP2 curriculum. Once you have entered these settings correctly, click **Next**.



Site-to-Site VPN Wizard

VPN Connection Information

Select the interface for this VPN connection: FastEthernet0/0

Peer Identity

Select the type of peer(s) used for this VPN connection: Peer with static IP address

Enter the IP address of the remote peer: 192.168.23.3

Authentication

Authentication ensures that each end of the VPN connection uses the same secret key.

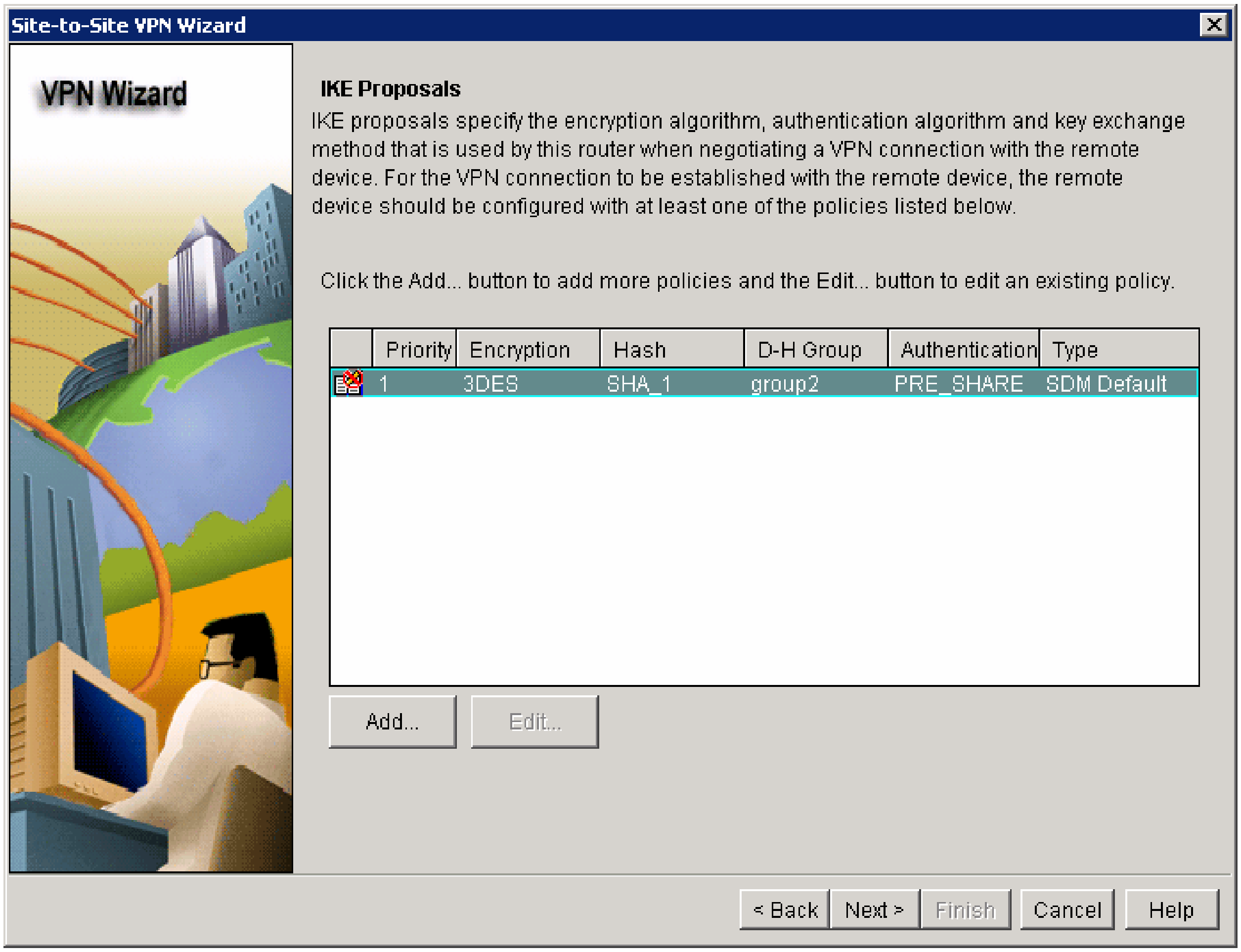
● Pre-shared keys ○ Digital Certificates

pre-shared key: *****

Re-enter Key: *****

Figure 4-3: VPN Connection and Authentication Information

On the next window you can edit the IKE proposals. One is already defined for you as an SDM default. Click **Add** to create your own.



IKE Proposals

IKE proposals specify the encryption algorithm, authentication algorithm and key exchange method that is used by this router when negotiating a VPN connection with the remote device. For the VPN connection to be established with the remote device, the remote device should be configured with at least one of the policies listed below.

Click the Add... button to add more policies and the Edit... button to edit an existing policy.

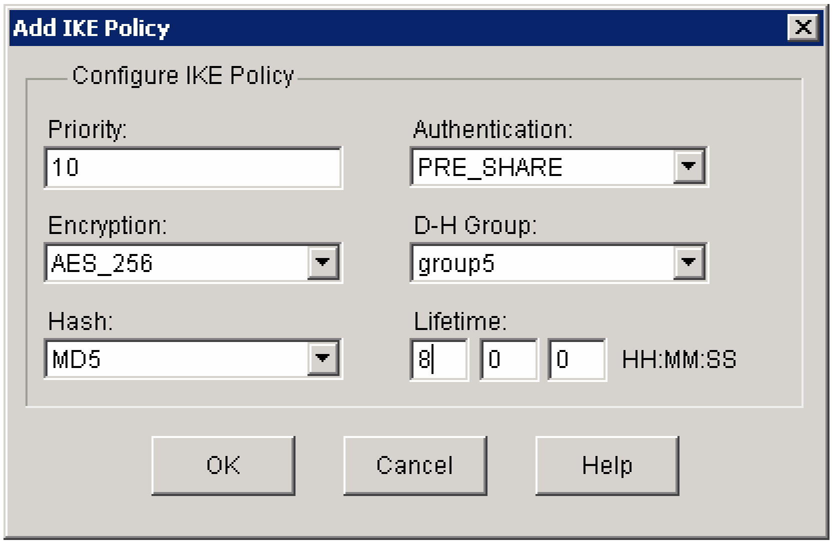
	Priority	Encryption	Hash	D-H Group	Authentication	Type
	1	3DES	SHA_1	group2	PRE_SHARE	SDM Default

Figure 4-4: IKE Proposals List

What function does this IKE proposal serve?

IKE policies are used while setting up the control channel between the two VPN endpoints for key exchange. This is also referred to as the IKE secure association (SA). In contrast, the IPsec policy is used during IKE Phase II to negotiate an IPsec security association to pass target data traffic.

Set up the security settings for this IKE policy as shown in the next figure. If your IOS image doesn't support all of the settings, configure what you can as long as your VPN settings match on both ends of the connection.



Priority: 10

Authentication: PRE_SHARE

Encryption: AES_256

D-H Group: group5

Hash: MD5

Lifetime: 8 0 0 HH:MM:SS

Figure 4-5: Add IKE Policy Dialog

The authentication type can either be pre-shared keys or digital certificates. The method of pre-shared keys involves manually typing a secret string on both VPN endpoints during the configuration process. The endpoints will later use that string as part of the authentication process. Make sure you set the authentication type to **PRE_SHARE** so that the pre-shared keys created earlier will work.

Each of the drop-down boxes shown has multiple protocols or algorithms that can be used to secure the control data.

What is the function of the encryption algorithm in the IKE policy?

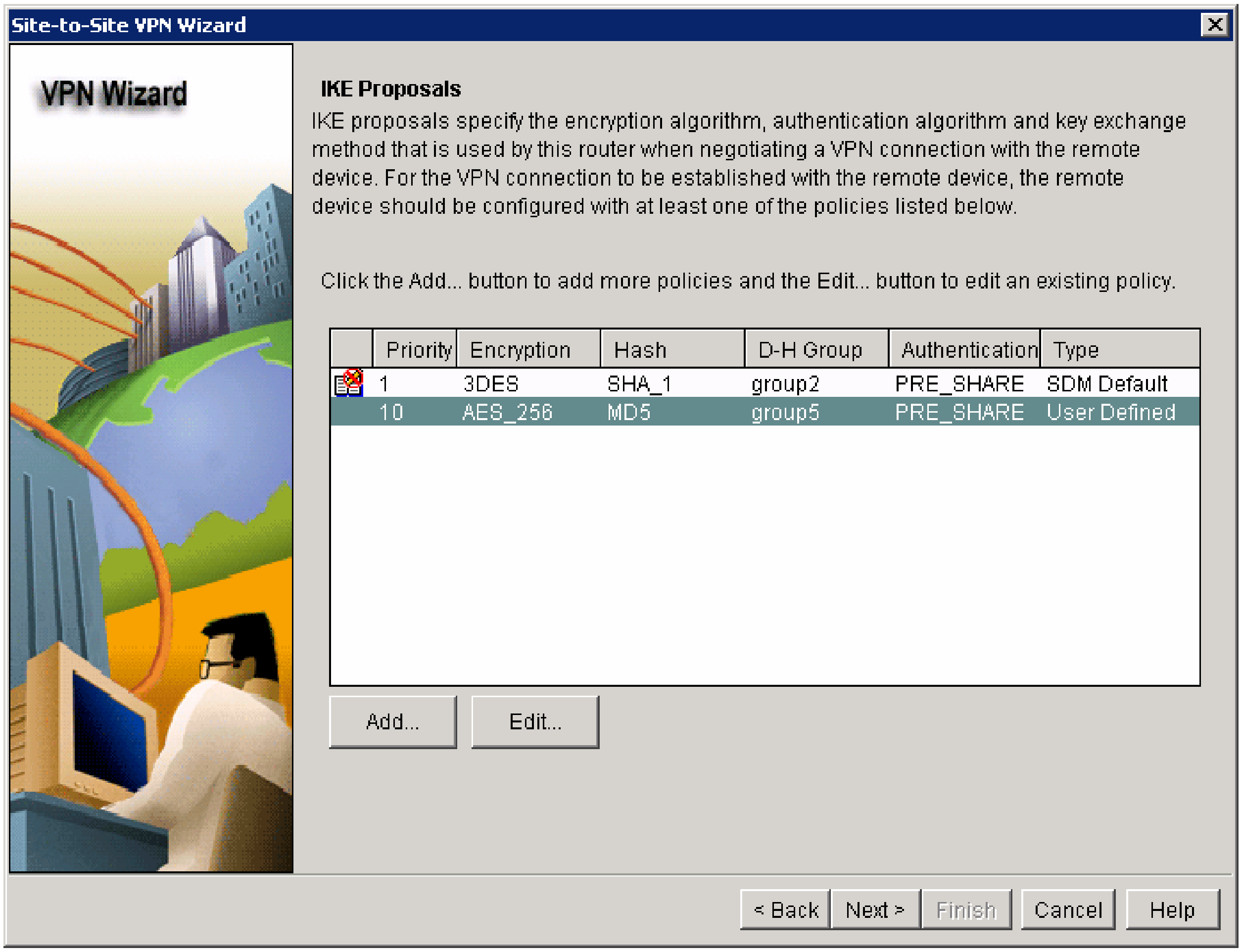
What is the purpose of the hash function?

What function does the authentication method serve?

How is the Diffie-Hellman group in the IKE policy used?

What event happens at the end of the IKE policy's lifetime?

Your new IKE proposal has been added to the list. Click **Next**.



IKE Proposals

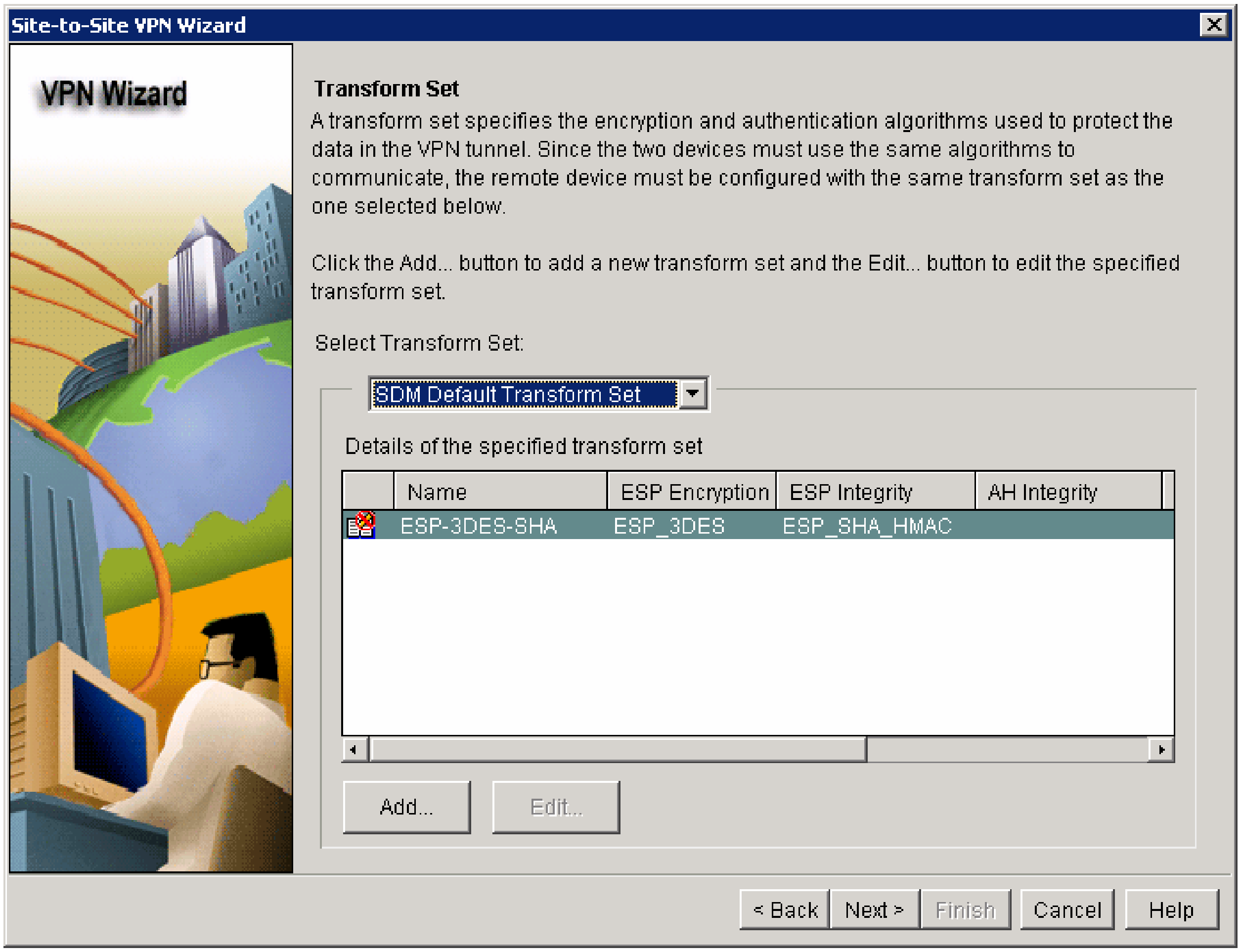
IKE proposals specify the encryption algorithm, authentication algorithm and key exchange method that is used by this router when negotiating a VPN connection with the remote device. For the VPN connection to be established with the remote device, the remote device should be configured with at least one of the policies listed below.

Click the Add... button to add more policies and the Edit... button to edit an existing policy.

	Priority	Encryption	Hash	D-H Group	Authentication	Type
	1	3DES	SHA_1	group2	PRE_SHARE	SDM Default
	10	AES_256	MD5	group5	PRE_SHARE	User Defined

Figure 4-6: IKE Proposals with Changes Applied

The next window allows you to add an IPsec transform set. Click **Add…** to bring up the **Add Transform Set** dialog.



Site-to-Site VPN Wizard

Transform Set

A transform set specifies the encryption and authentication algorithms used to protect the data in the VPN tunnel. Since the two devices must use the same algorithms to communicate, the remote device must be configured with the same transform set as the one selected below.

Click the Add... button to add a new transform set and the Edit... button to edit the specified transform set.

Select Transform Set: SDM Default Transform Set

Details of the specified transform set

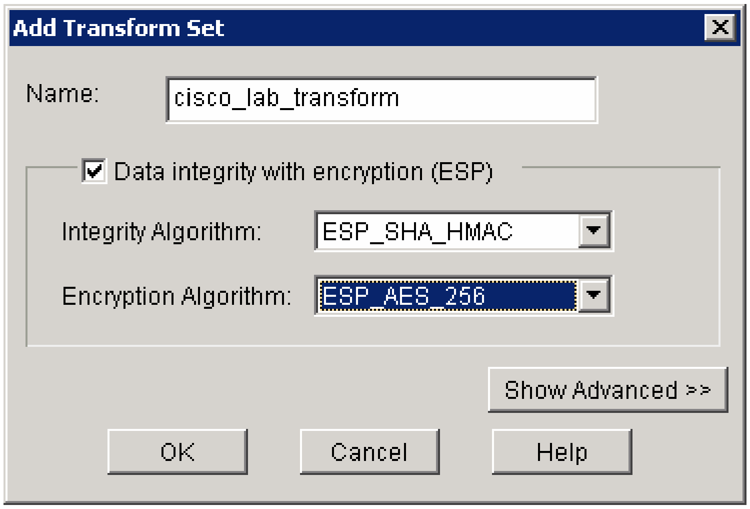
	Name	ESP Encryption	ESP Integrity	AH Integrity
	ESP-3DES-SHA	ESP_3DES	ESP_SHA_HMAC	

Figure 4-7: IPsec Transform Set List

Though the wizard does not explicitly state it, the transform set is the IPsec policy used to encrypt, hash, and authenticate packets that pass through the tunnel. The transform set is the IKE policy.

What is the function of the IPsec transform set?

Use the transform set settings shown in the following dialog box. If your IOS image doesn’t support those settings, configure the VPN settings as closely as possible. Ensure that you match the IPsec policies between the two VPN endpoints.



Add Transform Set

Name: cisco_lab_transform

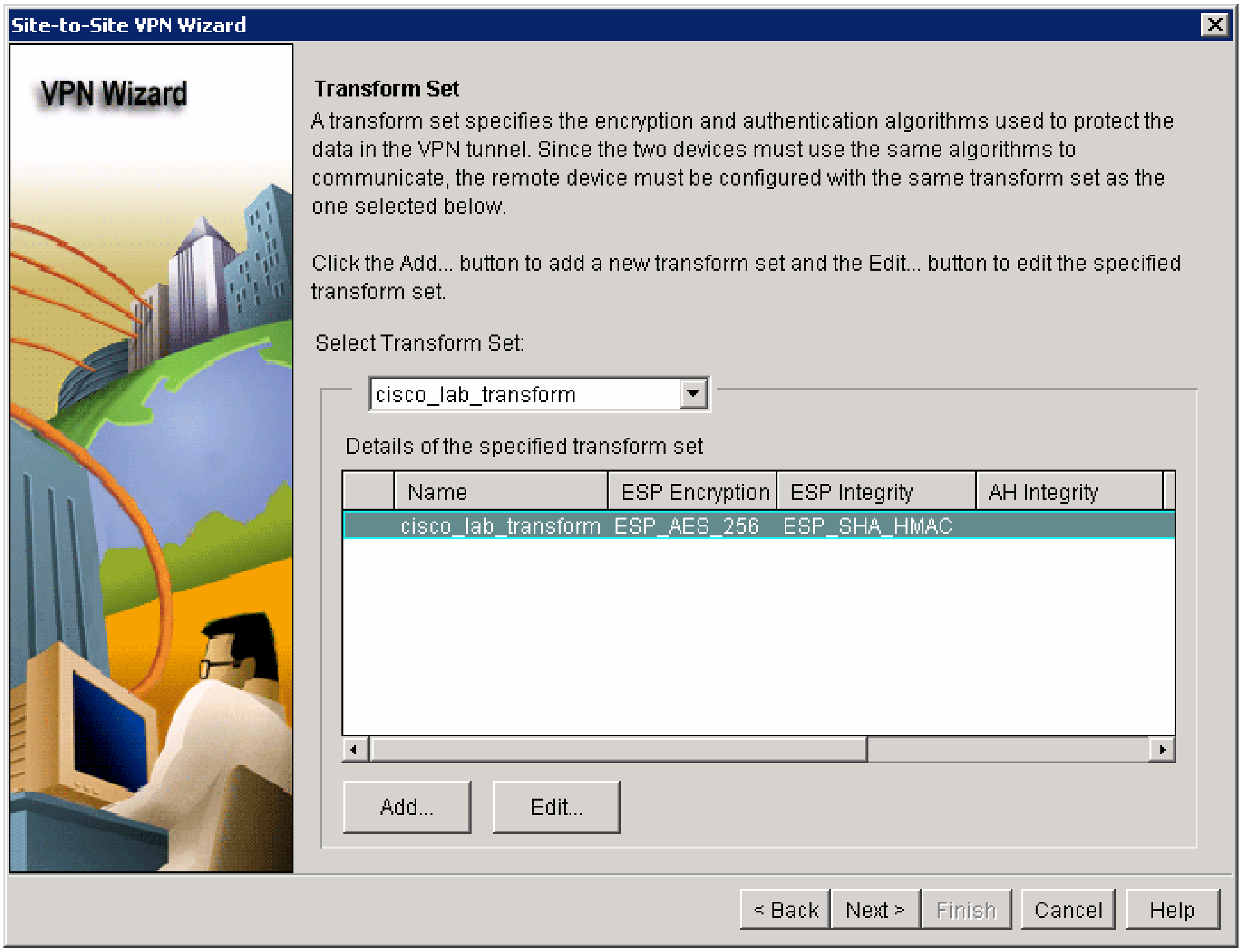
☑ Data integrity with encryption (ESP)

Integrity Algorithm: ESP_SHA_HMAC

Encryption Algorithm: ESP_AES_256

Figure 4-8: Add IPsec Transform Set Dialog

In the drop-down box, choose the transport set you just created. Click **Next** to continue.



Site-to-Site VPN Wizard

Transform Set

A transform set specifies the encryption and authentication algorithms used to protect the data in the VPN tunnel. Since the two devices must use the same algorithms to communicate, the remote device must be configured with the same transform set as the one selected below.

Click the Add... button to add a new transform set and the Edit... button to edit the specified transform set.

Select Transform Set: cisco_lab_transform

Details of the specified transform set

	Name	ESP Encryption	ESP Integrity	AH Integrity
	cisco_lab_transform	ESP_AES_256	ESP_SHA_HMAC	

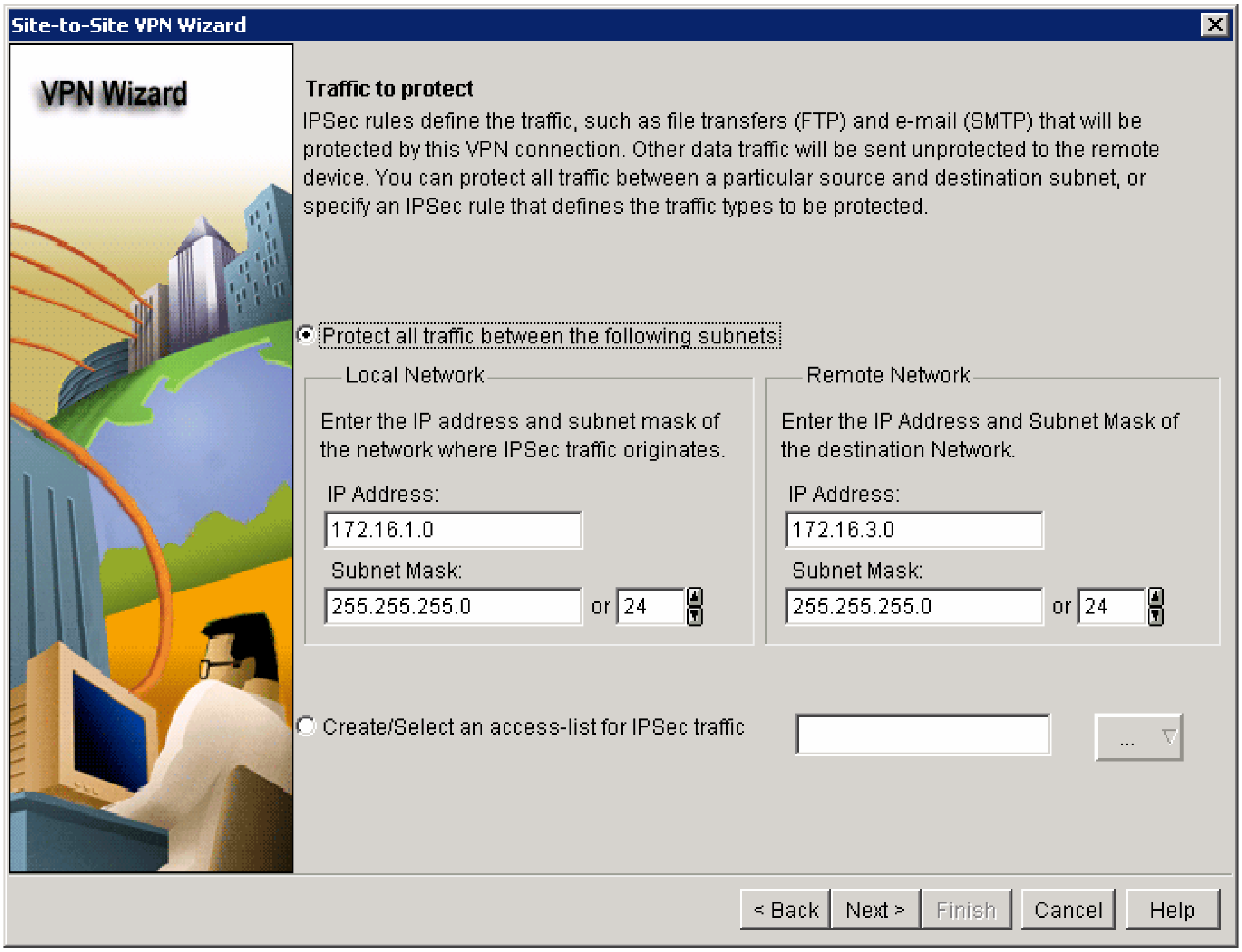
Figure 4-9: IPsec Transform Set List with Changes Applies

Finally you must define interesting traffic to be protected through the VPN tunnel. Interesting traffic will be defined through an access list when applied to the router. However, SDM allows users unfamiliar with access lists to define simple access lists based only on source and destination subnets.

If you enter source and destination subnets, such as this configuration will have, SDM will generate the access lists for you. If not, you can use an existing access list to mark which traffic to encrypt. In this example, the source and destination subnets are the loopback networks on R1 and R3, respectively.

Ensure that on R1 you define 172.16.1.0/24 as the source subnet and 172.16.3.0/24 as the destination subnet. Use the reverse for R3.

Click **Next** once you configure networks and masks.



Traffic to protect

IPSec rules define the traffic, such as file transfers (FTP) and e-mail (SMTP) that will be protected by this VPN connection. Other data traffic will be sent unprotected to the remote device. You can protect all traffic between a particular source and destination subnet, or specify an IPSec rule that defines the traffic types to be protected.

● Protect all traffic between the following subnets

Local Network

Enter the IP address and subnet mask of the network where IPSec traffic originates.

IP Address: 172.16.1.0

Subnet Mask: 255.255.255.0 or 24

Remote Network

Enter the IP Address and Subnet Mask of the destination Network.

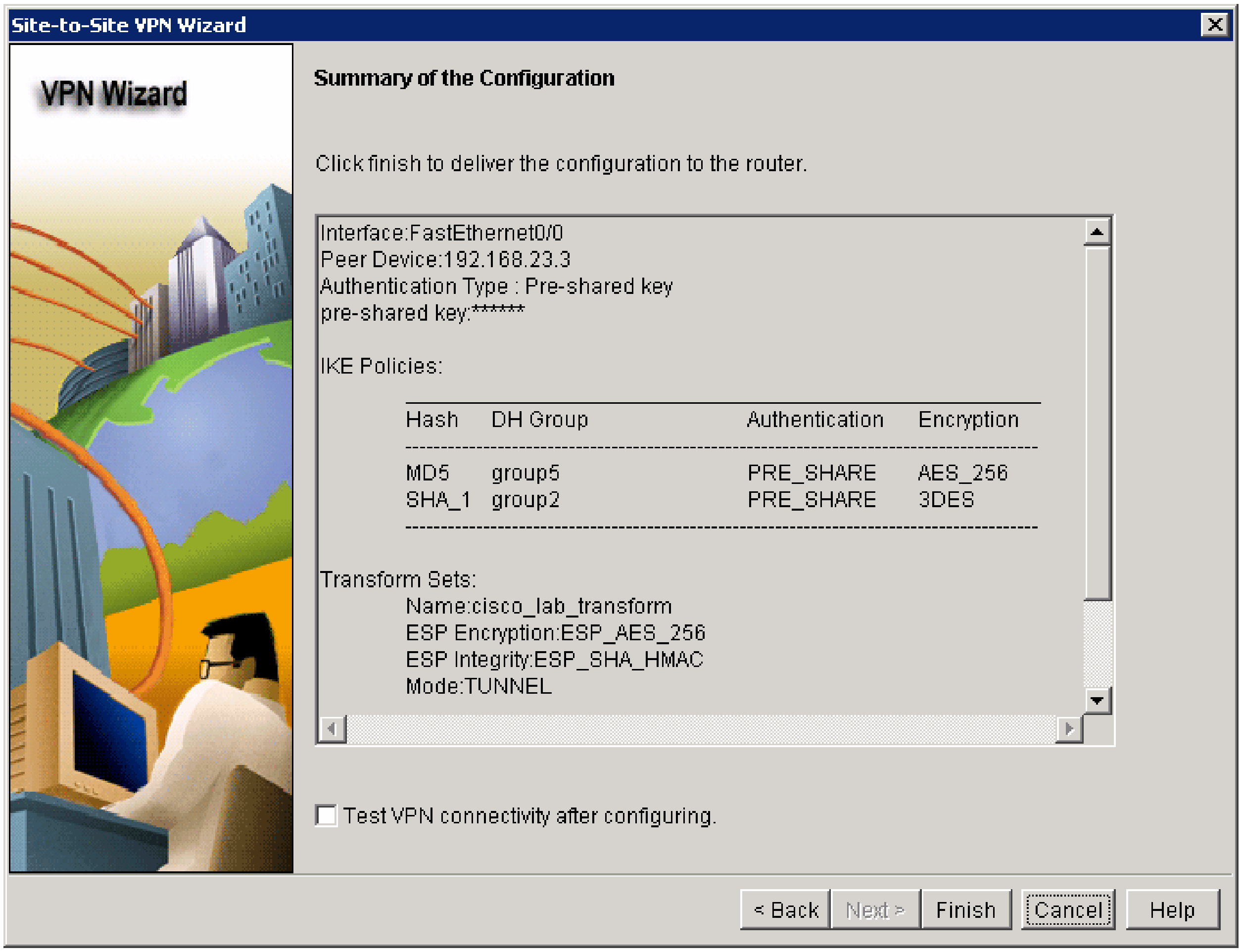
IP Address: 172.16.3.0

Subnet Mask: 255.255.255.0 or 24

○ Create/Select an access-list for IPSec traffic

Figure 4-10: Access List Definition

SDM presents a final summary of the changes it is going to make to the router. Do not check **Test VPN connectivity after configuring** because the VPN test will fail because you have not configured R3. Click **Finish**. SDM now modifies the R1's configuration based on the parameters you provided in this wizard.



Summary of the Configuration

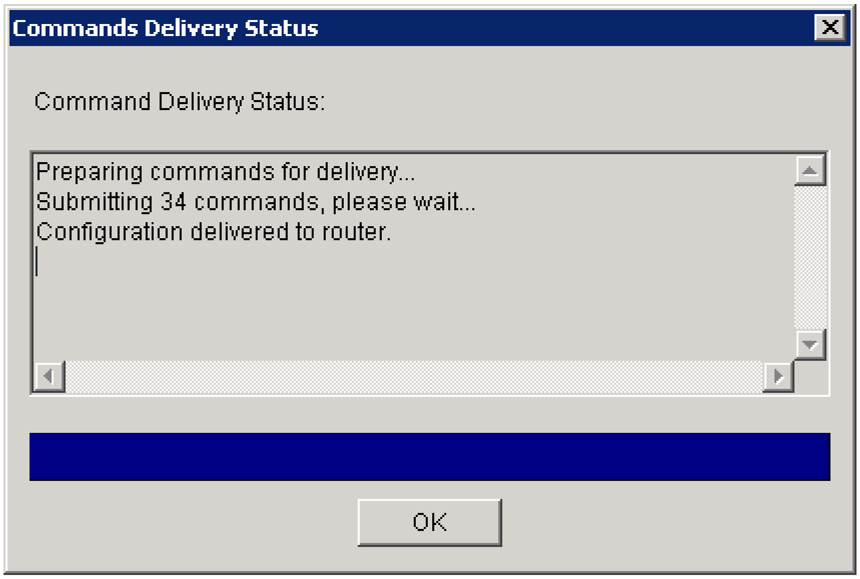
Click finish to deliver the configuration to the router.

```
Interface:FastEthernet0/0
Peer Device:192.168.23.3
Authentication Type : Pre-shared key
pre-shared key:******

IKE Policies:

          Hash     DH Group               Authentication    Encryption
          ------------------------------------------------------------
          MD5      group5                 PRE_SHARE         AES_256
          SHA_1    group2                 PRE_SHARE         3DES
          ------------------------------------------------------------

Transform Sets:
          Name:cisco_lab_transform
          ESP Encryption:ESP_AES_256
          ESP Integrity:ESP_SHA_HMAC
          Mode:TUNNEL
```

□ Test VPN connectivity after configuring.

Figure 4-11: Site-to-Site VPN Configuration Summary

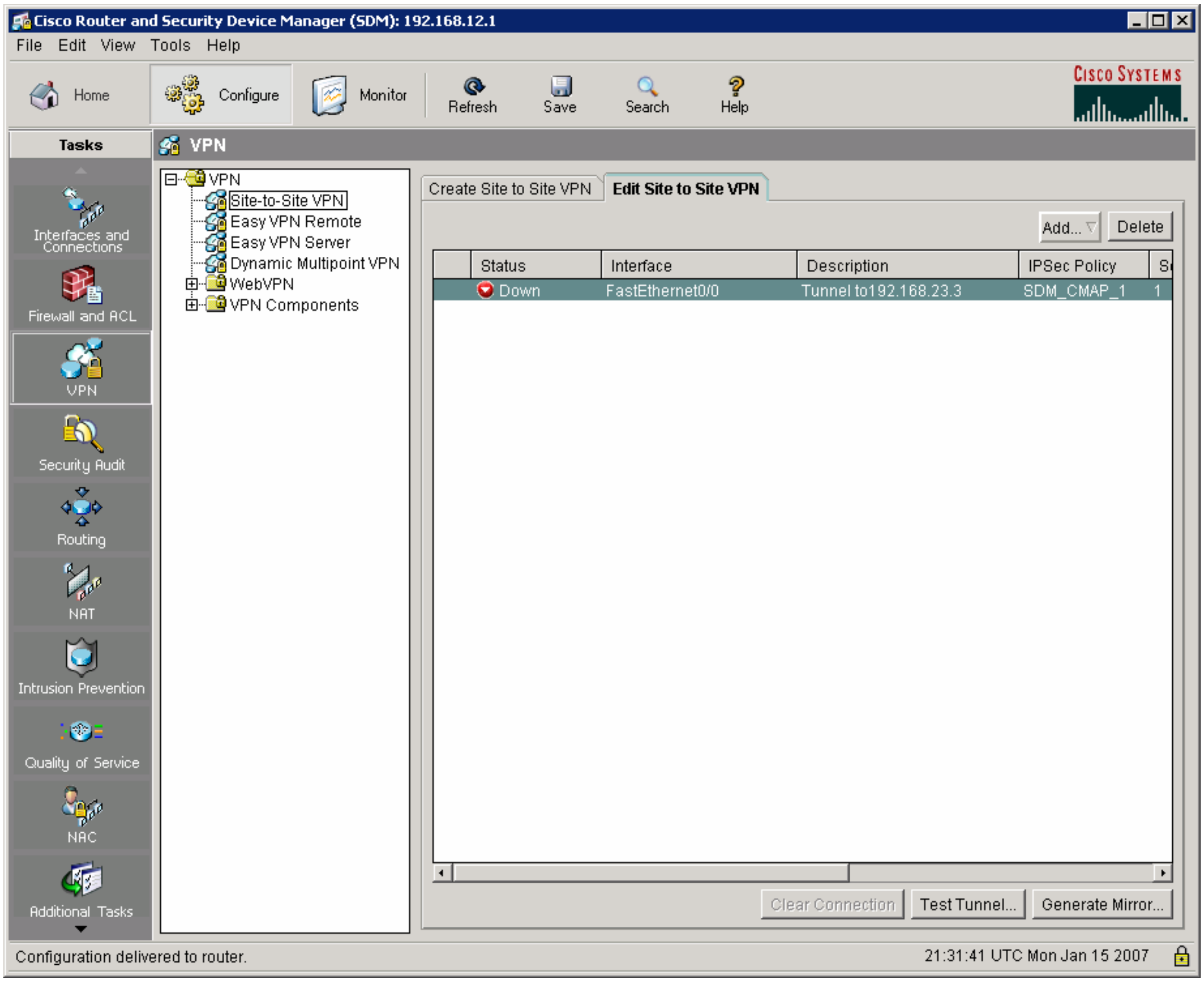
Once SDM has delivered the configuration to the router, click **OK**. The Site-to-Site VPN wizard closes, and you re-enter the VPN configuration window.



Preparing commands for delivery...
Submitting 34 commands, please wait...
Configuration delivered to router.

Figure 4-12: Command Delivery Progress Indicator

Step 5: Generate a Mirror Configuration for R3



Status	Interface	Description	IPSec Policy	S
Down	FastEthernet0/0	Tunnel to192.168.23.3	SDM_CMAP_1	1

Configuration delivered to router.

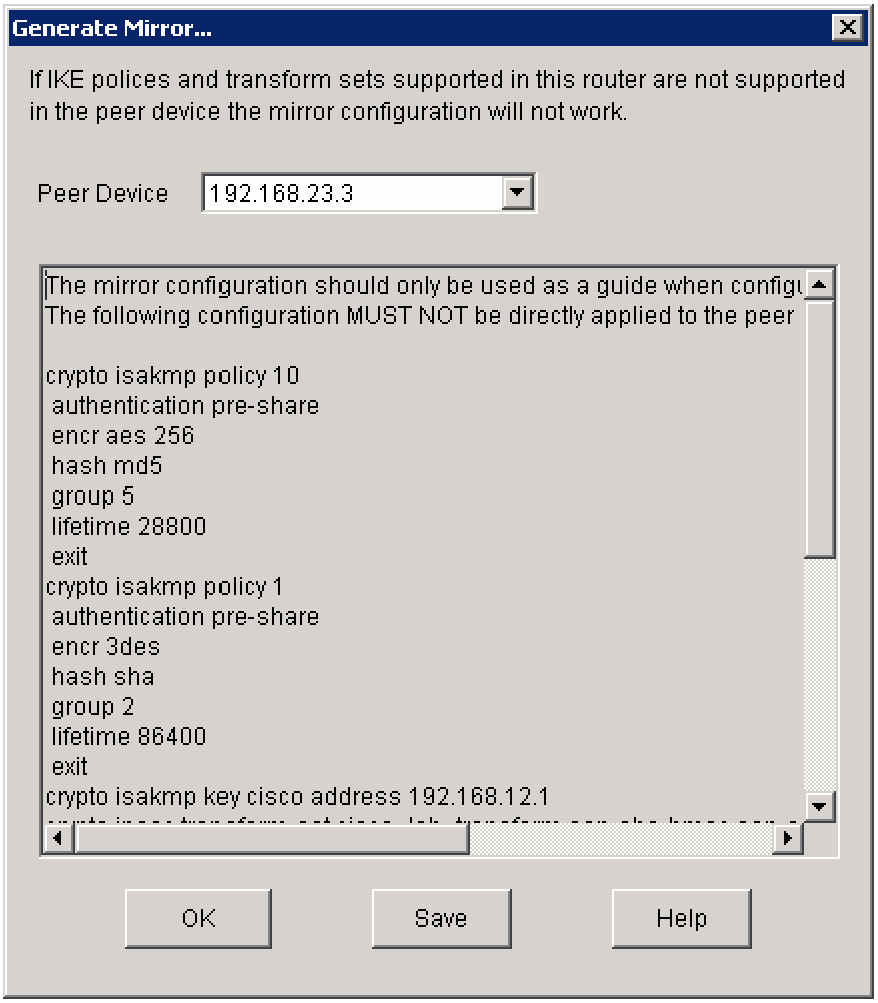
21:31:41 UTC Mon Jan 15 2007

Figure 5-1: VPN Configuration Screen

Navigate to the **Edit Site-to-Site VPN** tab.

Why is the status of the VPN that you just created “Down”?

Select the VPN policy you just configured and click the **Generate Mirror...** button in the lower right corner of the window.



If IKE polices and transform sets supported in this router are not supported in the peer device the mirror configuration will not work.

Peer Device 192.168.23.3

```
The mirror configuration should only be used as a guide when configu
The following configuration MUST NOT be directly applied to the peer

crypto isakmp policy 10
 authentication pre-share
 encr aes 256
 hash md5
 group 5
 lifetime 28800
 exit
crypto isakmp policy 1
 authentication pre-share
 encr 3des
 hash sha
 group 2
 lifetime 86400
 exit
crypto isakmp key cisco address 192.168.12.1
```

Figure 5-2: Mirror VPN Configuration

Enter global configuration mode on R3 by issuing the **configure terminal** command. Copy the commands in the SDM window and paste them into your configuration session with R3. You can also copy them by hand, but this method may be prone to error.

```
R3# configure terminal
R3(config)# crypto isakmp policy 10
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# encr aes 256
R3(config-isakmp)# hash md5
R3(config-isakmp)# group 5
R3(config-isakmp)# lifetime 28800
R3(config-isakmp)# exit
R3(config)# crypto isakmp policy 1
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# encr 3des
R3(config-isakmp)# hash sha
R3(config-isakmp)# group 2
R3(config-isakmp)# lifetime 86400
R3(config-isakmp)# exit
R3(config)# crypto isakmp key cisco address 192.168.12.1
```

```
R3(config)# crypto IPsec transform-set cisco_lab_transform esp-sha-hmac esp-
aes 256
R3(cfg-crypto-trans)# mode tunnel
R3(cfg-crypto-trans)# exit
R3(config)# ip access list extended SDM_1
R3(config-ext-nacl)# remark SDM_ACL Category=4
R3(config-ext-nacl)# remark IPsec Rule
R3(config-ext-nacl)# permit ip 172.16.3.0 0.0.0.255 172.16.1.0 0.0.0.255
R3(config-ext-nacl)# exit
R3(config)# crypto map SDM_CMAP_1 1 IPsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R3(config-crypto-map)# description Apply the crypto map on the peer router's
interface having IP address 192.168.23.3 that connects to this router.
R3(config-crypto-map)# set transform-set cisco_lab_transform
R3(config-crypto-map)# set peer 192.168.12.1
R3(config-crypto-map)# match address SDM_1
R3(config-crypto-map)# set security-association lifetime seconds 3600
R3(config-crypto-map)# set security-association lifetime kilobytes 4608000
R3(config-crypto-map)# exit
```

You may have noticed the warning in the **Generate Mirror…** window which stated that the configuration generated should only be used as a guide for setting up a site-to-site VPN. Although these configuration commands will apply most of the necessary commands to the remote router, they will not apply that configuration to any router interface. Without an associated interface, none of the cryptography settings that you just pasted into R3 are activated. Additionally, if this overwrote some existing IPsec settings, you could potentially destroy one or more existing VPN tunnels.

In this situation, both of your endpoints should not have any VPNs configured before you run the site-to-site VPN wizard or the generated commands for the remote endpoint.

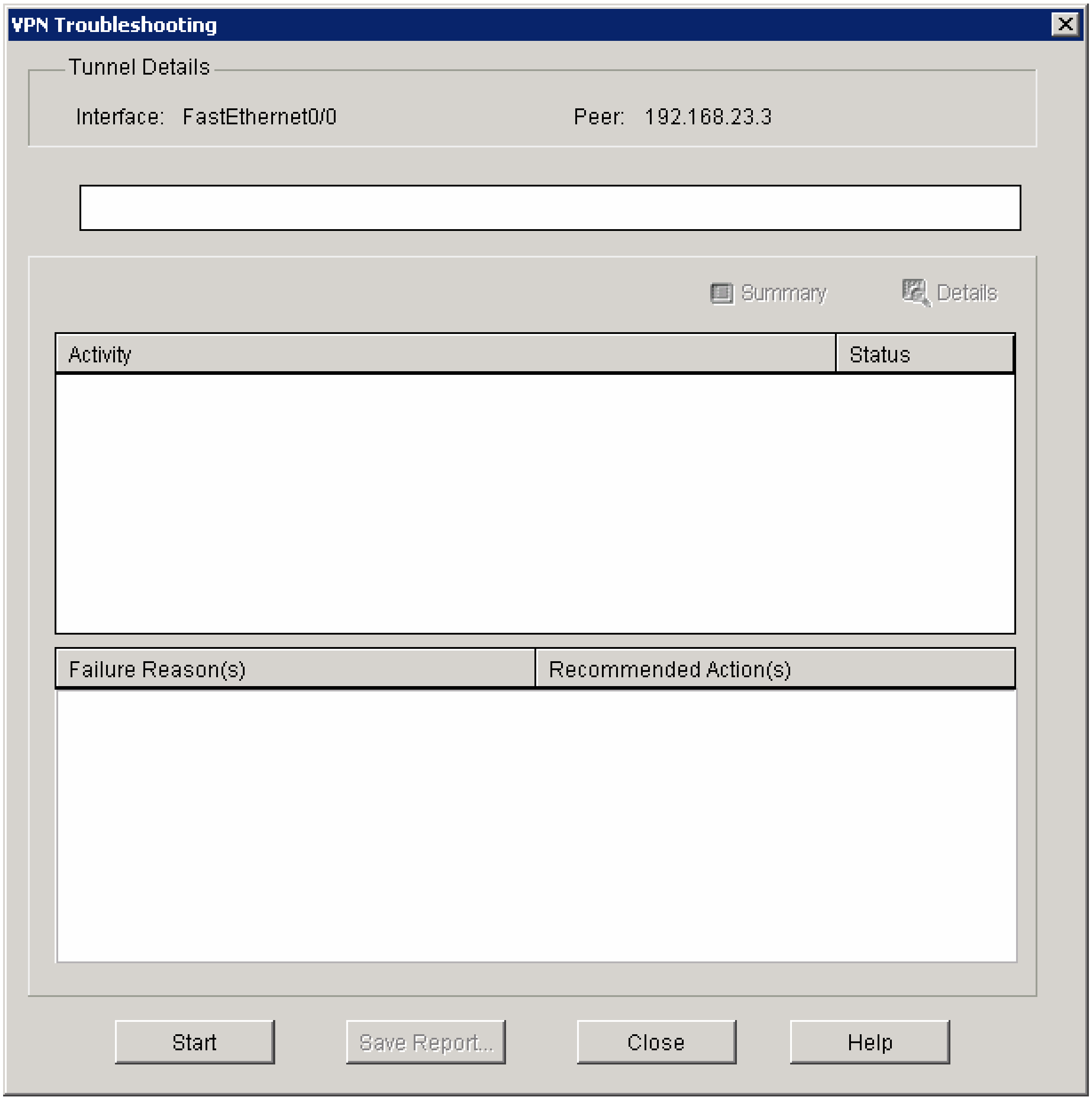
As previously noted, you now need to apply IPsec configuration to an interface. In the generated configuration, “SDM_CMAP_1” is the name of the crypto map that was created. Apply this crypto map to the serial interface facing R2 using the **crypto map** *name* command in interface configuration mode. This will generate a warning that the Internet Security Association and Key Management Protocol (ISAKMP) is now activated.

```
R3(config)# interface serial 0/0/1
R3(config-if)# crypto map SDM_CMAP_1
*Jan 15 22:00:38.184: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

Step 6: Verify the VPN Configuration using SDM

Now that you have configured R3 for a VPN, use SDM to test the configuration. On the **Edit Site to Site VPN** tab shown in Figure 5-1, choose the VPN you just created and click **Test Tunnel...**.

Click **Start** to have SDM start troubleshooting the tunnel.


VPN Troubleshooting

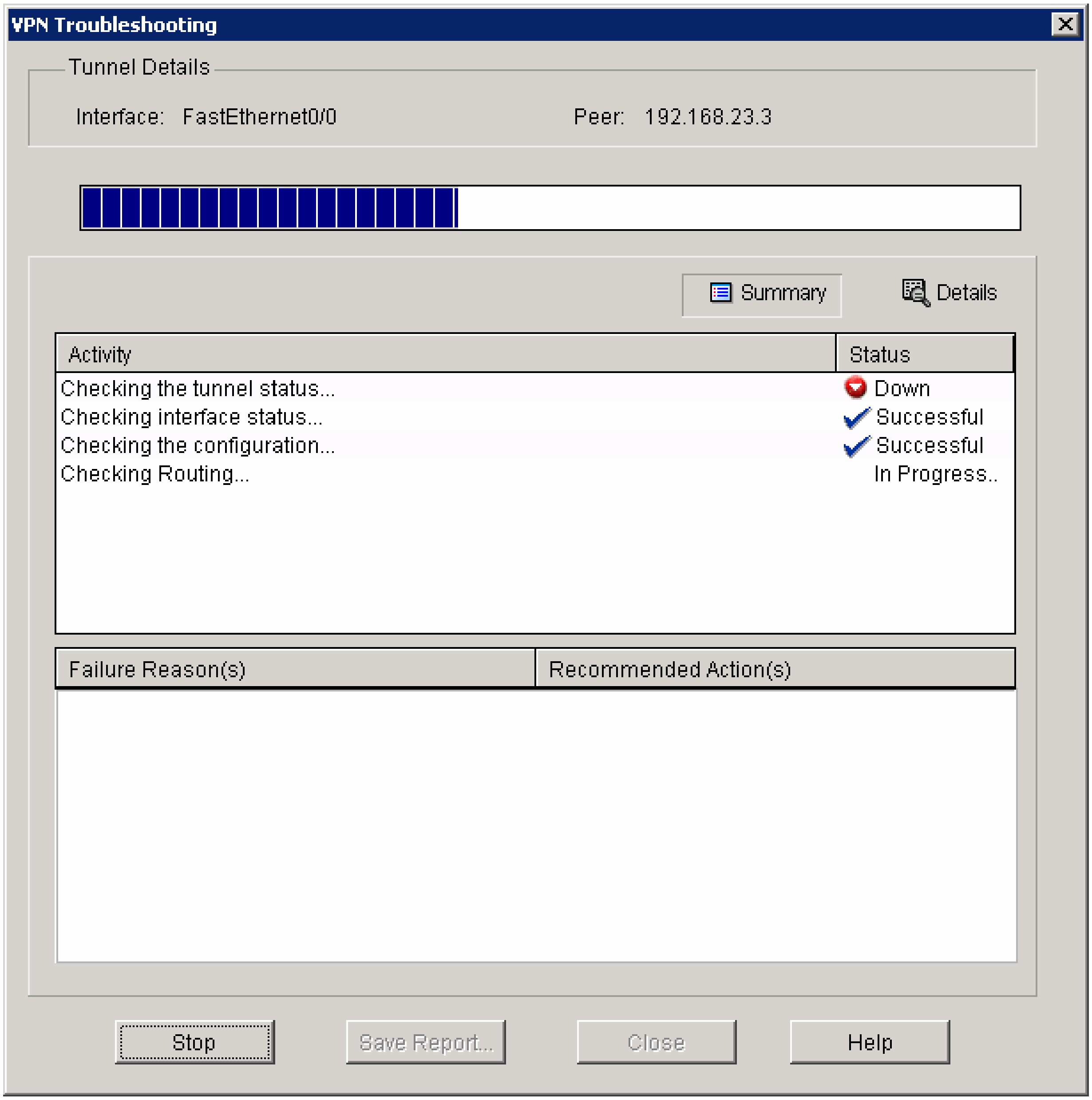
Tunnel Details

Interface: FastEthernet0/0

Peer: 192.168.23.3

Figure 6-1: VPN Testing Window

This process may take a few moments.



Tunnel Details

Interface: FastEthernet0/0

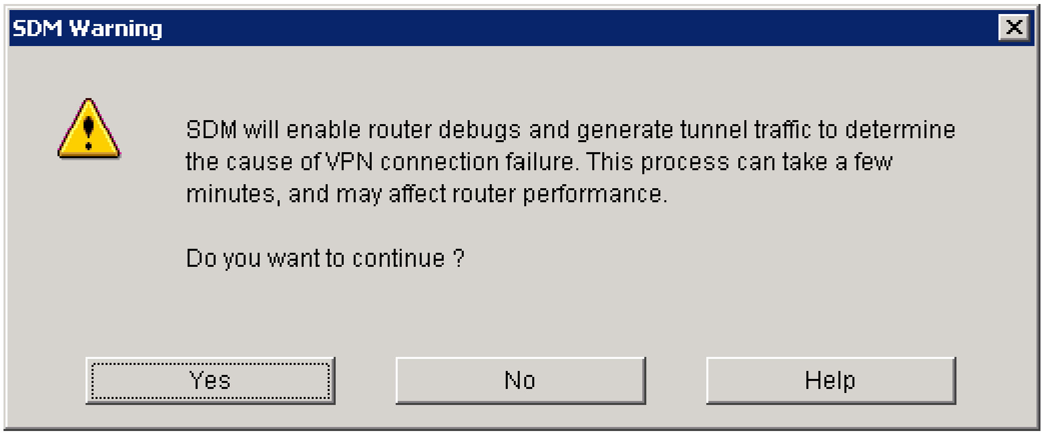
Peer: 192.168.23.3

Activity	Status
Checking the tunnel status...	Down
Checking interface status...	✓ Successful
Checking the configuration...	✓ Successful
Checking Routing...	In Progress..

Failure Reason(s)	Recommended Action(s)

Figure 6-2: VPN Test In Progress

If SDM encounters any errors, it will offer to troubleshoot the problem for you. Click **Yes** to continue.

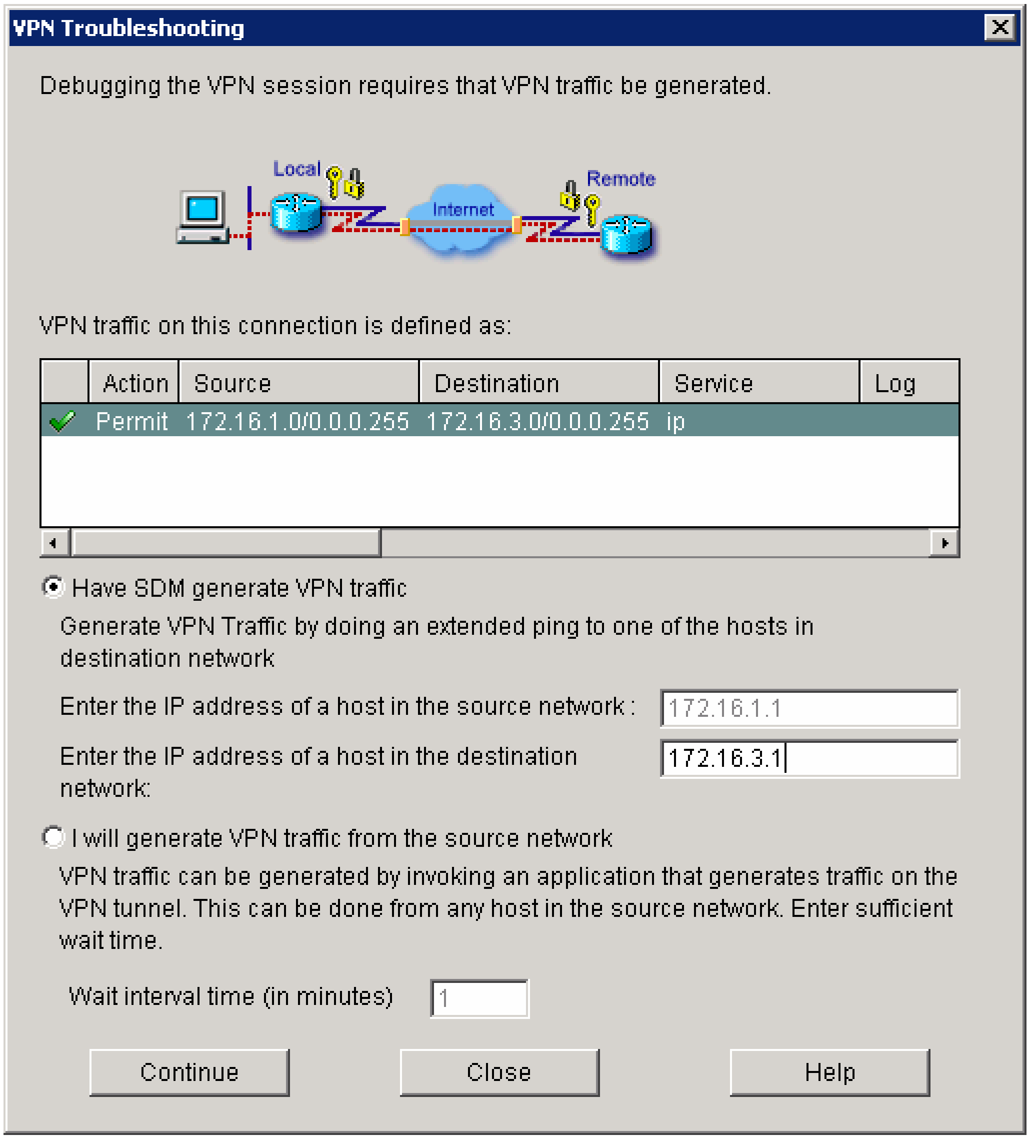

SDM Warning

SDM will enable router debugs and generate tunnel traffic to determine the cause of VPN connection failure. This process can take a few minutes, and may affect router performance.

Do you want to continue ?

Figure 6-3: SDM Performance Warning

Choose the **Have SDM generate VPN traffic** option. Enter R3’s loopback address as the destination address. Click **Continue**.



VPN Troubleshooting

Debugging the VPN session requires that VPN traffic be generated.

VPN traffic on this connection is defined as:

	Action	Source	Destination	Service	Log
✓	Permit	172.16.1.0/0.0.0.255	172.16.3.0/0.0.0.255	ip	

● Have SDM generate VPN traffic

Generate VPN Traffic by doing an extended ping to one of the hosts in destination network

Enter the IP address of a host in the source network : 172.16.1.1

Enter the IP address of a host in the destination network: 172.16.3.1

○ I will generate VPN traffic from the source network

VPN traffic can be generated by invoking an application that generates traffic on the VPN tunnel. This can be done from any host in the source network. Enter sufficient wait time.

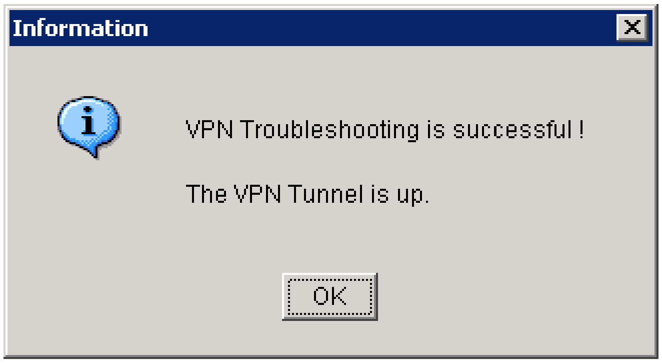
Wait interval time (in minutes) 1

Figure 6-4: Test Traffic Generation Window

Allow SDM to analyze the situation and continue running the test.

When it has completed the test, you should get a message box acknowledging that the VPN tunnel is up. Click **OK**.

If you do not receive a successful reply from the test, use SDM's suggestions to troubleshoot.



Information

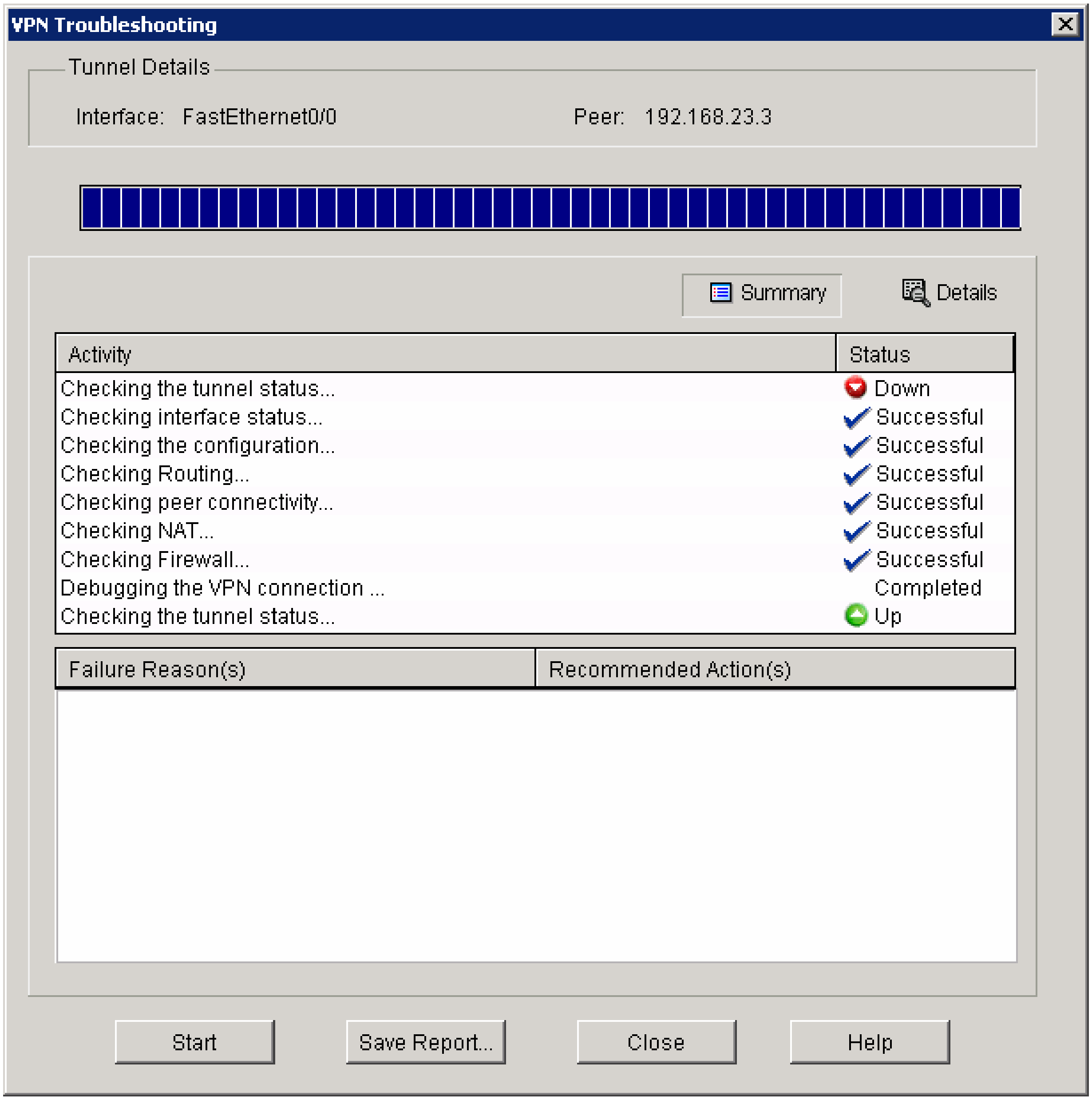
VPN Troubleshooting is successful !

The VPN Tunnel is up.

Figure 6-5: Successful VPN Test Status Window

The status displayed in the following window should be “Up,” indicating that the VPN connection is now active.

Click **Close** in the VPN Test window to go back to the main SDM console.



Interface: FastEthernet0/0

Peer: 192.168.23.3

Activity	Status
Checking the tunnel status...	Down
Checking interface status...	✓ Successful
Checking the configuration...	✓ Successful
Checking Routing...	✓ Successful
Checking peer connectivity...	✓ Successful
Checking NAT...	✓ Successful
Checking Firewall...	✓ Successful
Debugging the VPN connection ...	Completed
Checking the tunnel status...	Up

Failure Reason(s)	Recommended Action(s)

Figure 6-6: Detailed VPN Test Results

Step 7: Verify the VPN configuration using the IOS CLI

While it is beneficial to have SDM to help troubleshoot a VPN, this is not always possible. There will be times at which you only have console or telnet access to a router. Fortunately, the Cisco IOS has an extensive array of **show** and **debug** commands for analyzing cryptographic configurations.

A useful command for monitoring IPsec VPNs is the **show crypto IPsec sa** command. This command lists all current IPsec security associations and their parameters. Issue this command on R1 and R3.

```
R1# show crypto IPsec sa

interface: FastEthernet0/0
    Crypto map tag: SDM_CMAP_1, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 29, #pkts encrypt: 29, #pkts digest: 29
    #pkts decaps: 29, #pkts decrypt: 29, #pkts verify: 29
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 192.168.12.1, remote crypto endpt.: 192.168.23.3
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
     current outbound spi: 0x487708CA(1215760586)

     inbound esp sas:
      spi: 0xD182B74A(3515004746)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: NETGX:1, crypto map: SDM_CMAP_1
        sa timing: remaining key lifetime (k/sec): (4420862/2990)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x487708CA(1215760586)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: NETGX:2, crypto map: SDM_CMAP_1
        sa timing: remaining key lifetime (k/sec): (4420862/2989)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:
```

```
R3# show crypto IPsec sa

interface: Serial0/0/1
    Crypto map tag: SDM_CMAP_1, local addr 192.168.23.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   current_peer 192.168.12.1 port 500
```

```
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 29, #pkts encrypt: 29, #pkts digest: 29
    #pkts decaps: 29, #pkts decrypt: 29, #pkts verify: 29
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 192.168.23.3, remote crypto endpt.: 192.168.12.1
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/1
     current outbound spi: 0xD182B74A(3515004746)

     inbound esp sas:
      spi: 0x487708CA(1215760586)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3001, flow_id: NETGX:1, crypto map: SDM_CMAP_1
        sa timing: remaining key lifetime (k/sec): (4467883/2964)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xD182B74A(3515004746)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3002, flow_id: NETGX:2, crypto map: SDM_CMAP_1
        sa timing: remaining key lifetime (k/sec): (4467883/2962)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:
```

View the numbers of packets being encrypted and decrypted on each end. You can verify that the correct packets are being encrypted and decrypted by checking that these packet counts increment when traffic is sent. From R1 ping R3's loopback. Then look at the number of encrypted and decrypted packets on each side.

```
R1# ping 172.16.3.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

R1# show crypto IPsec sa

interface: FastEthernet0/0
    Crypto map tag: SDM_CMAP_1, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 29, #pkts encrypt: 29, #pkts digest: 29
    #pkts decaps: 29, #pkts decrypt: 29, #pkts verify: 29
<OUTPUT OMITTED>

R3# show crypto IPsec sa

interface: Serial0/0/1
    Crypto map tag: SDM_CMAP_1, local addr 192.168.23.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   current_peer 192.168.12.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 29, #pkts encrypt: 29, #pkts digest: 29
    #pkts decaps: 29, #pkts decrypt: 29, #pkts verify: 29
<OUTPUT OMITTED>
```

Why is the packet count unchanged?

Based on the configuration you enabled on the VPN tunnel, how could you create interesting traffic that would pass through the encrypted tunnel?

Use an extended **ping** to source packets from R1’s loopback interface toward 172.16.3.0/24.

Will these packets be encrypted by the VPN?

Test your answer.

```
R1# ping
Protocol [ip]:
Target IP address: 172.16.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: Loopback0
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 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/48/48 ms

R1# show crypto IPsec sa

interface: FastEthernet0/0
    Crypto map tag: SDM_CMAP_1, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 34, #pkts encrypt: 34, #pkts digest: 34
    #pkts decaps: 34, #pkts decrypt: 34, #pkts verify: 34
<OUTPUT OMITTED>

R3# show crypto IPsec sa

interface: Serial0/0/1
    Crypto map tag: SDM_CMAP_1, local addr 192.168.23.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   current_peer 192.168.12.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 34, #pkts encrypt: 34, #pkts digest: 34
    #pkts decaps: 34, #pkts decrypt: 34, #pkts verify: 34
<OUTPUT OMITTED>
```

Another useful command is **show crypto isakmp sa**, which shows ISAKMP security associations.

```
R1# show crypto isakmp sa
dst             src             state          conn-id slot status
192.168.23.3    192.168.12.1    QM_IDLE              1    0 ACTIVE

R3# show crypto isakmp sa
dst             src             state          conn-id slot status
192.168.23.3    192.168.12.1    QM_IDLE              1    0 ACTIVE
```

Remember that there are two types of security associations necessary to bring this VPN tunnel up. The ISAKMP security association is initiated by IKE Phase 1, and allows the routers to securely exchange IPsec policies. The second type of security association is initiated during IKE Phase 2, and allows the routers to securely send the data traffic.

These are just a few **show** commands. There are many other useful **show** and **debug** crypto commands.

Challenge: Use Wireshark Capture Encryption of Traffic

You can observe packets on the wire using Wireshark and see how their content looks unencrypted and then encrypted. To do this, first configure a SPAN session on the switch and open up Wireshark on a host attached to the SPAN destination port. You can use the host that you used for SDM because you don't need it anymore to configure the VPNs. If you do not know how to do this, refer to Lab 3.3: Configuring Wireshark and SPAN.

Next, you will remove the **crypto map** statements on R1 and R3. View the current configuration on the FastEthernet0/0 interface on R1 and Serial0/0/1 as shown below.

Then, issue the **no crypto map** *name* command in interface configuration mode to remove the ISAKMP security association. The router may issue a warning that ISAKMP is now off.

```
R1:

R1# show run | interface fastethernet0/0
Building configuration...

Current configuration : 120 bytes
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 duplex auto
 speed auto
 crypto map SDM_CMAP_1
end

R1# configure terminal
R1(config)# interface fastethernet0/0
R1(config-if)# no crypto map SDM_CMAP_1
*Jan 16 06:02:58.999: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF

R3:

R3# show run interface | serial0/0/1
Building configuration...

Current configuration : 91 bytes
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map SDM_CMAP_1
end

R3# configure terminal
R3(config)# interface serial0/0/1
R3(config-if)# no crypto map SDM_CMAP_1
*Jan 16 06:05:36.038: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
```

You will attempt to sniff telnet traffic from R1 to R3. Enable telnet access on R3 and configure a secure password to get to configuration mode on R3.

```
R3(config)# enable secret cisco
```

```
R3(config)# line vty 0 4
R3(config-line)# password cisco
R3(config-line)# login
```

The routers have now been configured to allow telnet access.

Have Wireshark start sniffing the packets that it receives via the SPAN session.

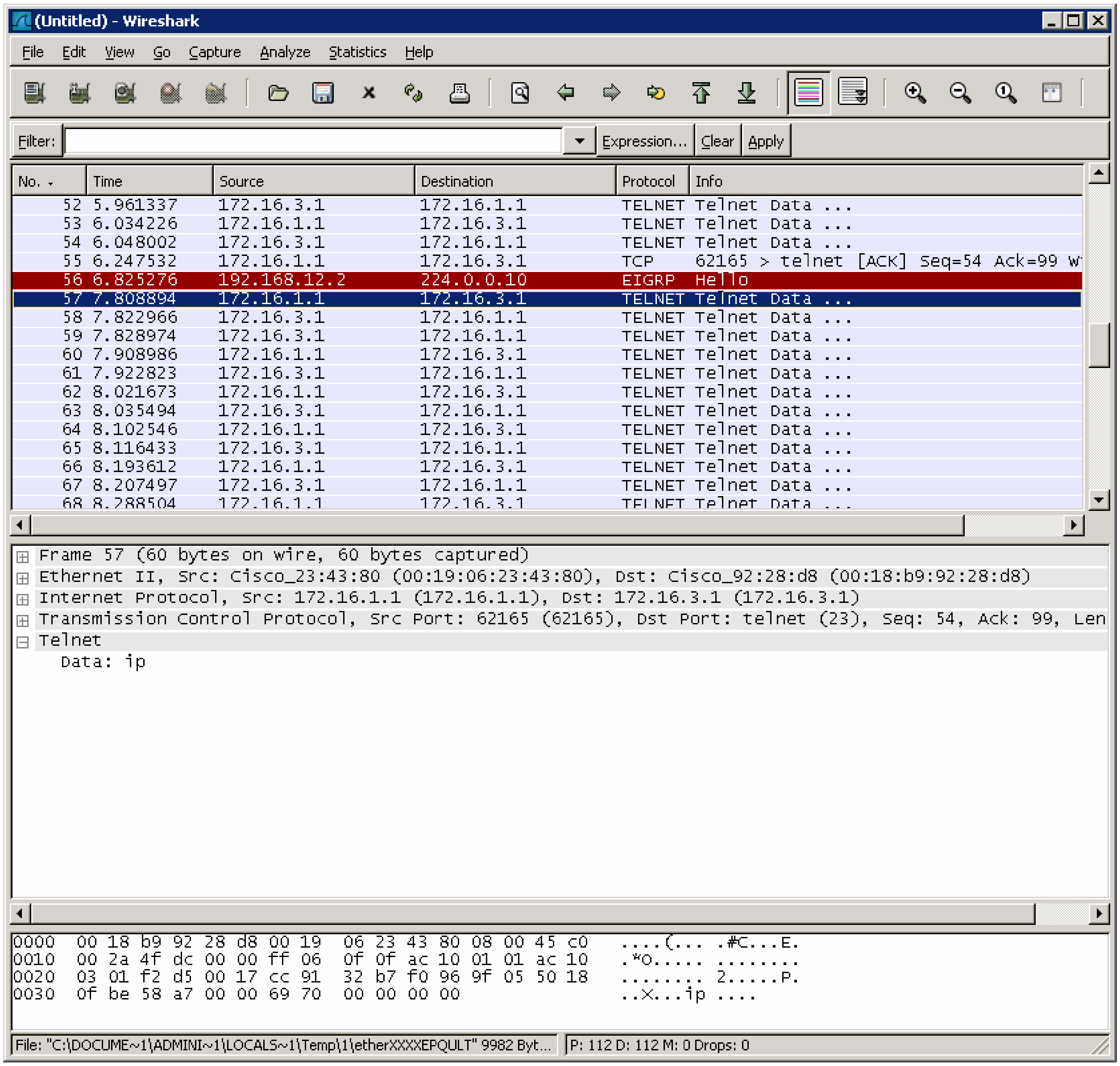
Choose **Capture > Interfaces...**. Then click the **Start** button associated with the interface connected to the SPAN destination port. SPAN should start capturing packets on the line, so you can now telnet from R1’s loopback to R3’s loopback. To source telnet traffic, use the **telnet** *destination* **/source** *interface* command.

As shown in the previous step, you must source the telnet session from R1’s loopback interface to simulate the interesting traffic that will match the VPN’s access list.

First, begin capturing using Wireshark. Then, begin the telnet session. Once you are connected to R3, try issuing a command or two and then logging out.

```
R1# telnet 172.16.3.1 /source Loopback0
Trying 172.16.3.1 ... Open


User Access Verification

Password: [cisco]
R3> en
Password: [cisco]

R3# show ip interface brief
Interface            IP-Address      OK? Method Status                Protocol
FastEthernet0/0      unassigned      YES unset  administratively down down
FastEthernet0/1      unassigned      YES unset  administratively down down
Serial0/0/0          unassigned      YES unset  administratively down down
Serial0/0/1          192.168.23.3    YES manual up                    up
Serial0/1/0          unassigned      YES unset  administratively down down
Serial0/1/1          unassigned      YES unset  administratively down down
Loopback0            172.16.3.1      YES manual up                    up

R3# exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

Now, end the capture and look at the output. You will see a set of telnet data packets. Some of these, especially the return packets, will show whole unencrypted streams of text. The reason some return packets having longer text strings is because return packets can be streamed consecutively from the router managing the connection, whereas the text you type into telnet gets sent in chunks of characters or even character by character, depending on your typing speed.



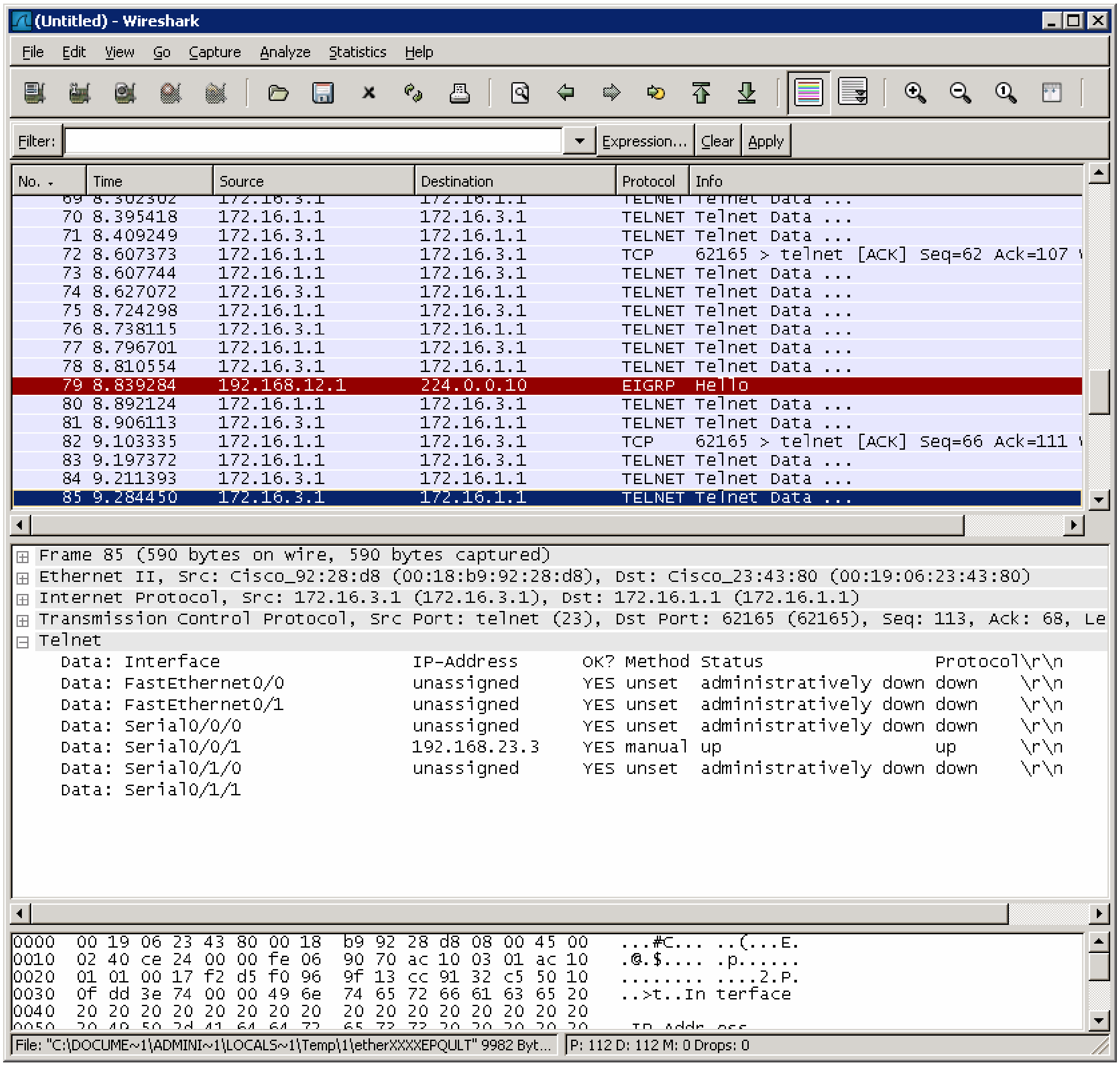
```
No. -  Time        Source         Destination    Protocol Info
    52 5.961337    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    53 6.034226    172.16.1.1     172.16.3.1     TELNET   Telnet Data ...
    54 6.048002    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    55 6.247532    172.16.1.1     172.16.3.1     TCP      62165 > telnet [ACK] Seq=54 Ack=99 W
    56 6.825276    192.168.12.2   224.0.0.10     EIGRP    Hello
    57 7.808894    172.16.1.1     172.16.3.1     TELNET   Telnet Data ...
    58 7.822966    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    59 7.828974    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    60 7.908986    172.16.1.1     172.16.3.1     TELNET   Telnet Data ...
    61 7.922823    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    62 8.021673    172.16.1.1     172.16.3.1     TELNET   Telnet Data ...
    63 8.035494    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    64 8.102546    172.16.1.1     172.16.3.1     TELNET   Telnet Data ...
    65 8.116433    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    66 8.193612    172.16.1.1     172.16.3.1     TELNET   Telnet Data ...
    67 8.207497    172.16.3.1     172.16.1.1     TELNET   Telnet Data ...
    68 8.288504    172.16.1.1     172.16.3.1     TELNET   Telnet Data ...
```

```
⊞ Frame 57 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: Cisco_23:43:80 (00:19:06:23:43:80), Dst: Cisco_92:28:d8 (00:18:b9:92:28:d8)
⊞ Internet Protocol, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.3.1 (172.16.3.1)
⊞ Transmission Control Protocol, Src Port: 62165 (62165), Dst Port: telnet (23), Seq: 54, Ack: 99, Len
⊟ Telnet
    Data: ip
```

```
0000  00 18 b9 92 28 d8 00 19  06 23 43 80 08 00 45 c0   ....(... .#C...E.
0010  00 2a 4f dc 00 00 ff 06  0f 0f ac 10 01 01 ac 10   .*O..... ........
0020  03 01 f2 d5 00 17 cc 91  32 b7 f0 96 9f 05 50 18   ........ 2.....P.
0030  0f be 58 a7 00 00 69 70  00 00 00 00               ..X...ip ....
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\etherXXXXEPQULT" 9982 Byt... P: 112 D: 112 M: 0 Drops: 0

Figure 8-1: Detailed Packet Data on Telnet String Sent From R1



```
No.  Time      Source         Destination    Protocol Info
 69 8.382302  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
 70 8.395418  172.16.1.1     172.16.3.1     TELNET Telnet Data ...
 71 8.409249  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
 72 8.607373  172.16.1.1     172.16.3.1     TCP    62165 > telnet [ACK] Seq=62 Ack=107 W
 73 8.607744  172.16.1.1     172.16.3.1     TELNET Telnet Data ...
 74 8.627072  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
 75 8.724298  172.16.1.1     172.16.3.1     TELNET Telnet Data ...
 76 8.738115  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
 77 8.796701  172.16.1.1     172.16.3.1     TELNET Telnet Data ...
 78 8.810554  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
 79 8.839284  192.168.12.1   224.0.0.10     EIGRP  Hello
 80 8.892124  172.16.1.1     172.16.3.1     TELNET Telnet Data ...
 81 8.906113  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
 82 9.103335  172.16.1.1     172.16.3.1     TCP    62165 > telnet [ACK] Seq=66 Ack=111 W
 83 9.197372  172.16.1.1     172.16.3.1     TELNET Telnet Data ...
 84 9.211393  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
 85 9.284450  172.16.3.1     172.16.1.1     TELNET Telnet Data ...
```

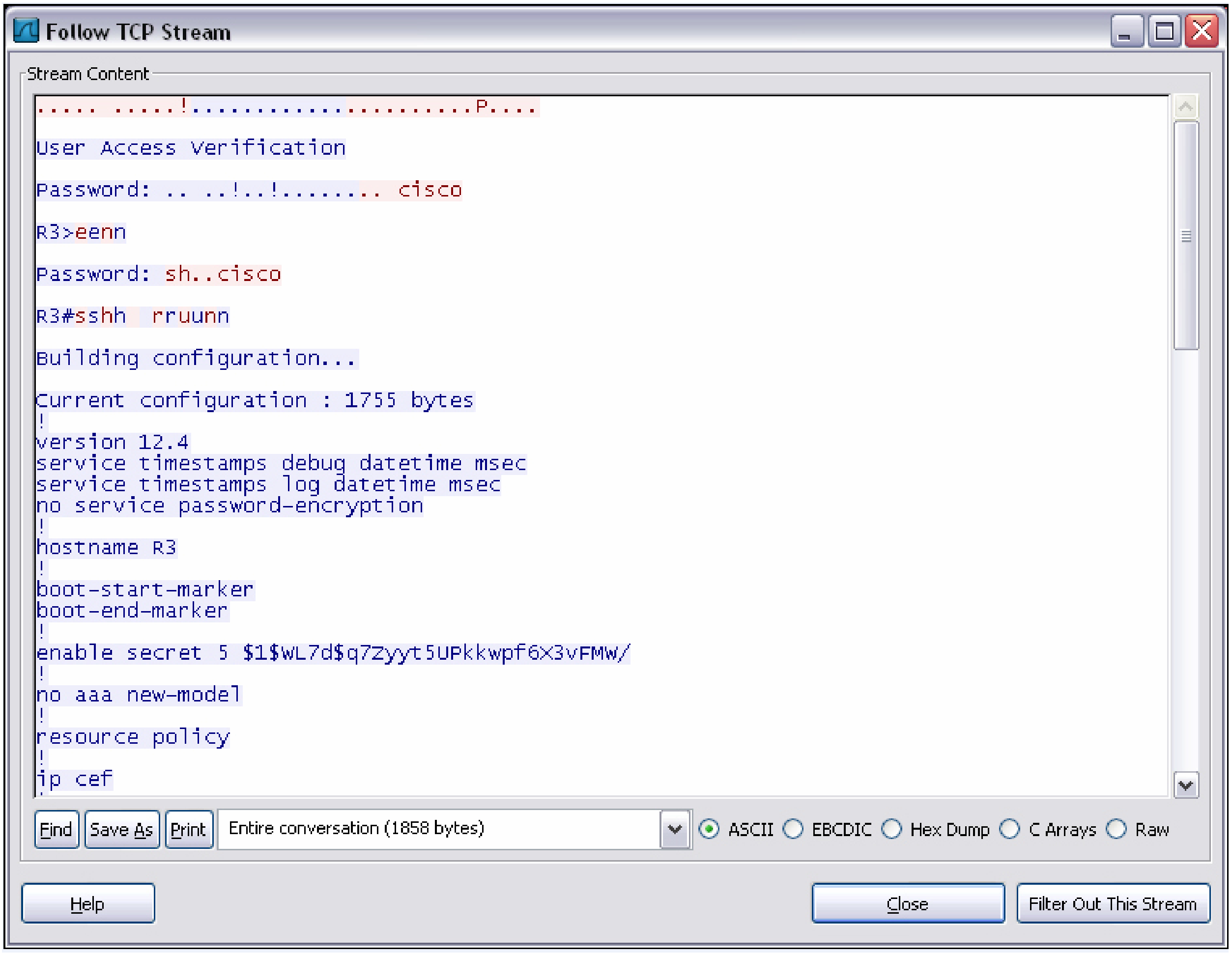
```
⊞ Frame 85 (590 bytes on wire, 590 bytes captured)
⊞ Ethernet II, Src: Cisco_92:28:d8 (00:18:b9:92:28:d8), Dst: Cisco_23:43:80 (00:19:06:23:43:80)
⊞ Internet Protocol, Src: 172.16.3.1 (172.16.3.1), Dst: 172.16.1.1 (172.16.1.1)
⊞ Transmission Control Protocol, Src Port: telnet (23), Dst Port: 62165 (62165), Seq: 113, Ack: 68, Le
⊟ Telnet
    Data: Interface                  IP-Address      OK? Method Status                Protocol\r\n
    Data: FastEthernet0/0            unassigned      YES unset  administratively down down    \r\n
    Data: FastEthernet0/1            unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/0/0                unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/0/1                192.168.23.3    YES manual up                    up      \r\n
    Data: Serial0/1/0                unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/1/1
```

```
0000   00 19 06 23 43 80 00 18  b9 92 28 d8 08 00 45 00   ...#C... ..(...E.
0010   02 40 ce 24 00 00 fe 06  90 70 ac 10 03 01 ac 10   .@.$.... .p......
0020   01 01 00 17 f2 d5 f0 96  9f 13 cc 91 32 c5 50 10   ........ ....2.P.
0030   0f dd 3e 74 00 00 49 6e  74 65 72 66 61 63 65 20   ..>t..In terface
0040   20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
0050   20 49 50 2d 41 64 64 72  65 73 73 20 20 20 20 20    IP-Addr ess
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\etherXXXXEPQULT" 9982 Byt... P: 112 D: 112 M: 0 Drops: 0

Figure 8-2: Detailed Packet Data on Return Telnet Traffic from R3

Highlight an entry that contains a telnet protocol. Right click on the entry and choose **Follow TCP Stream**.



```
..... .....!.....................P....

User Access Verification

Password: .. ..!..!........ cisco

R3>eenn

Password: sh..cisco

R3#sshh  rruunn

Building configuration...

Current configuration : 1755 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$WL7d$q7Zyyt5UPkkwpf6X3vFMW/
!
no aaa new-model
!
resource policy
!
ip cef
```

Entire conversation (1858 bytes)

● ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ○ Raw

Notice that there is a mixture of highlighted text in both red and blue. The red highlighted text represents text that is received while the blue highlighted text represents text that is sent.

Based on this output, you can see how easy it is for someone who is in the path of sensitive data to view unencrypted or clear text traffic.

Now, you will reapply the cryptography settings on R1 and R3 and begin a telnet session from R1 to R3 as before.

Begin by reapplying the crypto maps you removed earlier on R1 and R3.

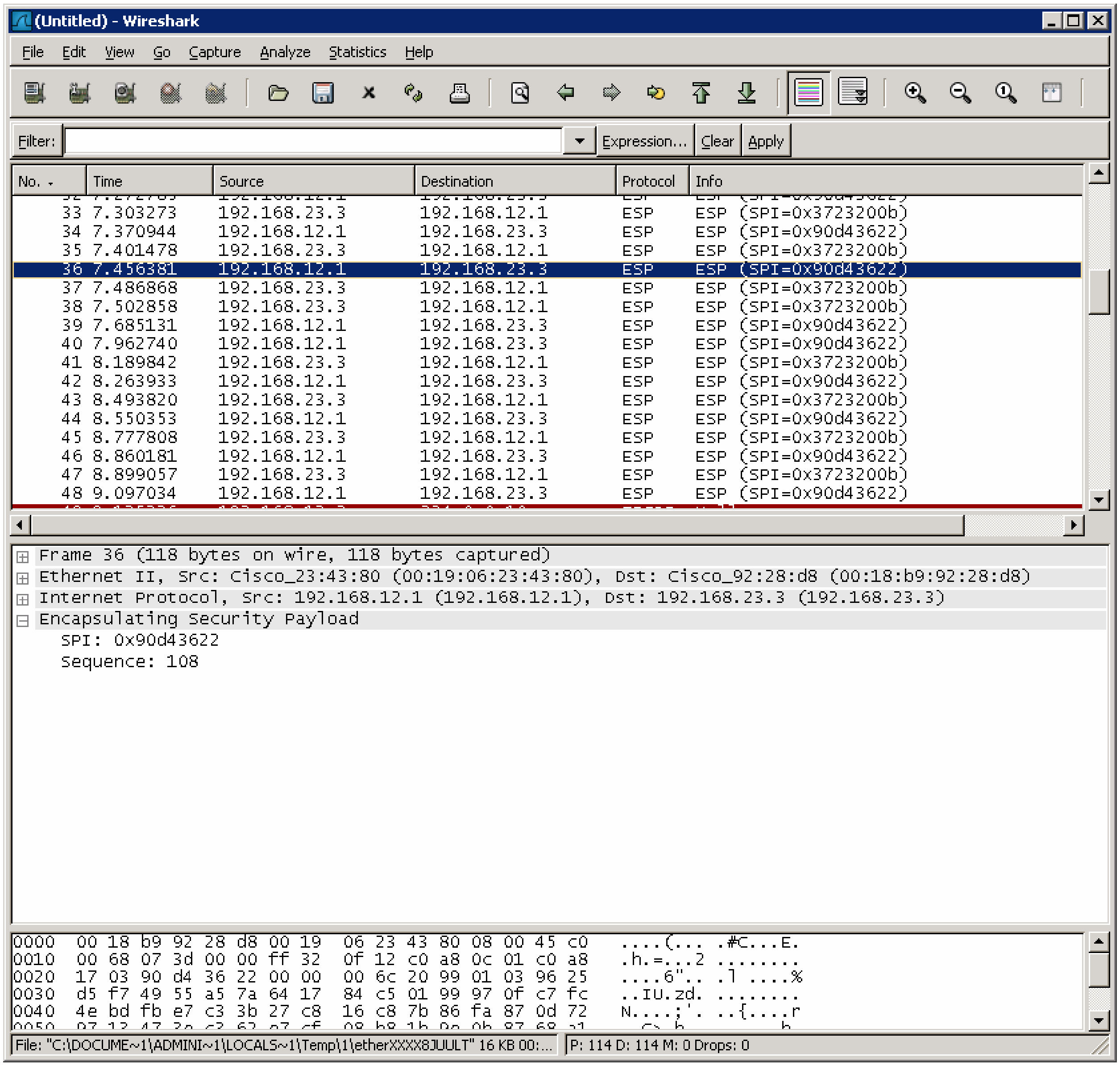
```
R1(config)# interface fastethernet0/0
R1(config-if)# crypto map SDM_CMAP_1
*Jan 16 06:36:10.295: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON

R3(config)# interface serial0/0/1
R3(config-if)# crypto map SDM_CMAP_1
*Jan 16 06:37:59.798: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

Start the packet capturing again in Wireshark, and then issue the same telnet sequence as you did previously.

```
R1# telnet 172.16.3.1 /source Loopback0
Trying 172.16.3.1 ... Open


User Access Verification

Password: [cisco]
R3> en
Password: [cisco]

R3# show ip interface brief
Interface            IP-Address      OK? Method Status                Protocol
FastEthernet0/0      unassigned      YES unset  administratively down down
FastEthernet0/1      unassigned      YES unset  administratively down down
Serial0/0/0          unassigned      YES unset  administratively down down
Serial0/0/1          192.168.23.3    YES manual up                    up
Serial0/1/0          unassigned      YES unset  administratively down down
Serial0/1/1          unassigned      YES unset  administratively down down
Loopback0            172.16.3.1      YES manual up                    up

R3# exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

End your Wireshark capture when you are finished with the telnet session.

As far as the user is concerned, the telnet session seems the same with and without encryption. However, the packet capture from Wireshark shows that the VPN is actively encapsulating and encrypting packets.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No.  Time      Source          Destination     Protocol  Info
 33 7.303273   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 34 7.370944   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
 35 7.401478   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 36 7.456381   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
 37 7.486868   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 38 7.502858   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 39 7.685131   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
 40 7.962740   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
 41 8.189842   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 42 8.263933   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
 43 8.493820   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 44 8.550353   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
 45 8.777808   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 46 8.860181   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
 47 8.899057   192.168.23.3    192.168.12.1    ESP       ESP (SPI=0x3723200b)
 48 9.097034   192.168.12.1    192.168.23.3    ESP       ESP (SPI=0x90d43622)
```

```
⊞ Frame 36 (118 bytes on wire, 118 bytes captured)
⊞ Ethernet II, Src: Cisco_23:43:80 (00:19:06:23:43:80), Dst: Cisco_92:28:d8 (00:18:b9:92:28:d8)
⊞ Internet Protocol, Src: 192.168.12.1 (192.168.12.1), Dst: 192.168.23.3 (192.168.23.3)
⊟ Encapsulating Security Payload
    SPI: 0x90d43622
    Sequence: 108
```

```
0000  00 18 b9 92 28 d8 00 19  06 23 43 80 08 00 45 c0   ....(... .#C...E.
0010  00 68 07 3d 00 00 ff 32  0f 12 c0 a8 0c 01 c0 a8   .h.=...2 ........
0020  17 03 90 d4 36 22 00 00  00 6c 20 99 01 03 96 25   ....6".. .l ....%
0030  d5 f7 49 55 a5 7a 64 17  84 c5 01 99 97 0f c7 fc   ..IU.zd. ........
0040  4e bd fb e7 c3 3b 27 c8  16 c8 7b 86 fa 87 0d 72   N....;'. ..{....r
```

```
File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\etherXXXX8JUULT" 16 KB 00:...   P: 114 D: 114 M: 0 Drops: 0
```

Figure 8-3: Detailed Packet Data on Encrypted Telnet String Sent From R1

Notice that the protocol is not telnet (TCP port 23), but the Encapsulating Security Protocol (ESP, IP protocol number 50). Remember, all traffic here matches the IPsec access list.

Also notice that the source and destination are not the actual source and destination of the addresses participating in this telnet conversation. Rather, they are the endpoints of the VPN.

Why do you use the VPN endpoints as the source and destination of packets?

Finally, and most important, if you look at the contents of these packets in Wireshark, no matter how you try to format or filter them, you will not be able to see what data was originally inside.

The encryption suite provided by IPsec successfully secures data through authentication, encryption, and data-integrity services.

Appendix A: TCL Script Output

```
R1# tclsh
R1(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
R1(tcl)# tclquit

R2# tclsh
R2(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/68 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
R2(tcl)# tclquit

R3# tclsh
R3(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/64 ms
R3(tcl)# tclquit
```

Final Configurations

```
R1# show run
!
hostname R1
!
crypto pki trustpoint TP-self-signed-1455051929
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1455051929
 revocation-check none
 rsakeypair TP-self-signed-1455051929
!
crypto pki certificate chain TP-self-signed-1455051929
 certificate self-signed 01
<OUTPUT OMITTED>
  8EAF0758 8E56E4F8 68C2872C 1BA64531 80ED01B7 84EB790C 43312206 575C
  quit
username ciscosdm privilege 15 password 0 ciscosdm
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 encr aes 256
 hash md5
 authentication pre-share
 group 5
 lifetime 28800
crypto isakmp key cisco address 192.168.23.3
!
crypto IPsec transform-set cisco_lab_transform esp-aes 256 esp-sha-hmac
!
crypto map SDM_CMAP_1 1 IPsec-isakmp
 description Tunnel to192.168.23.3
 set peer 192.168.23.3
 set transform-set cisco_lab_transform
 match address 101
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 crypto map SDM_CMAP_1
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 network 192.168.12.0
 no auto-summary
!
!
!
ip http server
ip http authentication local
ip http secure-server
!
access-list 100 remark SDM_ACL Category=4
access-list 100 remark IPsec Rule
access-list 100 permit ip 172.16.1.0 0.0.0.255 172.16.3.0 0.0.0.255
access-list 101 remark SDM_ACL Category=4
```

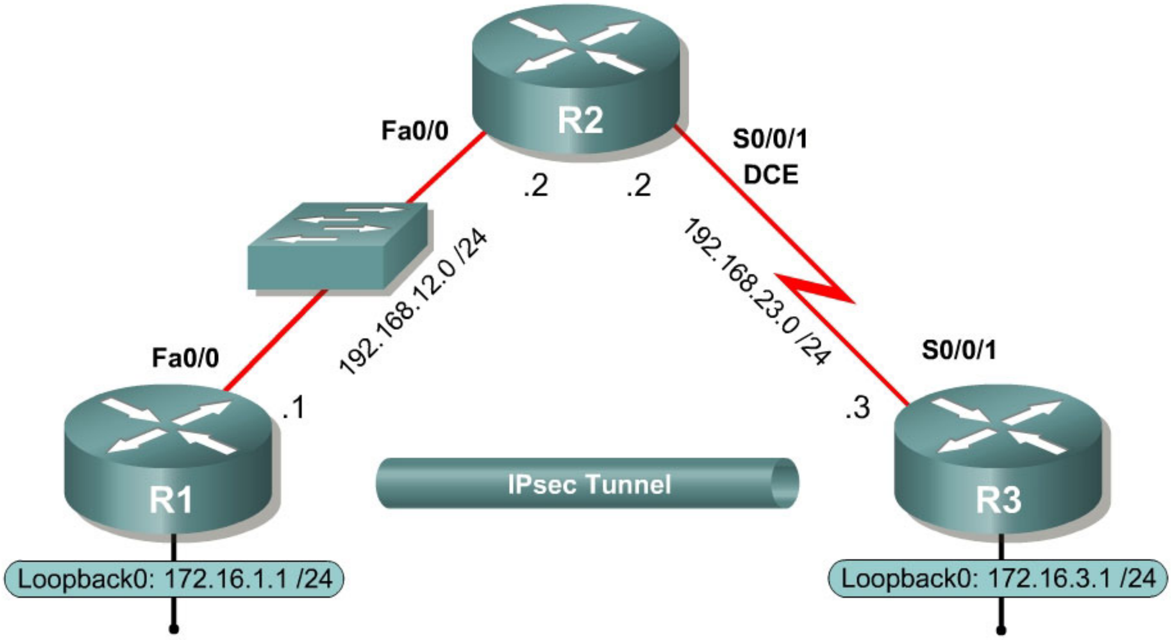
```
access-list 101 remark IPsec Rule
access-list 101 permit ip 172.16.1.0 0.0.0.255 172.16.3.0 0.0.0.255
!
line vty 0 4
 login local
 transport input telnet ssh
!
end

R2# show run
!
hostname R2
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 192.168.12.0
 network 192.168.23.0
 no auto-summary
!
end

R3# show run
!
hostname R3
!
enable secret 5 $1$gJqP$HsL/xMjpFvacHs7bWGvIK.
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 encr aes 256
 hash md5
 authentication pre-share
 group 5
 lifetime 28800
crypto isakmp key cisco address 192.168.12.1
!
crypto IPsec transform-set cisco_lab_transform esp-aes 256 esp-sha-hmac
!
crypto map SDM_CMAP_1 1 IPsec-isakmp
 description # Apply the crypto map on the peer router's interface having IP
address 192.168.23.3 that connects to this router.
 set peer 192.168.12.1
 set transform-set cisco_lab_transform
 match address SDM_1
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map SDM_CMAP_1
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 network 192.168.23.0
 no auto-summary
!
ip access-list extended SDM_1
 remark SDM_ACL Category=4
 remark IPsec Rule
 permit ip 172.16.3.0 0.0.0.255 172.16.1.0 0.0.0.255
!
line vty 0 4
 password cisco
 login
!
end
```

Lab 3.5 Configuring Site-to-Site IPsec VPNs with the IOS CLI

Learning Objectives

- Configure EIGRP on the routers
- Create a site-to-site IPsec VPN using IOS
- Verify IPsec operation

Topology Diagram



Scenario

In this lab, you will configure a site-to-site IPsec VPN. Once you have configured the VPN, the traffic between the loopback interfaces on R1 and R3 will be encrypted.

You will use the Cisco IOS command-line interface (CLI) for this lab exercise. Lab 3.4 involves the same function as this exercise, but implemented via the Cisco Security Device Manager (SDM).

Step 1: Configure Addressing

Configure the loopback interfaces and the serial interfaces with the addresses shown in the diagram. Set the clock rates on the appropriate interfaces and issue the **no shutdown** command on all physical connections. Verify that you have connectivity across local subnets using the **ping** command.

```
R1(config)# interface loopback0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface fastethernet0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown

R2(config)# interface fastethernet0/0
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/0/1
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3(config)# interface loopback0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface serial0/0/1
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Configure EIGRP

In order to maintain connectivity between remote networks, configure EIGRP to route between all networks in the diagram. Add all connected subnets into the EIGRP autonomous system on every router. Disable automatic summarization.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.0.0
R1(config-router)# network 192.168.12.0

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 192.168.12.0
R2(config-router)# network 192.168.23.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
R3(config-router)# network 172.16.0.0
R3(config-router)# network 192.168.23.0
```

Verify that you have full IP connectivity at this point using the following TCL script.

```
tclsh

foreach address {
172.16.1.1
192.168.12.1
192.168.12.2
192.168.23.2
172.16.3.1
192.168.23.3
} { ping $address }

tclquit
```

Compare your output with the output shown in Appendix A. Troubleshoot as necessary.

Step 3: Create IKE Policies

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF). It provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers.

Since IPSec is a framework, it allows us to exchange security protocols as new technologies (including encryption algorithms) are developed.

There are two central configuration elements to the implementation of an IPsec VPN:

1. Implement Internet Key Exchange (IKE) parameters
2. Implement IPsec parameters

The exchange method employed by IKE is first used to pass and validate IKE policies between peers. Then, the peers exchange and match IPSec policies for the authentication and encryption of data traffic. The IKE policy controls the authentication, encryption algorithm, and key exchange method used for IKE proposals that are sent and received by the IPSec endpoints. The IPSec policy is used to encrypt data traffic sent through the VPN tunnel.

IKE will need to be enabled for IPsec to work. IKE is enabled by default on IOS images with cryptographic feature sets. If it is disabled for some reason, you can enable it with the command **crypto isakmp enable**.

```
R1(config)# crypto isakmp enable
```

If you cannot execute this command on the router, you need to upgrade the IOS image to an image with a feature set that includes the Cisco cryptographic services.

The exchange method employed by IKE is first used to pass and validate IKE policies between peers. Then, the peers exchange and match IPsec policies for the authentication and encryption of data traffic. The IKE policy controls the authentication, encryption algorithm, and key exchange method that is used by IKE proposals that are sent and received by the IPsec endpoints. The IPsec policy is used to encrypt data traffic that is sent through the VPN tunnel.

To allow IKE Phase 1 negotiation, you must create an Internet Security Association and Key Management Protocol (ISAKMP) policy and configure a peer association involving that ISAKMP policy. An ISAKMP policy defines the authentication and encryption algorithms and hash function used to send control traffic between the two VPN endpoints. When an ISAKMP security association has been accepted by the IKE peers, IKE Phase 1 has been completed. IKE Phase 2 parameters will be configured later in the lab.

Issue the **crypto isakmp policy** *number* command in global configuration mode. This initiates the ISAKMP policy configuration mode. Once in this mode, you can view the various IKE parameters available by typing **?**. Enter into this configuration mode on R1 for policy 10, and view some of the possible settings.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# ?
ISAKMP commands:
  authentication  Set authentication method for protection suite
  default         Set a command to its defaults
  encryption      Set encryption algorithm for protection suite
  exit            Exit from ISAKMP protection suite configuration mode
  group           Set the Diffie-Hellman group
  hash            Set hash algorithm for protection suite
  lifetime        Set lifetime for ISAKMP security association
  no              Negate a command or set its defaults
```

Your choice of an encryption algorithm controls how confidential the control channel between the endpoints will be. The hash algorithm controls data integrity, that is, surety that the data received from a peer has not been tampered with in transit. The authentication type ensures that the packet was indeed sent and signed by the remote peer. The Diffie-Hellman group is used to create a secret key shared by the peers that has never been sent across the network.

Configure an authentication type of pre-shared keys. Use AES 256 encryption, SHA as your hash algorithm, and Diffie-Hellman group 5 for this IKE policy.

Give the policy a lifetime of 3600 seconds (one hour). Configure the same policy on R3. Older versions of the IOS do not support AES 256 encryption and/or SHA as your hash algorithm. Substitute whatever encryption and hashing algorithm your router supports. Be sure the same changes are made on the other VPN endpoint so that they are the same.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# hash sha
R1(config-isakmp)# group 5
R1(config-isakmp)# lifetime 3600

R3(config)# crypto isakmp policy 10
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# encryption aes 256
R3(config-isakmp)# hash sha
R3(config-isakmp)# group 5
R3(config-isakmp)# lifetime 3600
```

Although you only need to configure one policy here, you can configure multiple IKE policies. The different priority numbers refer to how secure a policy is. The lower the policy number is, the more secure a policy is. Routers will check to verify which security policies are compatible with their peer, starting with the lowest numbered (most secure) policies. You can verify your IKE policy with the **show crypto isakmp policy** command. Note that a default, less secure policy already exists on the router.

```
R1# show crypto isakmp policy

Global IKE policy
Protection suite of priority 10
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit
keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               3600 seconds, no volume limit
Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit

R3# show crypto isakmp policy

Global IKE policy
Protection suite of priority 10
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit
keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               3600 seconds, no volume limit
Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit
```

Step 4: Configure Pre-Shared Keys

Since we chose pre-shared keys as our authentication method in the IKE policy, we must configure a key on each router corresponding to the other VPN endpoint. These keys must match up for authentication to be successful and for the IKE peering to be completed. For simplicity we can use the key “cisco”. A production network should use a more complex key.

Use the global configuration command **crypto isakmp key** *key-string* **address** *address* to enter a pre-shared key. Use the IP address of the remote peer. Ensure that the IP address is the remote interface that the peer would use to route traffic to the local router.

Which IP addresses should you use to configure the IKE peers, given the topology diagram?

Each IP address that is used to configure the IKE peers are also referred to as the IP address of the remote VPN endpoint. You can also specify the peer by hostname (substitute the keyword **address** with **hostname**) if the IP address may change a lot. You either have to statically bind the IP address to the hostname on the router, or use a name-lookup service.

```
R1(config)# crypto isakmp key cisco address 192.168.23.3
```

```
R3(config)# crypto isakmp key cisco address 192.168.12.1
```

Step 5: Configure the IPsec Transform Set and Lifetimes

The IPsec transform set is another crypto configuration parameter that routers negotiate to form a security association. In the same way that ISAKMP policies can, multiple transform sets can exist on a router. Routers will compare their transform sets to the remote peer until they find a transform set that matches exactly.

Create an IPsec transform set, using the syntax **crypto ipsec transform-set** *tag parameters*. Use **?** to see what parameters are available. For R1 and R3, create a transform set with tag 50 and use an ESP transform with an AES 256 cipher first, with Encapsulating Security Protocol (ESP) and the SHA hash function and finally an authentication header using SHA.

```
R1(config)#crypto ipsec transform-set ?
  WORD  Transform set tag

R1(config)#crypto ipsec transform-set 50 ?
  ah-md5-hmac   AH-HMAC-MD5 transform
  ah-sha-hmac   AH-HMAC-SHA transform
  comp-lzs      IP Compression using the LZS compression algorithm
  esp-3des      ESP transform using 3DES(EDE) cipher (168 bits)
  esp-aes       ESP transform using AES cipher
  esp-des       ESP transform using DES cipher (56 bits)
  esp-md5-hmac  ESP transform using HMAC-MD5 auth
  esp-null      ESP transform w/o cipher
  esp-seal      ESP transform using SEAL cipher (160 bits)
  esp-sha-hmac  ESP transform using HMAC-SHA auth

R1(config)#crypto ipsec transform-set 50  esp-aes ?
  128           128 bit keys.
  192           192 bit keys.
  256           256 bit keys.
  ah-md5-hmac   AH-HMAC-MD5 transform
  ah-sha-hmac   AH-HMAC-SHA transform
  comp-lzs      IP Compression using the LZS compression algorithm
  esp-md5-hmac  ESP transform using HMAC-MD5 auth
  esp-sha-hmac  ESP transform using HMAC-SHA auth
  <cr>

R1(config)#crypto ipsec transform-set 50  esp-aes 256 ?
  ah-md5-hmac   AH-HMAC-MD5 transform
  ah-sha-hmac   AH-HMAC-SHA transform
  comp-lzs      IP Compression using the LZS compression algorithm
  esp-md5-hmac  ESP transform using HMAC-MD5 auth
  esp-sha-hmac  ESP transform using HMAC-SHA auth
  <cr>

R1(config)#crypto ipsec transform-set 50  esp-aes 256 esp-sha-hmac ?
  ah-md5-hmac  AH-HMAC-MD5 transform
  ah-sha-hmac  AH-HMAC-SHA transform
  comp-lzs     IP Compression using the LZS compression algorithm
  <cr>
```

Executing the above command sends you into the transform set configuration mode, although you can just type **exit** to leave it since you do not need to configure any additional transform parameters.

```
R1(config)# crypto ipsec transform-set 50 esp-aes 256 esp-sha-hmac ah-sha-hmac
R1(cfg-crypto-trans)# exit
R1(config)#

R3(config)# crypto ipsec transform-set 50 esp-aes 256 esp-sha-hmac ah-sha-hmac
R3(cfg-crypto-trans)# exit
R3(config)#
```

What is the function of the IPsec transform set?

You can also change the IPsec security association lifetimes from its default which is 3600 seconds or 4,608,000 kilobytes, whichever comes first. Change this with the global configuration command **crypto ipsec security-association lifetime seconds** *seconds* or **crypto ipsec security-association lifetime kilobytes** *kilobytes*. On R1 and R3, set the IPsec security association lifetime to 30 minutes, or 1800 seconds.

```
R1(config)# crypto ipsec security-association lifetime seconds 1800

R3(config)# crypto ipsec security-association lifetime seconds 1800
```

Step 6: Define Interesting Traffic

Now that most of the encryption settings are in place, define extended access lists to tell the router which traffic to encrypt. Like other access lists used to define “interesting traffic” rather than packet filtering, permit and deny do not have the usual meaning of a filtering access list. A packet which is permitted by an access list used for defining IPsec traffic will get encrypted if the IPsec session is configured correctly. A packet that is denied by one of these access lists will not be dropped; it will be sent unencrypted. Also, like any other access list, there is an implicit **deny** at the end, which in this case means the default action is to not encrypt traffic. If there is no IPsec security association correctly configured, then no traffic will be encrypted, but traffic will be forwarded as unencrypted traffic.

In this scenario, the traffic you want to encrypt is traffic going from R1’s loopback network to R3’s loopback network, or vice versa. These access lists are used outbound on the VPN endpoint interfaces, so configure them accordingly. The configuration of R1’s access list will need to be mirrored exactly on R3 for this to work properly.

```
R1(config)# access-list 101 permit ip 172.16.1.0 0.0.0.255 172.16.3.0
0.0.0.255
```

```
R3(config)# access-list 101 permit ip 172.16.3.0 0.0.0.255 172.16.1.0
0.0.0.255
```

Does IPsec evaluate whether the access lists are mirrored as a requirement to negotiate its security association?

Step 7: Create and Apply Crypto Maps

Now that you have created all of these small configuration modules, you can bring them together in a crypto map. A crypto map is a mapping that associates traffic matching an access list (like the one we created earlier) to a peer and various IKE and IPsec settings. Crypto maps can have multiple map statements, so you can have traffic that matches a certain access list being encrypted and sent to one IPsec peer, and have other traffic that matches a different access list being encrypted towards a different peer. After a crypto map is created, it can be applied to one or more interfaces. The interface(s) that it is applied to should be the one(s) facing the IPsec peer.

To create a crypto map, use the global configuration command **crypto map** *name sequence-num type* to enter the crypto map configuration mode for that sequence number. Multiple crypto map statements can belong to the same crypto map, and they will be evaluated in ascending numerical order.

Use a type of **ipsec-isakmp**, which means IKE will be used to establish IPsec security associations. Under normal circumstances, you would want to use this mode, as opposed to the **ipsec-manual** type. If ipsec-manual is used, IKE will not be used to configure the IPsec security association (SA). (This mode is beyond the scope of this lab.) Name the crypto map “MYMAP,” and use 10 as the sequence number. Enter the crypto map configuration mode on R1. When you do this, the crypto map will be created and this command will generate a warning that a peer must be fully configured before the crypto map is considered valid and can be actively applied.

```
R1(config)# crypto map MYMAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
```

Use the **match address** *access-list* command to specify which access list will define which traffic to encrypt. If you have ever configured route maps or similar maps on a router before, some of these map-related commands may seem familiar.

```
R1(config-crypto-map)# match address 101
```

There are many possible **set** commands that you can do in a crypto map. To view the list of possibilities, use the **?** help character.

```
R1(config-crypto-map)# set ?
  identity              Identity restriction.
  ip                    Interface Internet Protocol config commands
  isakmp-profile        Specify isakmp Profile
  nat                   Set NAT translation
  peer                  Allowed Encryption/Decryption peer.
  pfs                   Specify pfs settings
  security-association  Security association parameters
  transform-set         Specify list of transform sets in priority order
```

Setting a peer IP or hostname is required, so set it to R3’s remote VPN endpoint interface using the **set peer** *address* command. Hardcode the transform set to be used with this peer, using the **set transform-set** *tag* command.

Set the perfect forwarding secrecy type using the **set pfs** *type* command, and also modify the default IPsec security association lifetime with the **set security-association lifetime seconds** *seconds* command. As you can see in the output of the **?** above, there are more settings you can change in this crypto map. Create a matching crypto map on R3 using the mirrored access list to define interesting traffic.

```
R1(config-crypto-map)# set peer 192.168.23.3
R1(config-crypto-map)# set pfs group5
R1(config-crypto-map)# set transform-set 50
R1(config-crypto-map)# set security-association lifetime seconds 900

R3(config)# crypto map MYMAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R3(config-crypto-map)# match address 101
R3(config-crypto-map)# set peer 192.168.12.1
R3(config-crypto-map)# set pfs group5
R3(config-crypto-map)# set transform-set 50
R3(config-crypto-map)# set security-association lifetime seconds 900
```

Now that the crypto maps are created, the last step in the process of creating site-to-site VPNs is applying the maps to interfaces. This is done with the interface level **crypto map** *name* command. Note that the SAs will not be established until the crypto map has been activated by interesting traffic. Do not create interesting traffic yet, because you will want to enable some debugging during the next step. The router will generate a notification that crypto is now on.

```
R1(config)# interface fastethernet0/0
R1(config-if)# crypto map MYMAP
*Jan 17 04:09:09.150: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON

R3(config)# interface serial0/0/1
R3(config-if)# crypto map MYMAP
*Jan 17 04:10:54.138: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

Step 8: Verify IPsec Configuration

In Step 3, you used the **show crypto isakmp policy** command to show the configured ISAKMP policies on the router. Similarly, the **show crypto ipsec transform-set** command displays the configured IPsec policies in the form of the transport sets.

```
R1# show crypto ipsec transform-set
Transform set 50: { ah-sha-hmac  }
   will negotiate = { Tunnel,  },
   { esp-256-aes esp-sha-hmac  }
   will negotiate = { Tunnel,  },

R3# show crypto ipsec transform-set
Transform set 50: { ah-sha-hmac  }
   will negotiate = { Tunnel,  },
   { esp-256-aes esp-sha-hmac  }
   will negotiate = { Tunnel,  },
```

Use the **show crypto map** command to display the crypto maps that will be applied to the router.

```
R1# show crypto map
Crypto Map "MYMAP" 10 ipsec-isakmp
        Peer = 192.168.23.3
        Extended IP access list 101
           access-list 101 permit ip 172.16.1.0 0.0.0.255 172.16.3.0 0.0.0.255
        Current peer: 192.168.23.3
        Security association lifetime: 4608000 kilobytes/900 seconds
        PFS (Y/N): Y
        DH group:  group5
        Transform sets={
                50,
        }
        Interfaces using crypto map MYMAP:
                FastEthernet0/0

R3# show crypto map
Crypto Map "MYMAP" 10 ipsec-isakmp
        Peer = 192.168.12.1
        Extended IP access list 101
           access-list 101 permit ip 172.16.3.0 0.0.0.255 172.16.1.0 0.0.0.255
        Current peer: 192.168.12.1
        Security association lifetime: 4608000 kilobytes/900 seconds
        PFS (Y/N): Y
        DH group:  group5
        Transform sets={
                50,
        }
        Interfaces using crypto map MYMAP:
                Serial0/0/1
```

The output of these **show** commands will not change if interesting traffic goes across the connection.

Step 9: Verify IPsec Operation

If you use the **show crypto isakmp sa** command, it will reveal that no IKE SAs exist yet. Once we send some interesting traffic later in this lab, this command output will change.

```
R1# show crypto isakmp sa
dst             src             state          conn-id slot status

R3# show crypto isakmp sa
dst             src             state          conn-id slot status
```

If you use the **show crypto ipsec sa** command, this command will show the unused SA between R1 and R3. Note the number of packets sent across and the lack of any security associations listed towards the bottom of the output.

```
R1# show crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: MYMAP, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 192.168.12.1, remote crypto endpt.: 192.168.23.3
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
     current outbound spi: 0x0(0)

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

     outbound ah sas:

     outbound pcp sas:
```

```
R3# show crypto ipsec sa

interface: Serial0/0/1
    Crypto map tag: MYMAP, local addr 192.168.23.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   current_peer 192.168.12.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 192.168.23.3, remote crypto endpt.: 192.168.12.1
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/1
     current outbound spi: 0x0(0)

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

     outbound ah sas:

     outbound pcp sas:
```

Why have no security associations (SAs) been negotiated?

How could you force the IPsec peers to negotiate their security association?

Step 10: Interpret IPsec Event Debugging

In terms of the actual communication between the VPN endpoints, ISAKMP prescribes stringent rules as to how an SA can be established. IKE Phase I (ISAKMP) will negotiate the secure channel between the endpoints, authenticate the neighbor as having the correct secret key, and authenticate the remote endpoint through the secure channel. IKE Phase I will use “main mode,” which consists of six messages in three event-driven exchanges. The result is one bidirectional ISAKMP security association. The exchanges are input/outputdriven, so every event is recorded in the debug as an input event from either the local router or the remote router.

IKE Phase II (IPsec) will negotiate the IPsec tunnel between the two endpoints, authenticate the peers, and encrypt data traffic between them through the encrypted tunnel. IKE Phase II uses the process called “quick mode” to perform its exchange to establish two unidirectional security associations.

On R1, enable two debug commands: **debug crypto isakmp** and **debug crypto ipsec**.

```
R1# debug crypto isakmp
Crypto ISAKMP debugging is on
R1# debug crypto ipsec
Crypto IPSEC debugging is on
```

Now, send an extended ping from R1’s loopback to R3’s loopback, and watch the debug outputs on both routers. You will see both ISAKMP negotiation and IPsec SAs being established. This output is *very* verbose.

```
R1# ping
Protocol [ip]:
Target IP address: 172.16.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.1.1
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]:
```

At this point, the packets will be sent. What will happen next?

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.1.1

*Jan 17 05:11:39.142: IPSEC(sa_request): ,
  (key eng. msg.) OUTBOUND local= 192.168.12.1, remote= 192.168.23.3,
    local_proxy= 172.16.1.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 172.16.3.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= NONE  (Tunnel),
    lifedur= 900s and 4608000kb,
    spi= 0x0(0), conn_id= 0, keysize= 256, flags= 0x0
...
```

When R1 detects interesting traffic going toward R3, the **crypto map** statement defined on R1’s Fast Ethernet interface invokes an IPsec state change from inactive to active. The IPsec suite attempts to raise a security association between R1 and R3 to pass secure traffic with the IPsec parameters that were previously configured. The ISAKMP processes on each of the VPN endpoints are now aware that a security association will be attempted and prepare to send ISAKMP policies.

```
...
*Jan 17 05:11:39.146: ISAKMP:(0): SA request profile is (NULL)
*Jan 17 05:11:39.146: ISAKMP: Created a peer struct for 192.168.23.3, peer
port 500
*Jan 17 05:11:39.146: ISAKMP: New peer created peer = 0x46F56220 peer_handle =
0x80000002
*Jan 17 05:11:39.146: ISAKMP: Locking peer struct 0x46F56220, refcount 1 for
isakmp_initiator
*Jan 17 05:11:39.146: ISAKMP: local port 500, remote port 500
*Jan 17 05:11:39.146: ISAKMP: set new node 0 to QM_IDLE
*Jan 17 05:11:39.146: insert sa successfully sa = 477B9850
...
```

The Cisco IOS software builds an ISAKMP peer structure in memory, providing a means to store parameters and policies related to IKE Phase I key exchanges. The peers will communicate on port 500 on both ends between the IP addresses of the encrypting interfaces on each router. ISAKMP creates and inserts a memory structure representing the ISAKMP security association into the peer structure it just created.

```
...
*Jan 17 05:11:39.146: ISAKMP:(0):Can not start Aggressive mode, trying Main
mode
*Jan 17 05:11:39.146: ISAKMP:(0):found peer pre-shared key matching
192.168.23.3
*Jan 17 05:11:39.146: ISAKMP:(0): constructed NAT-T vendor-07 ID
*Jan 17 05:11:39.150: ISAKMP:(0): constructed NAT-T vendor-03 ID
*Jan 17 05:11:39.150: ISAKMP:(0): constructed NAT-T vendor-02 ID
*Jan 17 05:11:39.150: ISAKMP:(0):Input = IKE_MESG_FROM_IPSEC, IKE_SA_REQ_MM
*Jan 17 05:11:39.150: ISAKMP:(0):Old State = IKE_READY  New State = IKE_I_MM1
...
```

Aggressive mode is an exchange process in which all of IKE Phase I is negotiated with one exchange. This aggressive mode is clearly less secure than main mode, which relies on three exchanges: the ISAKMP policy exchange, the Diffie-Hellman key exchange, and an encrypted authentication test that initiates the ISAKMP security association used for Phase II. In main mode, less information is given to the remote node before the remote node must communicate and can be authenticated.

```
...
*Jan 17 05:11:39.150: ISAKMP:(0): beginning Main Mode exchange
*Jan 17 05:11:39.150: ISAKMP:(0): sending packet to 192.168.23.3 my_port 500
peer_port 500 (I) MM_NO_STATE
*Jan 17 05:11:39.282: ISAKMP (0:0): received packet from 192.168.23.3 dport
500 sport 500 Global (I) MM_NO_STATE
*Jan 17 05:11:39.286: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Jan 17 05:11:39.286: ISAKMP:(0):Old State = IKE_I_MM1  New State = IKE_I_MM2

*Jan 17 05:11:39.286: ISAKMP:(0): processing SA payload. message ID = 0

.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 52/54/56 ms
```

```
R1#

*Jan 17 05:11:39.286: ISAKMP:(0): processing vendor id payload
*Jan 17 05:11:39.286: ISAKMP:(0): vendor ID seems Unity/DPD but major 245
mismatch
*Jan 17 05:11:39.286: ISAKMP (0:0): vendor ID is NAT-T v7
*Jan 17 05:11:39.286: ISAKMP:(0):found peer pre-shared key matching
192.168.23.3
*Jan 17 05:11:39.286: ISAKMP:(0): local preshared key found
*Jan 17 05:11:39.286: ISAKMP : Scanning profiles for xauth ...
*Jan 17 05:11:39.286: ISAKMP:(0):Checking ISAKMP transform 1 against priority
10 policy
*Jan 17 05:11:39.286: ISAKMP:      encryption AES-CBC
*Jan 17 05:11:39.286: ISAKMP:      keylength of 256
*Jan 17 05:11:39.286: ISAKMP:      hash SHA
*Jan 17 05:11:39.286: ISAKMP:      default group 5
*Jan 17 05:11:39.286: ISAKMP:      auth pre-share
*Jan 17 05:11:39.286: ISAKMP:      life type in seconds
*Jan 17 05:11:39.286: ISAKMP:      life duration (basic) of 3600
*Jan 17 05:11:39.286: ISAKMP:(0):atts are acceptable. Next payload is 0
*Jan 17 05:11:39.286: ISAKMP:(0): processing vendor id payload
*Jan 17 05:11:39.286: ISAKMP:(0): vendor ID seems Unity/DPD but major 245
mismatch
*Jan 17 05:11:39.286: ISAKMP (0:0): vendor ID is NAT-T v7
*Jan 17 05:11:39.290: ISAKMP:(0):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_MAIN_MODE
*Jan 17 05:11:39.290: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM2
...
```

During the first exchange, the initiator sends its policy to the endpoint of the ISAKMP security association in the first message and the endpoint responds with its ISAKMP security policy in the second message.

Notice that each of the IKE peers sends the ISAKMP security policy to the other. R1 then processes the payload of the packet it received from R3 and determines that it has a pre-shared key associated with R3's address. R1 matches the ISAKMP transform set (policy) from R3 against its own priority 10 policy. After this check is performed, R1 determines that the attributes of the ISAKMP policy are acceptable and signals the ISAKMP process to continue to the next main mode exchange. Finally, R1 informs R3 that it has accepted the policy and entered the second exchange by beginning the second exchange with R3.

Also, note that the Internet Control Message Protocol (ICMP) packets passed through the tunnel although the debug output has not finished displaying to the screen. The entire exchange took about one second to perform, according to the timestamps on the debug messages.

```
...
*Jan 17 05:11:39.290: ISAKMP:(0): sending packet to 192.168.23.3 my_port 500
peer_port 500 (I) MM_SA_SETUP
*Jan 17 05:11:39.290: ISAKMP:(0):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_COMPLETE
*Jan 17 05:11:39.290: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM3
*Jan 17 05:11:39.502: ISAKMP (0:0): received packet from 192.168.23.3 dport
500 sport 500 Global (I) MM_SA_SETUP
*Jan 17 05:11:39.502: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
```

```
*Jan 17 05:11:39.502: ISAKMP:(0):Old State = IKE_I_MM3  New State = IKE_I_MM4
*Jan 17 05:11:39.506: ISAKMP:(0): processing KE payload. message ID = 0
*Jan 17 05:11:39.638: ISAKMP:(0): processing NONCE payload. message ID = 0
*Jan 17 05:11:39.638: ISAKMP:(0):found peer pre-shared key matching
192.168.23.3
*Jan 17 05:11:39.642: ISAKMP:(1001): processing vendor id payload
*Jan 17 05:11:39.642: ISAKMP:(1001): vendor ID is Unity
*Jan 17 05:11:39.642: ISAKMP:(1001): processing vendor id payload
*Jan 17 05:11:39.642: ISAKMP:(1001): vendor ID is DPD
*Jan 17 05:11:39.642: ISAKMP:(1001): processing vendor id payload
*Jan 17 05:11:39.642: ISAKMP:(1001): speaking to another IOS box!
*Jan 17 05:11:39.642: ISAKMP:(1001):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_MAIN_MODE
*Jan 17 05:11:39.642: ISAKMP:(1001):Old State = IKE_I_MM4  New State =
IKE_I_MM4
...
```

During main mode’s second exchange, the ISAKMP initiator sends the remote endpoint an RSA nonce (random number), to be used in the Diffie-Hellman algorithm, and the key it receives as the output of the Diffie-Hellman function. This is the third message in the main mode process. The remote endpoint R3 then replies in the fourth message of main mode with its respective nonce and key. R1 and R3 authenticate each other as both sharing the same pre-shared key used to generate the key it received from the peer.

At this point, R1 prepares to send a packet to R3 that will be passed through the secure channel. This packet will be used by R3 to authenticate R1 as the remote end of the ISAKMP security association.

```
...
*Jan 17 05:11:39.642: ISAKMP:(1001):Send initial contact
*Jan 17 05:11:39.646: ISAKMP:(1001):SA is doing pre-shared key authentication
using id type ID_IPV4_ADDR
*Jan 17 05:11:39.646: ISAKMP (0:1001): ID payload
        next-payload : 8
        type         : 1
        address      : 192.168.12.1
        protocol     : 17
        port         : 500
        length       : 12
*Jan 17 05:11:39.646: ISAKMP:(1001):Total payload length: 12
*Jan 17 05:11:39.646: ISAKMP:(1001): sending packet to 192.168.23.3 my_port
500 peer_port 500 (I) MM_KEY_EXCH
*Jan 17 05:11:39.646: ISAKMP:(1001):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_COMPLETE
*Jan 17 05:11:39.646: ISAKMP:(1001):Old State = IKE_I_MM4  New State =
IKE_I_MM5
*Jan 17 05:11:39.690: ISAKMP (0:1001): received packet from 192.168.23.3 dport
500 sport 500 Global (I) MM_KEY_EXCH
*Jan 17 05:11:39.690: ISAKMP:(1001): processing ID payload. message ID = 0
*Jan 17 05:11:39.690: ISAKMP (0:1001): ID payload
        next-payload : 8
        type         : 1
        address      : 192.168.23.3
        protocol     : 17
        port         : 500
        length       : 12
*Jan 17 05:11:39.690: ISAKMP:(0):: peer matches *none* of the profiles
*Jan 17 05:11:39.690: ISAKMP:(1001): processing HASH payload. message ID = 0
```

```
*Jan 17 05:11:39.690: ISAKMP:(1001):SA authentication status:
        authenticated
*Jan 17 05:11:39.690: ISAKMP:(1001):SA has been authenticated with
192.168.23.3
...
```

When R1 receives the authentication packet from R3, it checks the ID payload and hash value of the packet. If ISAKMP authenticates the security association through the encrypted channel, IKE Phase I is complete and the security association has been established.

```
...
*Jan 17 05:11:39.690: ISAKMP: Trying to insert a peer
192.168.12.1/192.168.23.3/500/,  and inserted successfully 46F56220.
*Jan 17 05:11:39.690: ISAKMP:(1001):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Jan 17 05:11:39.690: ISAKMP:(1001):Old State = IKE_I_MM5  New State =
IKE_I_MM6
*Jan 17 05:11:39.694: ISAKMP:(1001):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_MAIN_MODE
*Jan 17 05:11:39.694: ISAKMP:(1001):Old State = IKE_I_MM6  New State =
IKE_I_MM6
*Jan 17 05:11:39.694: ISAKMP:(1001):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_COMPLETE
*Jan 17 05:11:39.694: ISAKMP:(1001):Old State = IKE_I_MM6  New State =
IKE_P1_COMPLETE
...
```

Finally, R1 inserts the peer into the memory structure reserved at the beginning of the ISAKMP negotiation. R1 begins IKE Phase II over the ISAKMP security association created in IKE Phase I.

```
...
*Jan 17 05:11:39.694: ISAKMP:(1001):beginning Quick Mode exchange, M-ID of
787769575
*Jan 17 05:11:39.694: ISAKMP:(1001):QM Initiator gets spi
*Jan 17 05:11:39.698: ISAKMP:(1001): sending packet to 192.168.23.3 my_port
500 peer_port 500 (I) QM_IDLE
*Jan 17 05:11:39.698: ISAKMP:(1001):Node 787769575, Input = IKE_MESG_INTERNAL,
IKE_INIT_QM
*Jan 17 05:11:39.698: ISAKMP:(1001):Old State = IKE_QM_READY  New State =
IKE_QM_I_QM1
*Jan 17 05:11:39.698: ISAKMP:(1001):Input = IKE_MESG_INTERNAL,
IKE_PHASE1_COMPLETE
*Jan 17 05:11:39.698: ISAKMP:(1001):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
...
```

Quick mode uses three messages to create an IPsec security association. The initiator, R1, sends the first message including the hash, IPsec security association policies, a nonce, and a key created using the pre-shared keys and two ID payloads. R3 processes R1’s IPsec proposal and replies with a message with its values for the above parameters.

```
...
*Jan 17 05:11:40.014: ISAKMP (0:1001): received packet from 192.168.23.3 dport
500 sport 500 Global (I) QM_IDLE
*Jan 17 05:11:40.018: ISAKMP:(1001): processing HASH payload. message ID =
787769575
```

```
*Jan 17 05:11:40.018: ISAKMP:(1001): processing SA payload. message ID =
787769575
*Jan 17 05:11:40.018: ISAKMP:(1001):Checking IPSec proposal 1
*Jan 17 05:11:40.018: ISAKMP: transform 1, AH_SHA
*Jan 17 05:11:40.018: ISAKMP:   attributes in transform:
*Jan 17 05:11:40.018: ISAKMP:      encaps is 1 (Tunnel)
*Jan 17 05:11:40.018: ISAKMP:      SA life type in seconds
*Jan 17 05:11:40.018: ISAKMP:      SA life duration (basic) of 900
*Jan 17 05:11:40.018: ISAKMP:      SA life type in kilobytes
*Jan 17 05:11:40.018: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50
0x0
*Jan 17 05:11:40.018: ISAKMP:      group is 5
*Jan 17 05:11:40.018: ISAKMP:      authenticator is HMAC-SHA
*Jan 17 05:11:40.018: ISAKMP:(1001):atts are acceptable.
*Jan 17 05:11:40.018: ISAKMP:(1001):Checking IPSec proposal 1
*Jan 17 05:11:40.018: ISAKMP: transform 1, ESP_AES
*Jan 17 05:11:40.018: ISAKMP:   attributes in transform:
*Jan 17 05:11:40.018: ISAKMP:      encaps is 1 (Tunnel)
*Jan 17 05:11:40.018: ISAKMP:      SA life type in seconds
*Jan 17 05:11:40.018: ISAKMP:      SA life duration (basic) of 900
*Jan 17 05:11:40.018: ISAKMP:      SA life type in kilobytes
*Jan 17 05:11:40.018: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50
0x0
*Jan 17 05:11:40.018: ISAKMP:      authenticator is HMAC-SHA
*Jan 17 05:11:40.018: ISAKMP:      key length is 256
*Jan 17 05:11:40.018: ISAKMP:      group is 5
*Jan 17 05:11:40.018: ISAKMP:(1001):atts are acceptable.
...
```

R3 responds with a second similar message containing the same properties. R1 processes the hash and IPsec policy as shown above. R1 determines that the IPsec proposal from R3 is acceptable with the attributes in its own IPsec policies.

Notice that two transforms will both be used, with an authentication header (AH) applied to the ESP-encapsulated and encrypted data. A packet routed on R1 from 172.16.1.1 to 172.16.3.1 will first be encrypted and encapsulated as the payload of an ESP packet, and then the authentication header will be applied. The packet will be sent with a source IP address of 192.168.12.1 and destination IP address of 192.168.23.3.

```
...
*Jan 17 05:11:40.018: IPSEC(validate_proposal_request): proposal part #1
*Jan 17 05:11:40.018: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 192.168.12.1, remote= 192.168.23.3,
    local_proxy= 172.16.1.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 172.16.3.0/255.255.255.0/0/0 (type=4),
    protocol= AH, transform= ah-sha-hmac  (Tunnel),
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x0
*Jan 17 05:11:40.018: IPSEC(validate_proposal_request): proposal part #2
*Jan 17 05:11:40.018: IPSEC(validate_proposal_request): proposal part #2,
  (key eng. msg.) INBOUND local= 192.168.12.1, remote= 192.168.23.3,
    local_proxy= 172.16.1.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 172.16.3.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-aes 256 esp-sha-hmac  (Tunnel),
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 256, flags= 0x0
*Jan 17 05:11:40.022: Crypto mapdb : proxy_match
```

```
        src addr     : 172.16.1.0
        dst addr     : 172.16.3.0
        protocol     : 0
        src port     : 0
        dst port     : 0
*Jan 17 05:11:40.022: ISAKMP:(1001): processing NONCE payload. message ID =
787769575
*Jan 17 05:11:40.022: ISAKMP:(1001): processing KE payload. message ID =
787769575
*Jan 17 05:11:40.158: ISAKMP:(1001): processing ID payload. message ID =
787769575
*Jan 17 05:11:40.158: ISAKMP:(1001): processing ID payload. message ID =
787769575
...
```

R1 checks the proposal against its own IPsec policies, including the access lists on each end.

Note that the crypto map database could also evaluate the transport protocol or the source and destination ports using other properties in the extended access-list, but these were unused in the access list you supplied.

The nonce and keyed payload validate that two peers have the same keys. When the processing of this second message is complete, R1 must create the IPsec SAs so that interesting data can be encrypted using the policies negotiated.

Since there are two transforms in the transform set, two security associations and two proposals need to be created for each unidirectional path between the endpoints.

```
...
*Jan 17 05:11:40.158: ISAKMP:(1001): Creating IPSec SAs
*Jan 17 05:11:40.158:         inbound SA from 192.168.23.3 to 192.168.12.1
(f/i)  0/ 0
        (proxy 172.16.3.0 to 172.16.1.0)
*Jan 17 05:11:40.158:         has spi 0x588AA60C and conn_id 0
*Jan 17 05:11:40.158:         lifetime of 900 seconds
*Jan 17 05:11:40.158:         lifetime of 4608000 kilobytes
*Jan 17 05:11:40.158:         outbound SA from 192.168.12.1 to 192.168.23.3
(f/i) 0/0
        (proxy 172.16.1.0 to 172.16.3.0)
*Jan 17 05:11:40.162:         has spi  0x897F9209 and conn_id 0
*Jan 17 05:11:40.162:         lifetime of 900 seconds
*Jan 17 05:11:40.162:         lifetime of 4608000 kilobytes
*Jan 17 05:11:40.162: ISAKMP:(1001): Creating IPSec SAs
*Jan 17 05:11:40.162:         inbound SA from 192.168.23.3 to 192.168.12.1
(f/i)  0/ 0
        (proxy 172.16.3.0 to 172.16.1.0)
*Jan 17 05:11:40.162:         has spi 0x2E2954C0 and conn_id 0
*Jan 17 05:11:40.162:         lifetime of 900 seconds
*Jan 17 05:11:40.162:         lifetime of 4608000 kilobytes
*Jan 17 05:11:40.162:         outbound SA from 192.168.12.1 to 192.168.23.3
(f/i) 0/0
        (proxy 172.16.1.0 to 172.16.3.0)
*Jan 17 05:11:40.162:         has spi  0xAE4C8E5A and conn_id 0
*Jan 17 05:11:40.162:         lifetime of 900 seconds
*Jan 17 05:11:40.162:         lifetime of 4608000 kilobytes
```

```
*Jan 17 05:11:40.162: ISAKMP:(1001): sending packet to 192.168.23.3 my_port
500 peer_port 500 (I) QM_IDLE
*Jan 17 05:11:40.162: ISAKMP:(1001):deleting node 787769575 error FALSE reason
"No Error"
*Jan 17 05:11:40.162: ISAKMP:(1001):Node 787769575, Input =
IKE_MESG_FROM_PEER, IKE_QM_EXCH
*Jan 17 05:11:40.162: ISAKMP:(1001):Old State = IKE_QM_I_QM1  New State =
IKE_QM_PHASE2_COMPLETE
*Jan 17 05:11:40.166: IPSEC(key_engine): got a queue event with 1 KMI
message(s)
*Jan 17 05:11:40.166: Crypto mapdb : proxy_match
        src addr     : 172.16.1.0
        dst addr     : 172.16.3.0
        protocol     : 0
        src port     : 0
        dst port     : 0
*Jan 17 05:11:40.166: IPSEC(crypto_IPSec_sa_find_ident_head): reconnecting
with the same proxies and peer 192.168.23.3
*Jan 17 05:11:40.166: IPSEC(policy_db_add_ident): src 172.16.1.0, dest
172.16.3.0, dest_port 0

*Jan 17 05:11:40.166: IPSEC(create_sa): sa created,
  (sa) sa_dest= 192.168.12.1, sa_proto= 51,
    sa_spi= 0x588AA60C(1485481484),
    sa_trans= ah-sha-hmac , sa_conn_id= 2001
*Jan 17 05:11:40.166: IPSEC(create_sa): sa created,
  (sa) sa_dest= 192.168.23.3, sa_proto= 51,
    sa_spi= 0x897F9209(2306839049),
    sa_trans= ah-sha-hmac , sa_conn_id= 2002
*Jan 17 05:11:40.166: IPSEC(create_sa): sa created,
  (sa) sa_dest= 192.168.12.1, sa_proto= 50,
    sa_spi= 0x2E2954C0(774460608),
    sa_trans= esp-aes 256 esp-sha-hmac , sa_conn_id= 2001
*Jan 17 05:11:40.166: IPSEC(create_sa): sa created,
  (sa) sa_dest= 192.168.23.3, sa_proto= 50,
    sa_spi= 0xAE4C8E5A(2924252762),
    sa_trans= esp-aes 256 esp-sha-hmac , sa_conn_id= 2002
*Jan 17 05:11:40.166: IPSEC(update_current_outbound_sa): updated peer
192.168.23.3 current outbound sa to SPI AE4C8E5A
*Jan 17 05:12:30.162: ISAKMP:(1001):purging node 787769575
```

R1 has now established four security associations with R3: two in the outbound direction and two in the inbound direction. One in each direction is used for the AH transform and the other is used for the ESP transform. At this point, the ICMP replies are passing naturally through the IPsec tunnel.

Disable debugging once you are done.

```
R1# undebug all
All possible debugging has been turned off
```

As you can see, the output from the debugs is extensive and verbose. However, it can help you a lot if you are troubleshooting an IPsec problem. As mentioned earlier, we can view crypto commands and see that they now are populated with data.

```
R1# show crypto isakmp sa
dst             src             state          conn-id slot status
192.168.23.3    192.168.12.1    QM_IDLE           1001    0 ACTIVE
```

```
R1# show crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: MYMAP, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 192.168.12.1, remote crypto endpt.: 192.168.23.3
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
     current outbound spi: 0xAE4C8E5A(2924252762)

     inbound esp sas:
      spi: 0x2E2954C0(774460608)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: NETGX:1, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4506913/334)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:
      spi: 0x588AA60C(1485481484)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: NETGX:1, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4506913/332)
        replay detection support: Y
        Status: ACTIVE

     inbound pcp sas:

     outbound esp sas:
      spi: 0xAE4C8E5A(2924252762)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: NETGX:2, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4506913/332)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:
      spi: 0x897F9209(2306839049)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: NETGX:2, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4506913/332)
        replay detection support: Y
        Status: ACTIVE

     outbound pcp sas:
```

```
R3# show crypto isakmp sa
dst             src             state          conn-id slot status
192.168.23.3    192.168.12.1    QM_IDLE              1    0 ACTIVE

R3# show crypto ipsec sa

interface: Serial0/0/1
    Crypto map tag: MYMAP, local addr 192.168.23.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.3.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
   current_peer 192.168.12.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 192.168.23.3, remote crypto endpt.: 192.168.12.1
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/1
     current outbound spi: 0x2E2954C0(774460608)

     inbound esp sas:
      spi: 0xAE4C8E5A(2924252762)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3001, flow_id: NETGX:1, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4385199/319)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:
      spi: 0x897F9209(2306839049)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3001, flow_id: NETGX:1, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4385199/318)
        replay detection support: Y
        Status: ACTIVE

     inbound pcp sas:

     outbound esp sas:
      spi: 0x2E2954C0(774460608)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3002, flow_id: NETGX:2, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4385199/318)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:
      spi: 0x588AA60C(1485481484)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3002, flow_id: NETGX:2, crypto map: MYMAP
        sa timing: remaining key lifetime (k/sec): (4385199/318)
        replay detection support: Y
        Status: ACTIVE

     outbound pcp sas:
```

Why are there four security associations on each router?

Challenge: Use Wireshark to Monitor Encryption of Traffic

You can observe packets on the wire using Wireshark and see how their content looks unencrypted and then encrypted. To do this, first configure a SPAN session on the switch and open up Wireshark on a host attached to the SPAN destination port. You can use the host that you used for SDM because you don't need it anymore to configure the VPNs. If you do not know how to do this, refer to Lab 3.3: Configuring Wireshark and SPAN.

Next, you will remove the **crypto map** statements on R1 and R3. View the current configuration on the FastEthernet0/0 interface on R1 and Serial0/0/1 as shown below.

Then, issue the **no crypto map** *name* command in interface configuration mode to remove the ISAKMP security association. The router may issue a warning that ISAKMP is now off.

```
R1:

R1# show run interface fastethernet0/0
Building configuration...

Current configuration : 120 bytes
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 duplex auto
 speed auto
 crypto map SDM_CMAP_1
end

R1# configure terminal
R1(config)# interface fastethernet0/0
R1(config-if)# no crypto map SDM_CMAP_1
*Jan 16 06:02:58.999: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
```

```
R3:

R3# show run interface serial0/0/1
Building configuration...

Current configuration : 91 bytes
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map SDM_CMAP_1
end

R3# configure terminal
R3(config)# interface serial0/0/1
R3(config-if)# no crypto map SDM_CMAP_1
*Jan 16 06:05:36.038: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
```

You will attempt to sniff telnet traffic from R1 to R3. Enable telnet access on R3 and configure a secure password to get to configuration mode on R3.

```
R3(config)# enable secret cisco
R3(config)# line vty 0 4
R3(config-line)# password cisco
R3(config-line)# login
```

The routers have now been configured to allow telnet access.

Have Wireshark start sniffing the packets that it receives via the SPAN session.

Choose **Capture > Interfaces...**. Then click the **Start** button associated with the interface connected to the SPAN destination port. SPAN should start capturing packets on the line, so you can now telnet from R1's loopback to R3's loopback. To source telnet traffic, use the **telnet** *destination* **/source** *interface* command.

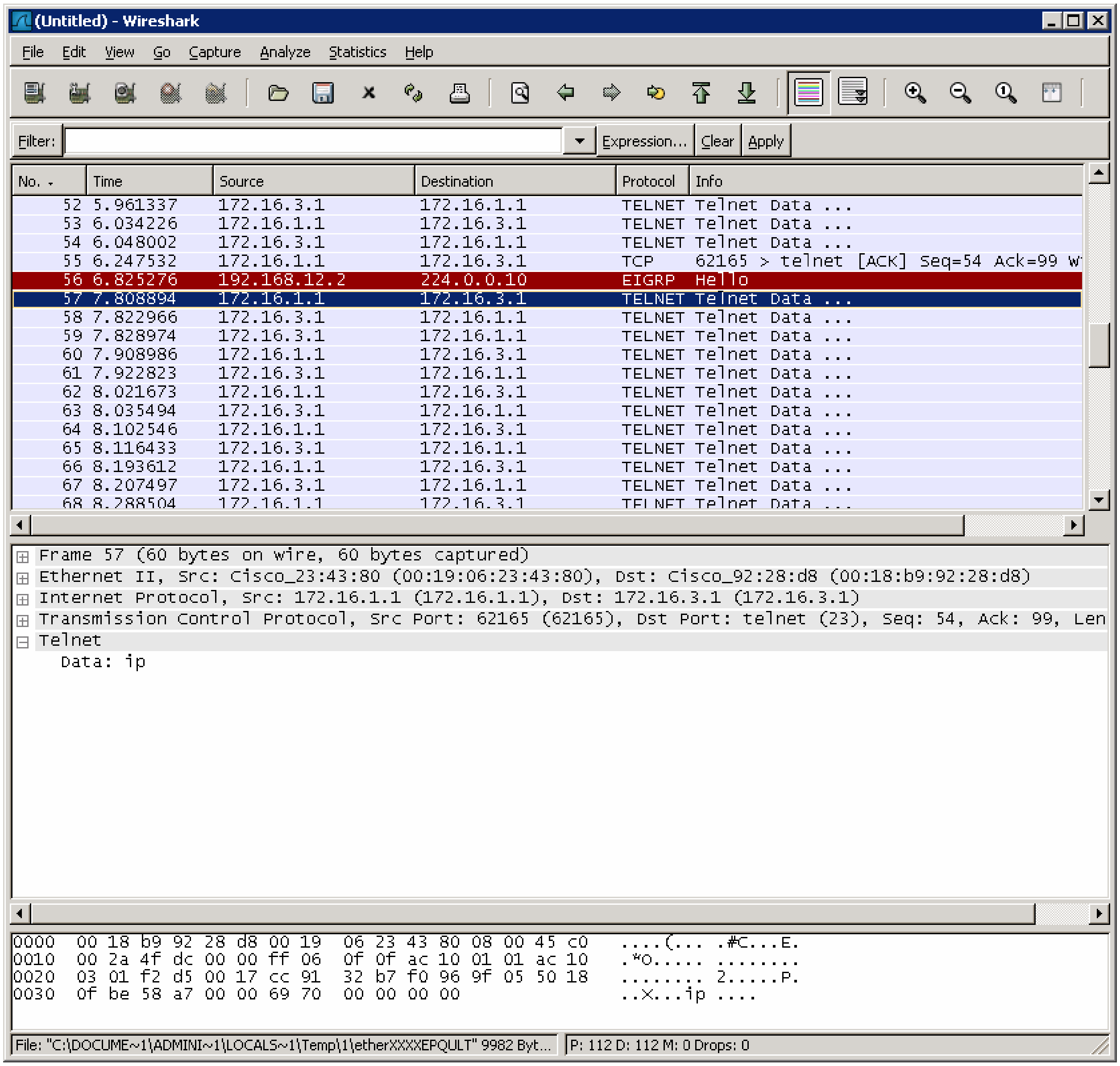
As shown in the previous step, you must source the telnet session from R1's loopback interface to simulate the interesting traffic that will match the VPN's access list.

First, begin capturing using Wireshark. Then, begin the telnet session. Once you are connected to R3, try issuing a command or two and then logging out.

```
R1# telnet 172.16.3.1 /source Loopback0
Trying 172.16.3.1 ... Open


User Access Verification

Password: [cisco]
R3> en
Password: [cisco]

R3# show ip interface brief
Interface           IP-Address      OK? Method Status                Protocol
FastEthernet0/0     unassigned      YES unset  administratively down down
FastEthernet0/1     unassigned      YES unset  administratively down down
Serial0/0/0         unassigned      YES unset  administratively down down
```

```
Serial0/0/1          192.168.23.3    YES manual up                    up
Serial0/1/0          unassigned      YES unset  administratively down down
Serial0/1/1          unassigned      YES unset  administratively down down
Loopback0            172.16.3.1      YES manual up                    up

R3# exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

Now, end the capture and look at the output. You will see a set of telnet data packets. Some of these, especially the return packets, will show whole unencrypted streams of text. The reason some return packets having longer text strings is because return packets can be streamed consecutively from the router managing the connection, whereas the text you type into telnet gets sent in chunks of characters or even character by character, depending on your typing speed.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

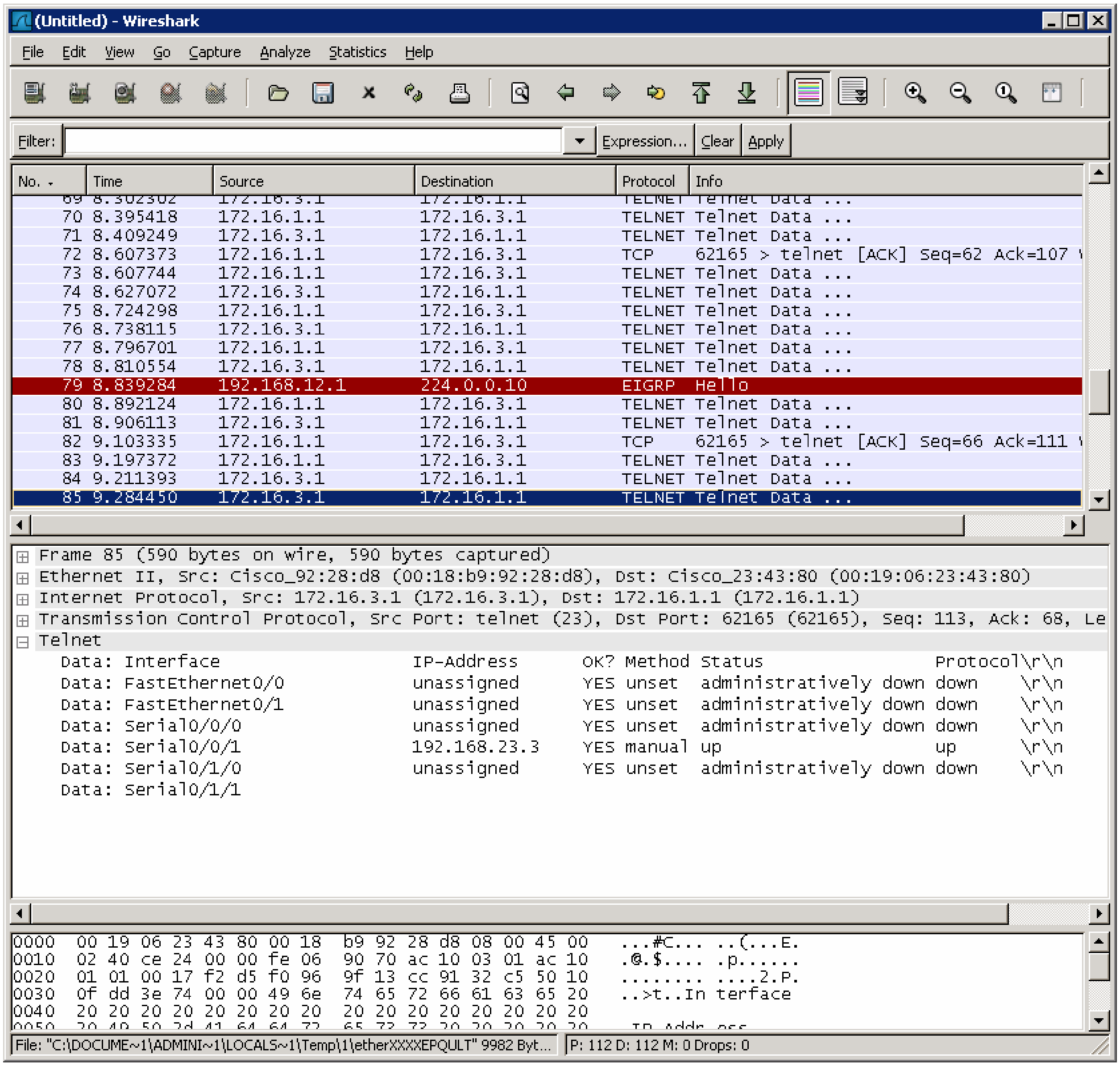
```
No. -  Time       Source          Destination     Protocol Info
    52 5.961337   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    53 6.034226   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
    54 6.048002   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    55 6.247532   172.16.1.1      172.16.3.1      TCP      62165 > telnet [ACK] Seq=54 Ack=99 W
    56 6.825276   192.168.12.2    224.0.0.10      EIGRP    Hello
    57 7.808894   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
    58 7.822966   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    59 7.828974   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    60 7.908986   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
    61 7.922823   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    62 8.021673   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
    63 8.035494   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    64 8.102546   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
    65 8.116433   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    66 8.193612   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
    67 8.207497   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
    68 8.288504   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
```

```
⊞ Frame 57 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: Cisco_23:43:80 (00:19:06:23:43:80), Dst: Cisco_92:28:d8 (00:18:b9:92:28:d8)
⊞ Internet Protocol, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.3.1 (172.16.3.1)
⊞ Transmission Control Protocol, Src Port: 62165 (62165), Dst Port: telnet (23), Seq: 54, Ack: 99, Len
⊟ Telnet
    Data: ip
```

```
0000  00 18 b9 92 28 d8 00 19  06 23 43 80 08 00 45 c0   ....(... .#C...E.
0010  00 2a 4f dc 00 00 ff 06  0f 0f ac 10 01 01 ac 10   .*O..... ........
0020  03 01 f2 d5 00 17 cc 91  32 b7 f0 96 9f 05 50 18   ........ 2.....P.
0030  0f be 58 a7 00 00 69 70  00 00 00 00               ..X...ip ....
```

```
File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\etherXXXXEPQULT" 9982 Byt... | P: 112 D: 112 M: 0 Drops: 0
```

Figure 11-1: Detailed Packet Data on Telnet String Sent From R1



```
No.  Time      Source          Destination     Protocol Info
 69 8.302302   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
 70 8.395418   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
 71 8.409249   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
 72 8.607373   172.16.1.1      172.16.3.1      TCP      62165 > telnet [ACK] Seq=62 Ack=107 W
 73 8.607744   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
 74 8.627072   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
 75 8.724298   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
 76 8.738115   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
 77 8.796701   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
 78 8.810554   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
 79 8.839284   192.168.12.1    224.0.0.10      EIGRP    Hello
 80 8.892124   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
 81 8.906113   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
 82 9.103335   172.16.1.1      172.16.3.1      TCP      62165 > telnet [ACK] Seq=66 Ack=111 W
 83 9.197372   172.16.1.1      172.16.3.1      TELNET   Telnet Data ...
 84 9.211393   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
 85 9.284450   172.16.3.1      172.16.1.1      TELNET   Telnet Data ...
```

```
⊞ Frame 85 (590 bytes on wire, 590 bytes captured)
⊞ Ethernet II, Src: Cisco_92:28:d8 (00:18:b9:92:28:d8), Dst: Cisco_23:43:80 (00:19:06:23:43:80)
⊞ Internet Protocol, Src: 172.16.3.1 (172.16.3.1), Dst: 172.16.1.1 (172.16.1.1)
⊞ Transmission Control Protocol, Src Port: telnet (23), Dst Port: 62165 (62165), Seq: 113, Ack: 68, Le
⊟ Telnet
    Data: Interface                  IP-Address      OK? Method Status                Protocol\r\n
    Data: FastEthernet0/0            unassigned      YES unset  administratively down down     \r\n
    Data: FastEthernet0/1            unassigned      YES unset  administratively down down     \r\n
    Data: Serial0/0/0                unassigned      YES unset  administratively down down     \r\n
    Data: Serial0/0/1                192.168.23.3    YES manual up                    up       \r\n
    Data: Serial0/1/0                unassigned      YES unset  administratively down down     \r\n
    Data: Serial0/1/1
```

```
0000  00 19 06 23 43 80 00 18  b9 92 28 d8 08 00 45 00   ...#C... ..(...E.
0010  02 40 ce 24 00 00 fe 06  90 70 ac 10 03 01 ac 10   .@.$.... .p......
0020  01 01 00 17 f2 d5 f0 96  9f 13 cc 91 32 c5 50 10   ........ ....2.P.
0030  0f dd 3e 74 00 00 49 6e  74 65 72 66 61 63 65 20   ..>t..In terface
0040  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
0050  20 49 50 2d 41 64 64 72  65 73 73 20 20 20 20 20    IP-Addr ess
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\etherXXXXEPQULT" 9982 Byt... | P: 112 D: 112 M: 0 Drops: 0

Figure 11-2: Detailed Packet Data on Return Telnet Traffic from R3

Based on this output, you can see how easy it is for someone who is in the path of sensitive data to view unencrypted or clear text traffic.

Now, you will reapply the cryptography settings on R1 and R3 and begin a telnet session from R1 to R3 as before.

Begin by reapplying the crypto maps you removed earlier on R1 and R3.

```
R1(config)# interface fastethernet0/0
R1(config-if)# crypto map SDM_CMAP_1
*Jan 16 06:36:10.295: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON

R3(config)# interface serial0/0/1
R3(config-if)# crypto map SDM_CMAP_1
```

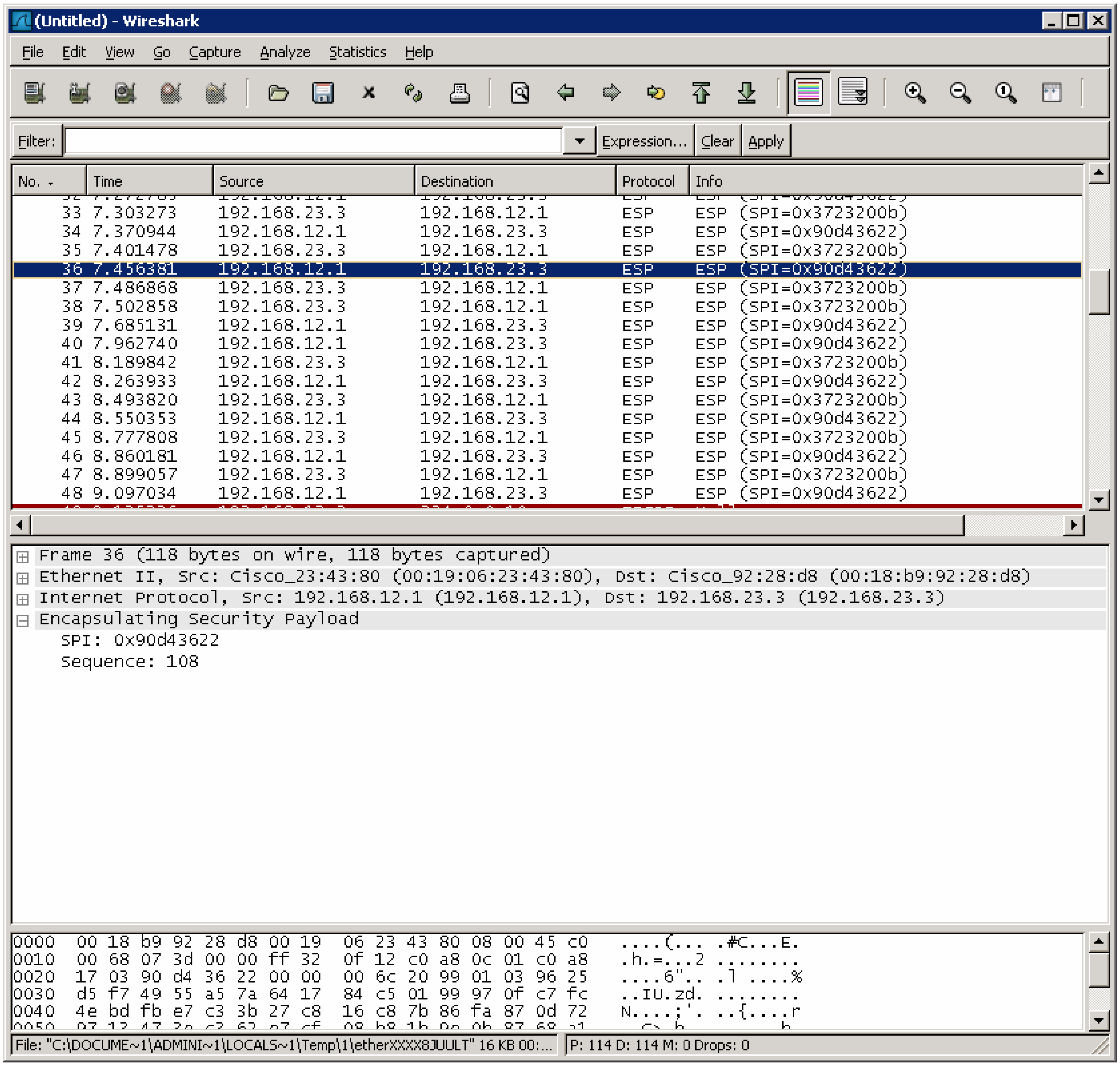
```
*Jan 16 06:37:59.798: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

Start the packet capturing again in Wireshark, and then issue the same telnet sequence that you did previously.

```
R1# telnet 172.16.3.1 /source Loopback0
Trying 172.16.3.1 ... Open


User Access Verification

Password: [cisco]
R3> en
Password: [cisco]

R3# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES unset  administratively down down
FastEthernet0/1        unassigned      YES unset  administratively down down
Serial0/0/0            unassigned      YES unset  administratively down down
Serial0/0/1            192.168.23.3    YES manual up                    up
Serial0/1/0            unassigned      YES unset  administratively down down
Serial0/1/1            unassigned      YES unset  administratively down down
Loopback0              172.16.3.1      YES manual up                    up

R3# exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

End your Wireshark capture when you are finished with the telnet session.

As far as the user is concerned, the telnet session seems the same with and without encryption. However, the packet capture from Wireshark shows that the VPN is actively encapsulating and encrypting packets.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No.  Time      Source          Destination     Protocol Info
  33 7.303273  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  34 7.370944  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
  35 7.401478  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  36 7.456381  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
  37 7.486868  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  38 7.502858  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  39 7.685131  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
  40 7.962740  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
  41 8.189842  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  42 8.263933  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
  43 8.493820  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  44 8.550353  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
  45 8.777808  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  46 8.860181  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
  47 8.899057  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x3723200b)
  48 9.097034  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x90d43622)
```

```
⊞ Frame 36 (118 bytes on wire, 118 bytes captured)
⊞ Ethernet II, Src: Cisco_23:43:80 (00:19:06:23:43:80), Dst: Cisco_92:28:d8 (00:18:b9:92:28:d8)
⊞ Internet Protocol, Src: 192.168.12.1 (192.168.12.1), Dst: 192.168.23.3 (192.168.23.3)
⊟ Encapsulating Security Payload
    SPI: 0x90d43622
    Sequence: 108
```

```
0000  00 18 b9 92 28 d8 00 19  06 23 43 80 08 00 45 c0   ....(... .#C...E.
0010  00 68 07 3d 00 00 ff 32  0f 12 c0 a8 0c 01 c0 a8   .h.=...2 ........
0020  17 03 90 d4 36 22 00 00  00 6c 20 99 01 03 96 25   ....6".. .l ....%
0030  d5 f7 49 55 a5 7a 64 17  84 c5 01 99 97 0f c7 fc   ..IU.zd. ........
0040  4e bd fb e7 c3 3b 27 c8  16 c8 7b 86 fa 87 0d 72   N....;'. ..{....r
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\etherXXXX8JUULT" 16 KB 00:... P: 114 D: 114 M: 0 Drops: 0

Figure 11-3: Detailed Packet Data on Encrypted Telnet String Sent From R1

Notice that the protocol is not telnet (TCP port 23), but the Encapsulating Security Protocol (ESP, IP protocol number 50). Remember, all traffic here matches the IPsec access list.

Also, notice that the source and destination are not the actual source and destination of the addresses participating in this telnet conversation. Rather, they are the endpoints of the VPN.

Why do you use the VPN endpoints as the source and destination of packets?

Finally, and most important, if you look at the contents of these packets in Wireshark, no matter how you try to format or filter them, you will not be able to see what data was originally inside.

The encryption suite provided by IPsec successfully secures data through authentication, encryption, and data-integrity services.

Appendix A: TCL Script Output

```
R1# tclsh
R1(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
R1(tcl)# tclquit

R2# tclsh
R2(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/68 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
R2(tcl)# tclquit

R3# tclsh
R3(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/64 ms
R3(tcl)# tclquit
```

Final Configurations:

```
R1# show run
!
hostname R1
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.23.3
!
crypto ipsec security-association lifetime seconds 1800
!
crypto ipsec transform-set 50 ah-sha-hmac esp-aes 256 esp-sha-hmac
!
crypto map MYMAP 10 ipsec-isakmp
 set peer 192.168.23.3
 set security-association lifetime seconds 900
 set transform-set 50
 set pfs group5
 match address 101
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 crypto map MYMAP
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 network 192.168.12.0
 no auto-summary
!
access-list 101 permit ip 172.16.1.0 0.0.0.255 172.16.3.0 0.0.0.255
!
end

R2# show run
!
hostname R2
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 192.168.12.0
 network 192.168.23.0
 no auto-summary
!
end

R3# show run
```

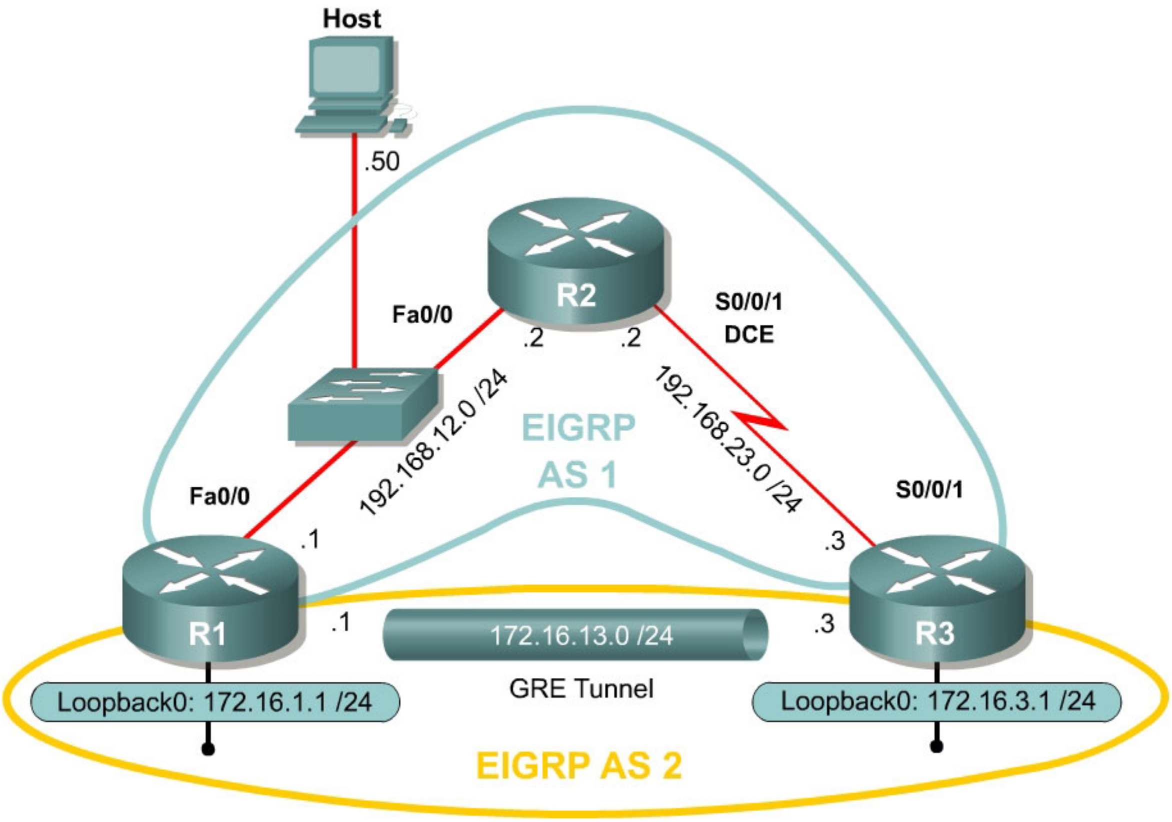
```
!hostname R3
!
enable secret 5 $1$LT7i$MY2NhpGjl5uL1zNAoR2tf.
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.12.1
!
crypto ipsec security-association lifetime seconds 1800
!
crypto ipsec transform-set 50 ah-sha-hmac esp-aes 256 esp-sha-hmac
!
crypto map MYMAP 10 ipsec-isakmp
 set peer 192.168.12.1
 set security-association lifetime seconds 900
 set transform-set 50
 set pfs group5
 match address 101
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map MYMAP
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 network 192.168.23.0
 no auto-summary
!
access-list 101 permit ip 172.16.3.0 0.0.0.255 172.16.1.0 0.0.0.255
!
line vty 0 4
 password cisco
 login
!
end
```

Lab 3.6 Configuring a Secure GRE Tunnel with SDM

Learning Objectives

- Configure EIGRP on the routers
- Use SDM to configure a secure GRE tunnel

Topology Diagram



Scenario

In this lab, you will use the Cisco Security Device Manager (SDM) to configure a secure generic routing encapsulation (GRE) tunnel using IPsec. It will help if you have previously completed Labs 3.2, 3.4, and 3.5 since this lab will build on concepts covered in those labs.

Step 1: Configure Addressing

Configure the loopback interfaces with the addresses shown in the diagram. Configure the other interfaces as depicted in the topology above Do not forget to set the clockrates on the appropriate interfaces and issue the **no shutdown** command on all serial connections, as necessary. Verify that you have connectivity across the local subnet using the **ping** command. Do not set up the tunnel interface until the next step.

```
R1# configure terminal
R1(config)# interface loopback 0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface fastethernet 0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown

R2# configure terminal
R2(config)# interface fastethernet 0/0
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/0/1
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3# configure terminal
R3(config)# interface loopback 0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface serial0/0/1
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Configure EIGRP AS 1

Configure EIGRP AS 1 for the major networks 192.168.12.0/24 and 192.168.23.0/24. Do not include the networks in the diagram falling in the 172.16.0.0/16 range. The Class C networks will serve as the transit networks for the tunnel network. Make sure you disable EIGRP automatic summarization.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 192.168.12.0

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 192.168.12.0
R2(config-router)# network 192.168.23.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
R3(config-router)# network 192.168.23.0
```

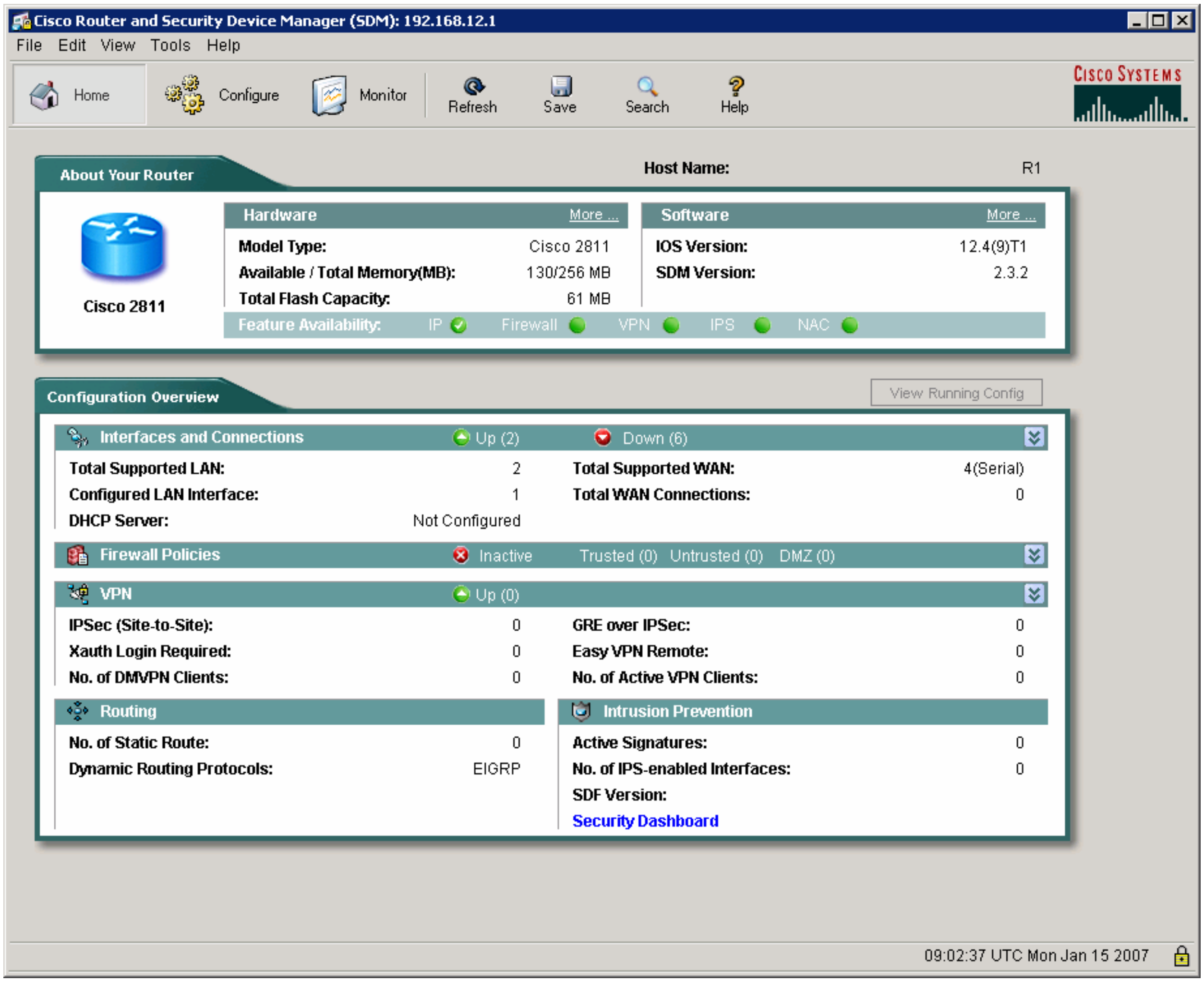
Given the above configuration, will the 172.16.1.0/24 network be reachable from R3? Explain.

Will the 172.16.3.0/24 network be reachable from R1?

Step 3: Connect to the Router using SDM

Prepare R1 for access using SDM as described in Lab 3.1: Configuring SDM on a Router.

Configure the IP address shown in the diagram on the host PC and install SDM to either the router or the PC. Connect to the router using SDM so that you are at the SDM home screen. For information on how to configure SDM, refer to the configuring SDM lab.



Host Name: R1

Cisco 2811

Hardware	More ...
Model Type:	Cisco 2811
Available / Total Memory(MB):	130/256 MB
Total Flash Capacity:	61 MB

Software	More ...
IOS Version:	12.4(9)T1
SDM Version:	2.3.2

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Interfaces and Connections ● Up (2) ● Down (6)

Total Supported LAN:	2	Total Supported WAN:	4(Serial)
Configured LAN Interface:	1	Total WAN Connections:	0
DHCP Server:	Not Configured		

Firewall Policies ☒ Inactive Trusted (0) Untrusted (0) DMZ (0)

VPN ● Up (0)

IPSec (Site-to-Site):	0	GRE over IPSec:	0
Xauth Login Required:	0	Easy VPN Remote:	0
No. of DMVPN Clients:	0	No. of Active VPN Clients:	0

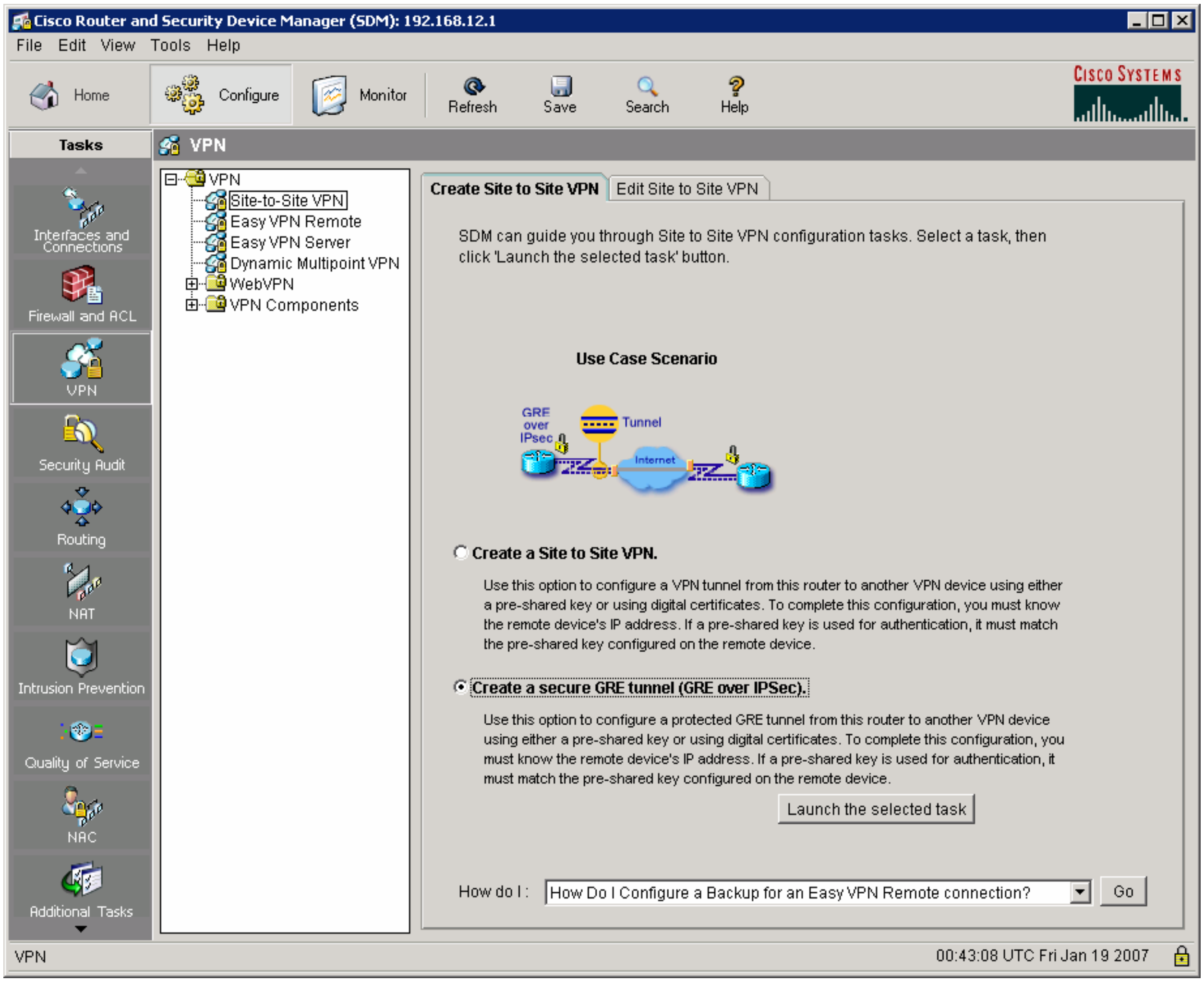
Routing		Intrusion Prevention	
No. of Static Route:	0	Active Signatures:	0
Dynamic Routing Protocols:	EIGRP	No. of IPS-enabled Interfaces:	0
		SDF Version:	
		Security Dashboard	

09:02:37 UTC Mon Jan 15 2007

Figure 3-1: SDM Home Screen

Step 4: Configure an IPsec VTI using SDM

SDM contains a wizard that makes configuring an IPsec virtual tunnel interface (VTI) very simple. Click the **Configure** tab at the top, and then choose **VPN** on the left side bar. In the second column from the left, click **Site-to-Site VPN**, and then in the **Create Site to Site VPN** tab choose **Create a secure GRE tunnel (GRE over IPsec)**. Click the **Launch the selected task** button.



SDM can guide you through Site to Site VPN configuration tasks. Select a task, then click 'Launch the selected task' button.

○ **Create a Site to Site VPN.**

Use this option to configure a VPN tunnel from this router to another VPN device using either a pre-shared key or using digital certificates. To complete this configuration, you must know the remote device's IP address. If a pre-shared key is used for authentication, it must match the pre-shared key configured on the remote device.

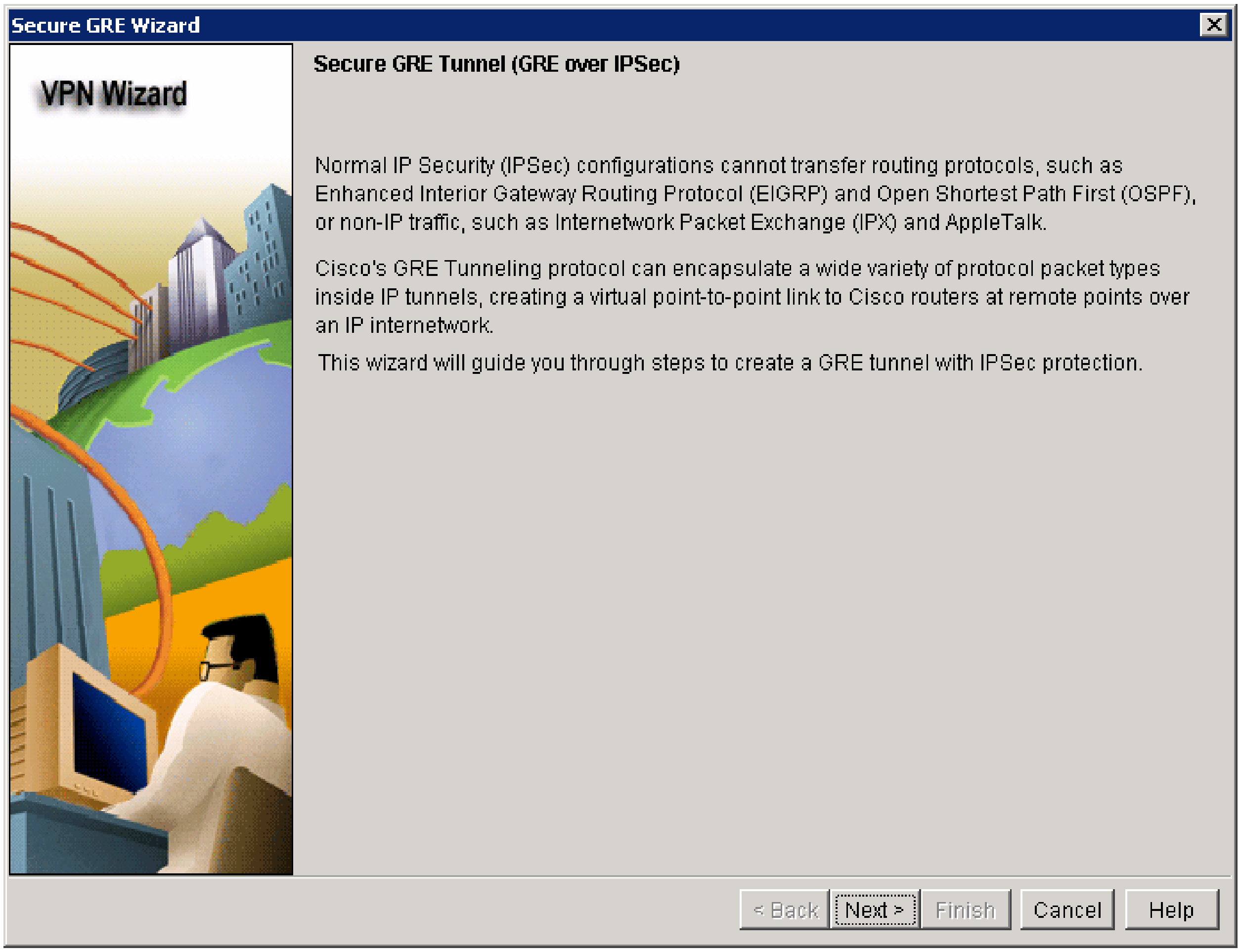
● **Create a secure GRE tunnel (GRE over IPSec).**

Use this option to configure a protected GRE tunnel from this router to another VPN device using either a pre-shared key or using digital certificates. To complete this configuration, you must know the remote device's IP address. If a pre-shared key is used for authentication, it must match the pre-shared key configured on the remote device.

How do I : How Do I Configure a Backup for an Easy VPN Remote connection?

Figure 4-1: Site-to-Site VPN Tab

After reading the brief introduction to IPsec VTIs, click **Next** to start the wizard.



Secure GRE Tunnel (GRE over IPSec)

Normal IP Security (IPSec) configurations cannot transfer routing protocols, such as Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF), or non-IP traffic, such as Internetwork Packet Exchange (IPX) and AppleTalk.

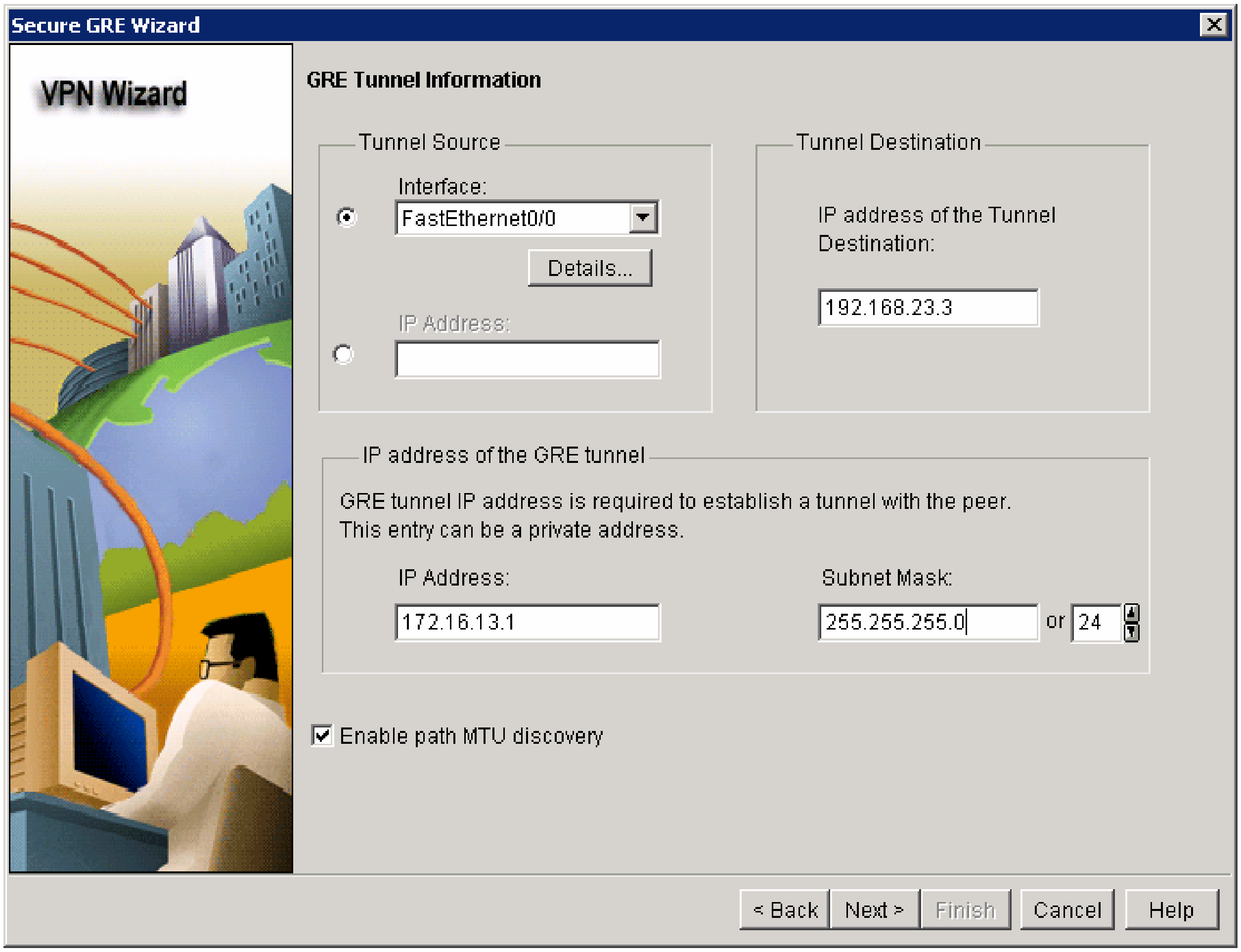
Cisco's GRE Tunneling protocol can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork.

This wizard will guide you through steps to create a GRE tunnel with IPSec protection.

Figure 4-2: Secure GRE Wizard

What IP addresses should you use as the endpoints for your GRE tunnel? Why?

Configure the tunnel source using the FastEthernet0/0 interface on R1. Choose the IP address destination using the closest interface on R3 to R1. The internal IP address and subnet mask of the tunnel are given in the diagram on page 1 of this lab.



GRE Tunnel Information

Tunnel Source

● Interface: FastEthernet0/0

Details...

○ IP Address:

Tunnel Destination

IP address of the Tunnel Destination: 192.168.23.3

IP address of the GRE tunnel

GRE tunnel IP address is required to establish a tunnel with the peer. This entry can be a private address.

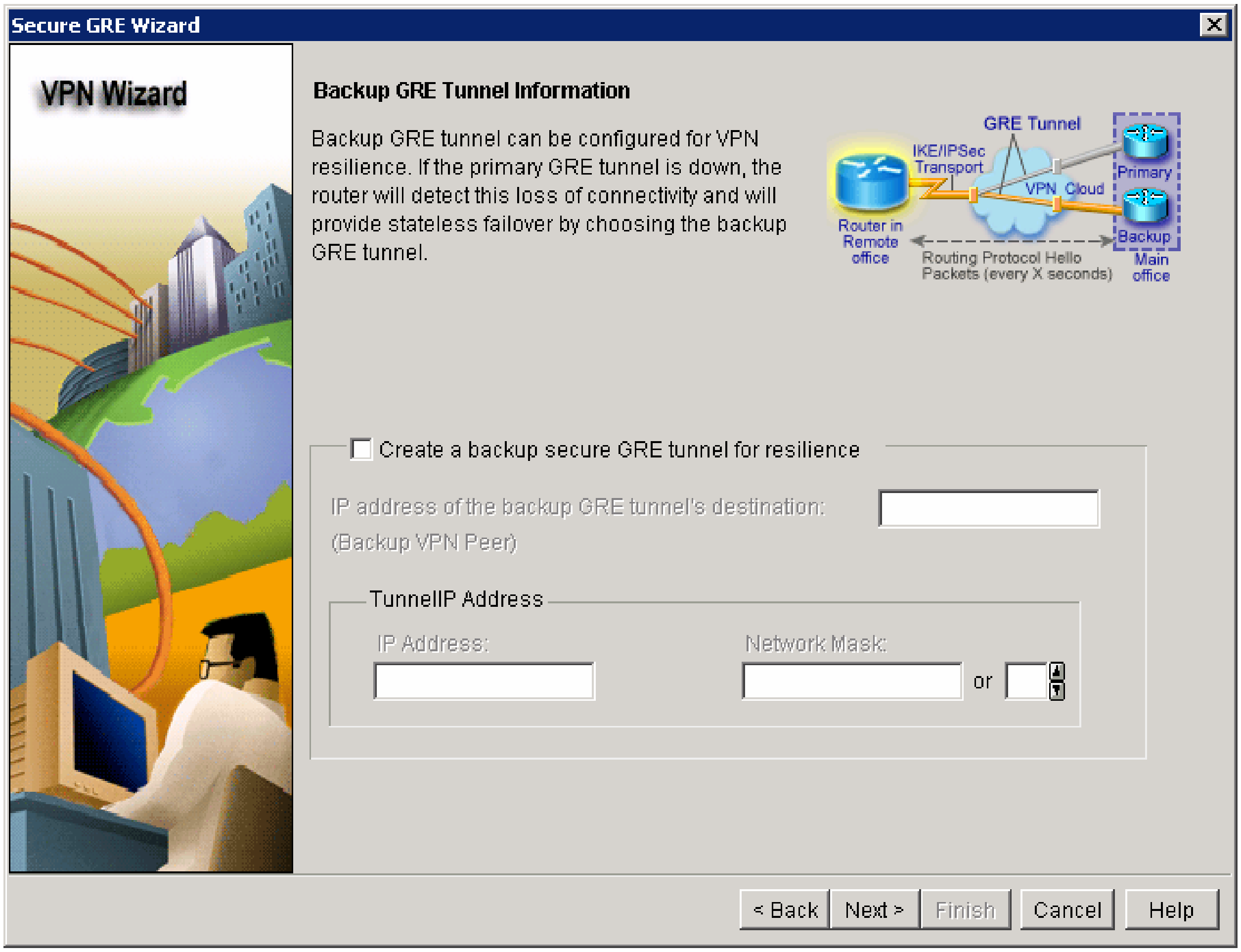
IP Address: 172.16.13.1

Subnet Mask: 255.255.255.0 or 24

☑ Enable path MTU discovery

Figure 4-3: GRE Tunnel Configuration

At the next prompt in the wizard, do not check **Create a backup GRE tunnel for resilience**. Just click the **Next** button.



Secure GRE Wizard

VPN Wizard

Backup GRE Tunnel Information

Backup GRE tunnel can be configured for VPN resilience. If the primary GRE tunnel is down, the router will detect this loss of connectivity and will provide stateless failover by choosing the backup GRE tunnel.

□ Create a backup secure GRE tunnel for resilience

IP address of the backup GRE tunnel's destination: (Backup VPN Peer)

TunnelIP Address

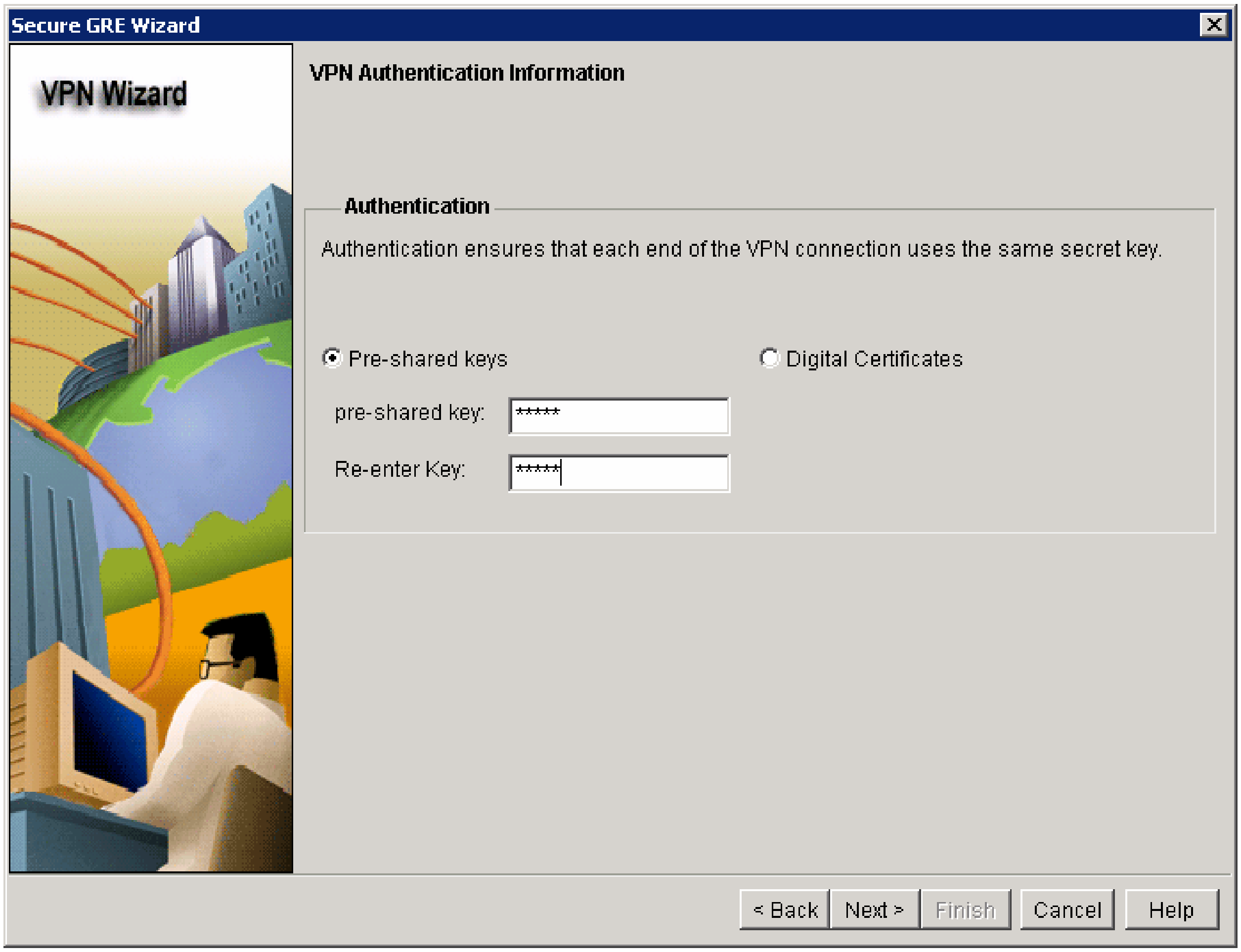
IP Address:

Network Mask: or

Figure 4-4: Backup GRE Tunnel Options

Click **Pre-shared keys** for the authentication method and use “cisco” as your pre-shared key.

What is a pre-shared key and what purpose does it serve?



VPN Authentication Information

Authentication

Authentication ensures that each end of the VPN connection uses the same secret key.

● Pre-shared keys ○ Digital Certificates

pre-shared key: *****

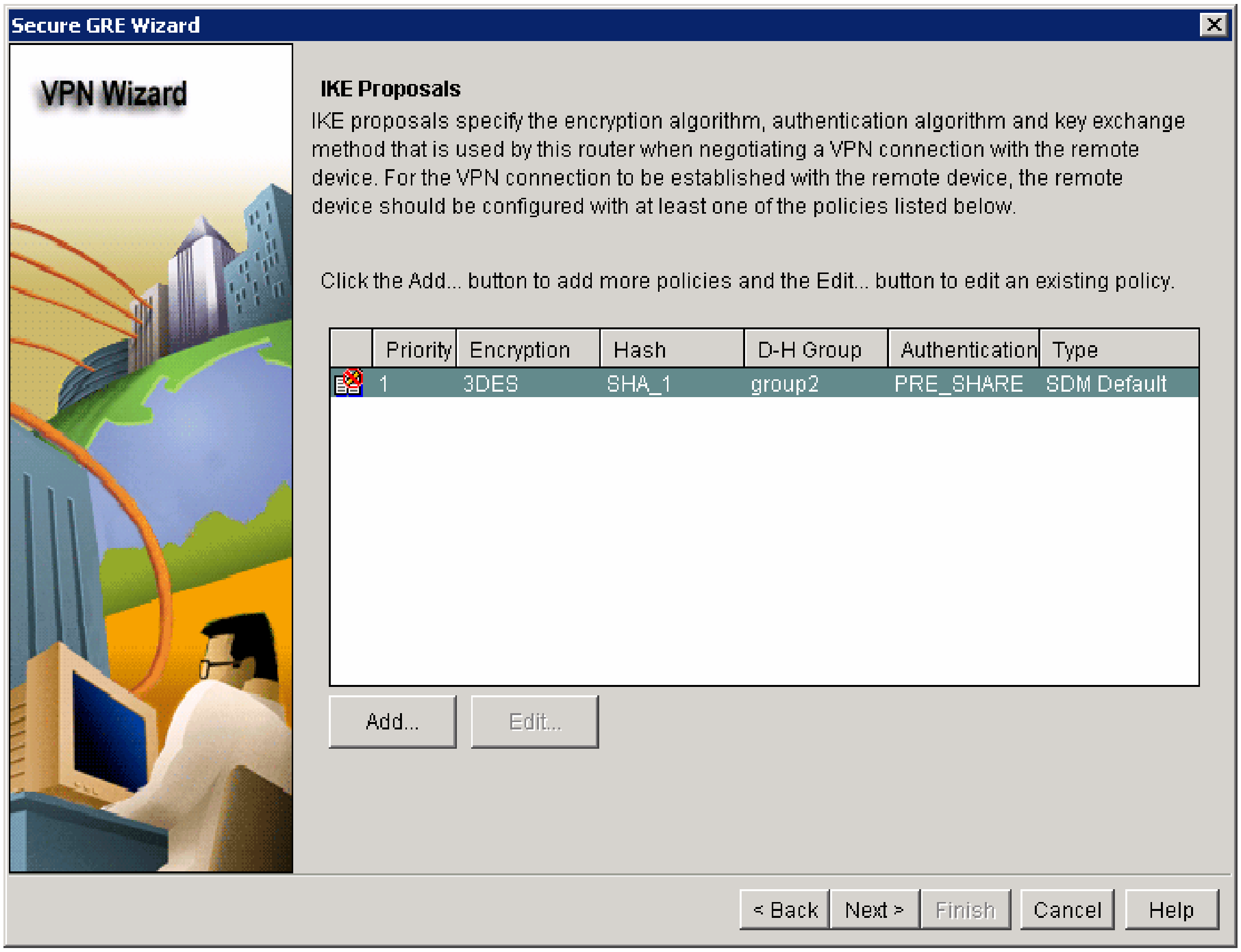
Re-enter Key: *****

Figure 4-5: VPN Authentication Information

Based on your work on the IPsec VPN labs, what is the function of the Internet Key Exchange (IKE) protocol?

What attributes may be configured in an IKE policy? Describe at least three attributes.

Create a new IKE policy by clicking the **Add...** button.



IKE Proposals

IKE proposals specify the encryption algorithm, authentication algorithm and key exchange method that is used by this router when negotiating a VPN connection with the remote device. For the VPN connection to be established with the remote device, the remote device should be configured with at least one of the policies listed below.

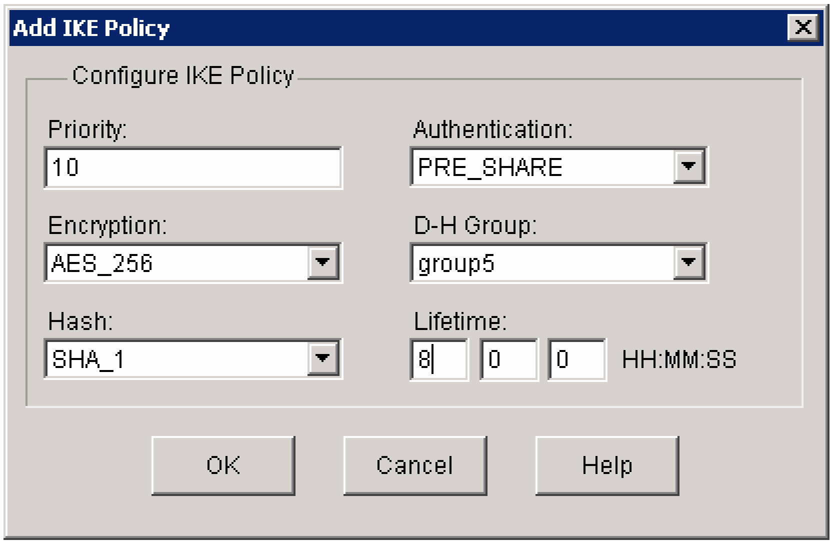
Click the Add... button to add more policies and the Edit... button to edit an existing policy.

	Priority	Encryption	Hash	D-H Group	Authentication	Type
	1	3DES	SHA_1	group2	PRE_SHARE	SDM Default

Figure 4-6: IKE Proposals List

Create the new policy using the settings shown in figure 4-7. If your IOS image doesn’t support all of the settings, configure what you can. Just make sure your VPN settings match on both ends of the connection.

Then click **OK**.



Configure IKE Policy

Priority: 10

Authentication: PRE_SHARE

Encryption: AES_256

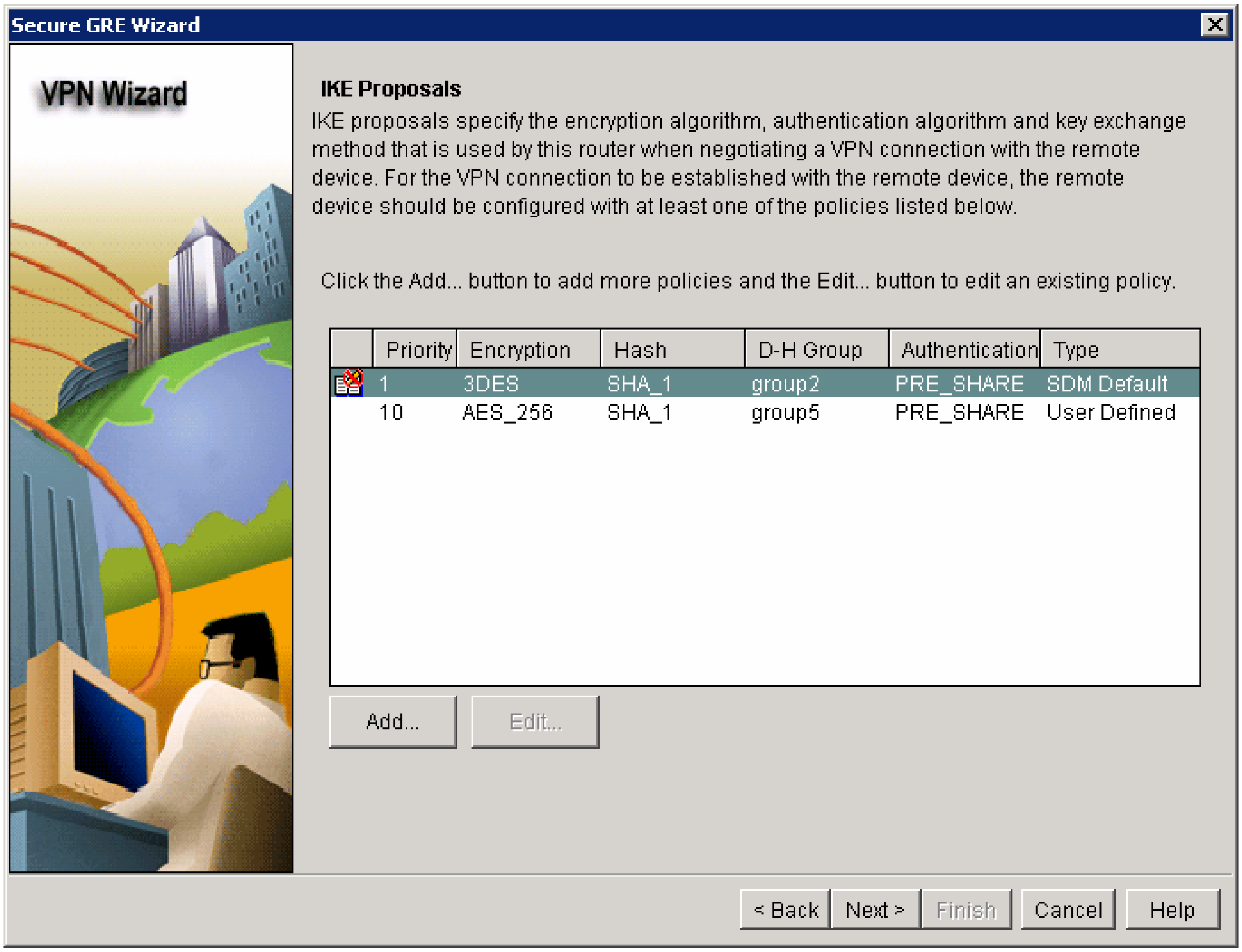
D-H Group: group5

Hash: SHA_1

Lifetime: 8 0 0 HH:MM:SS

Figure 4-7: Add IKE Policy Dialog

You should now see your new IKE proposal in the list. Click **Next** to continue.



IKE Proposals

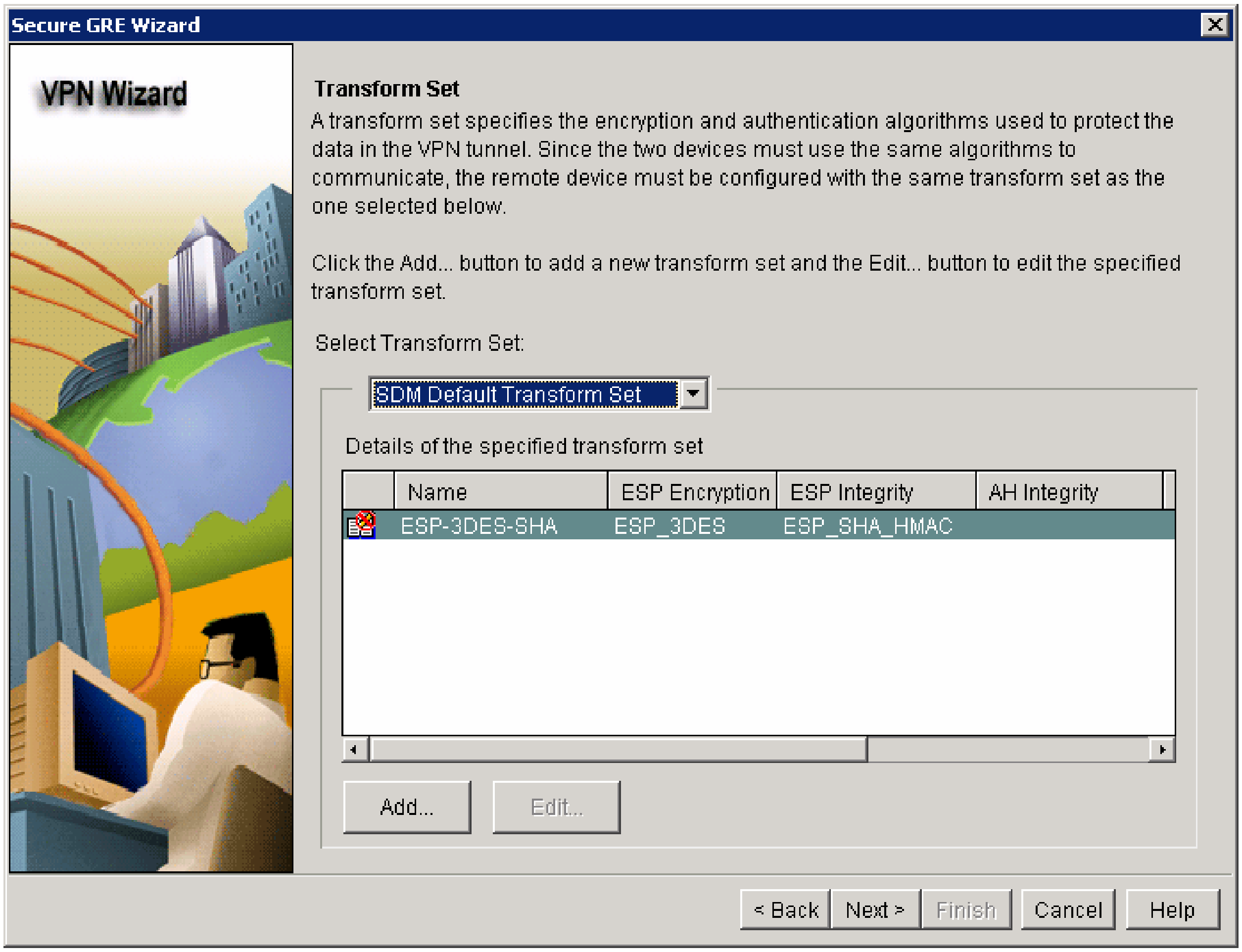
IKE proposals specify the encryption algorithm, authentication algorithm and key exchange method that is used by this router when negotiating a VPN connection with the remote device. For the VPN connection to be established with the remote device, the remote device should be configured with at least one of the policies listed below.

Click the Add... button to add more policies and the Edit... button to edit an existing policy.

	Priority	Encryption	Hash	D-H Group	Authentication	Type
	1	3DES	SHA_1	group2	PRE_SHARE	SDM Default
	10	AES_256	SHA_1	group5	PRE_SHARE	User Defined

Figure 4-8: IKE Proposals with Changes Applied

On the **Transform Set** page, create a new transform set by clicking **Add...**.



Transform Set

A transform set specifies the encryption and authentication algorithms used to protect the data in the VPN tunnel. Since the two devices must use the same algorithms to communicate, the remote device must be configured with the same transform set as the one selected below.

Click the Add... button to add a new transform set and the Edit... button to edit the specified transform set.

Select Transform Set: SDM Default Transform Set

Details of the specified transform set

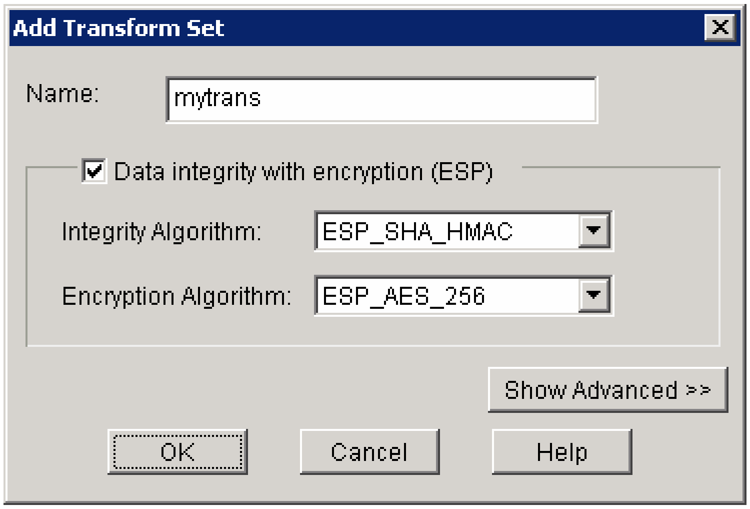
Name	ESP Encryption	ESP Integrity	AH Integrity
ESP-3DES-SHA	ESP_3DES	ESP_SHA_HMAC	

Figure 4-9: IPsec Transform Set List

What is the function of an IPsec transform set?

What are the main differences between the authentication header (AH) and the encapsulated security payload (ESP) as methods to ensure data integrity?

Create a new transform set using the name “mytrans.” Use the settings shown in the following screenshot. If these settings are not supported on your router, use whichever settings you can. However, remember to keep the settings consistent on both sides of the tunnel.



Add Transform Set

Name: mytrans

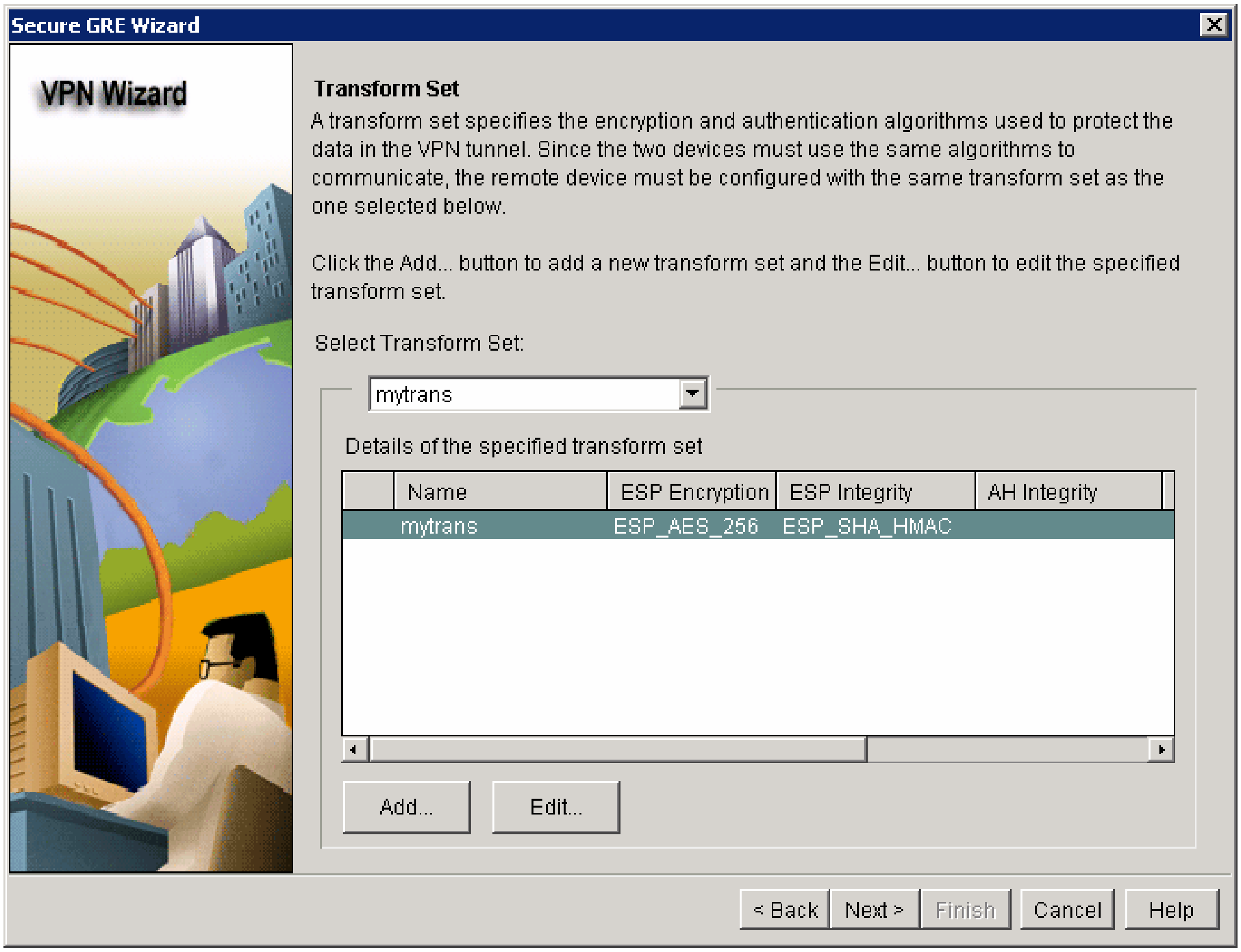
☑ Data integrity with encryption (ESP)

Integrity Algorithm: ESP_SHA_HMAC

Encryption Algorithm: ESP_AES_256

Figure 4-10: Add IPsec Transform Set Dialog

Click **OK** to continue. You should see your new transform set appear in the window. Click **Next**.



Secure GRE Wizard

Transform Set

A transform set specifies the encryption and authentication algorithms used to protect the data in the VPN tunnel. Since the two devices must use the same algorithms to communicate, the remote device must be configured with the same transform set as the one selected below.

Click the Add... button to add a new transform set and the Edit... button to edit the specified transform set.

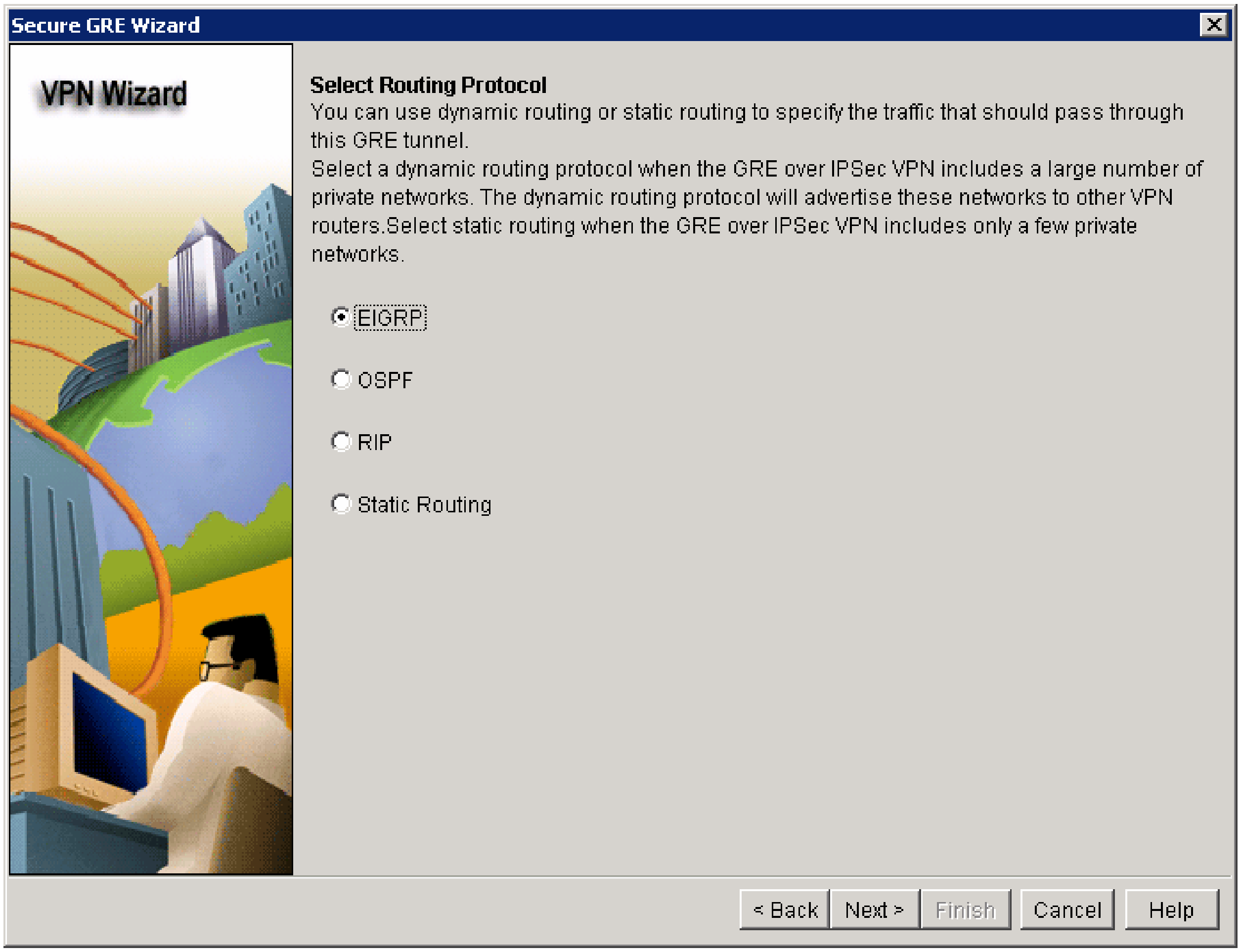
Select Transform Set: mytrans

Details of the specified transform set

Name	ESP Encryption	ESP Integrity	AH Integrity
mytrans	ESP_AES_256	ESP_SHA_HMAC	

Figure 4-11: IPsec Transform Set List with Changes Applies

Choose **EIGRP** as the routing protocol and click **Next**.



Select Routing Protocol

You can use dynamic routing or static routing to specify the traffic that should pass through this GRE tunnel.

Select a dynamic routing protocol when the GRE over IPSec VPN includes a large number of private networks. The dynamic routing protocol will advertise these networks to other VPN routers.Select static routing when the GRE over IPSec VPN includes only a few private networks.

● EIGRP

○ OSPF

○ RIP

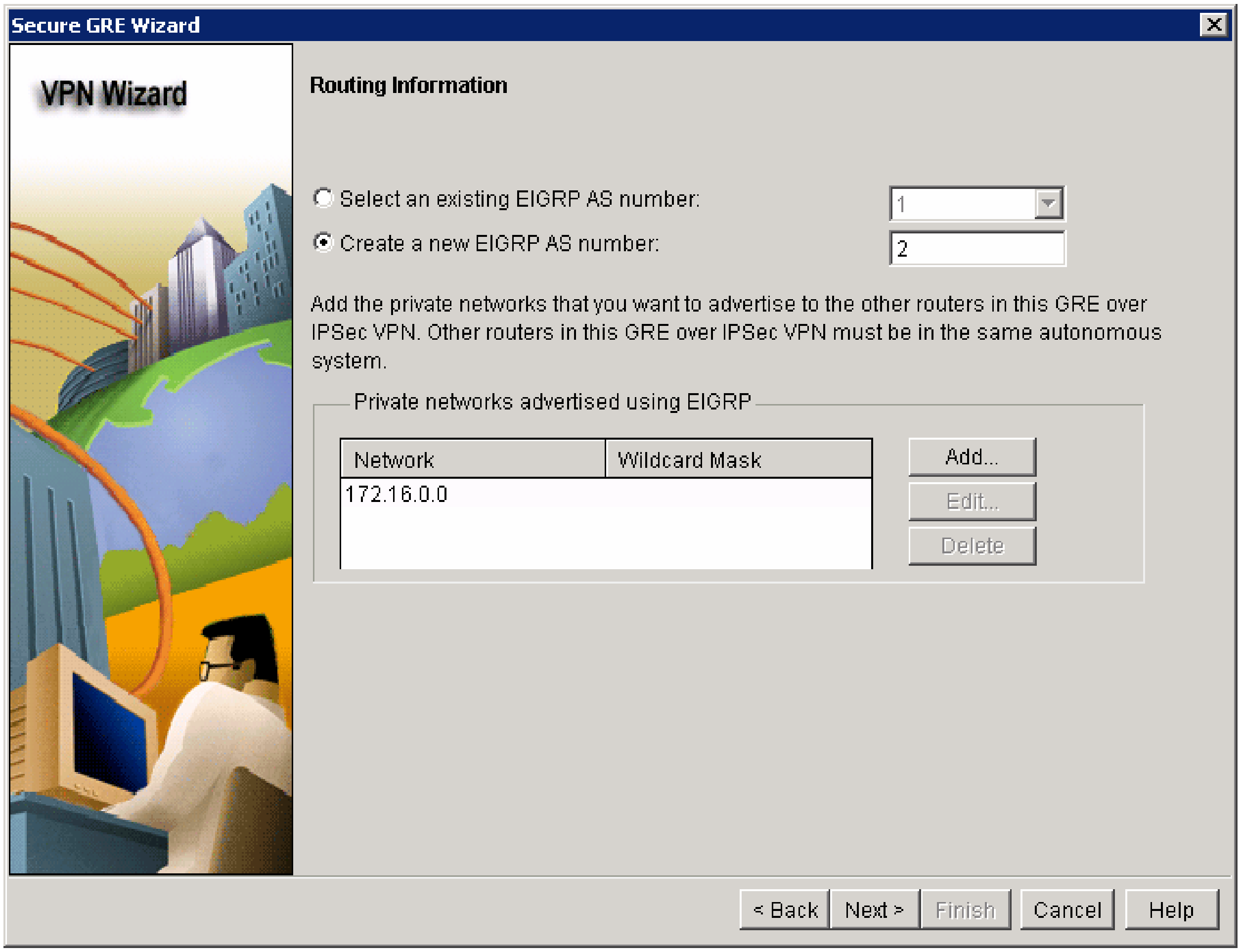
○ Static Routing

Figure 4-12: Routing Protocol Selection

Choose **Create a New EIGRP AS Number** and use EIGRP AS number 2, as shown in the diagram on page 1 of this lab, to route over the tunnel.

Will EIGRP AS 1 and EIGRP AS 2 automatically redistribute routes between autonomous systems?

Add the entire 172.16.0.0 major network into this EIGRP autonomous system on R1.



Secure GRE Wizard

Routing Information

○ Select an existing EIGRP AS number: 1

● Create a new EIGRP AS number: 2

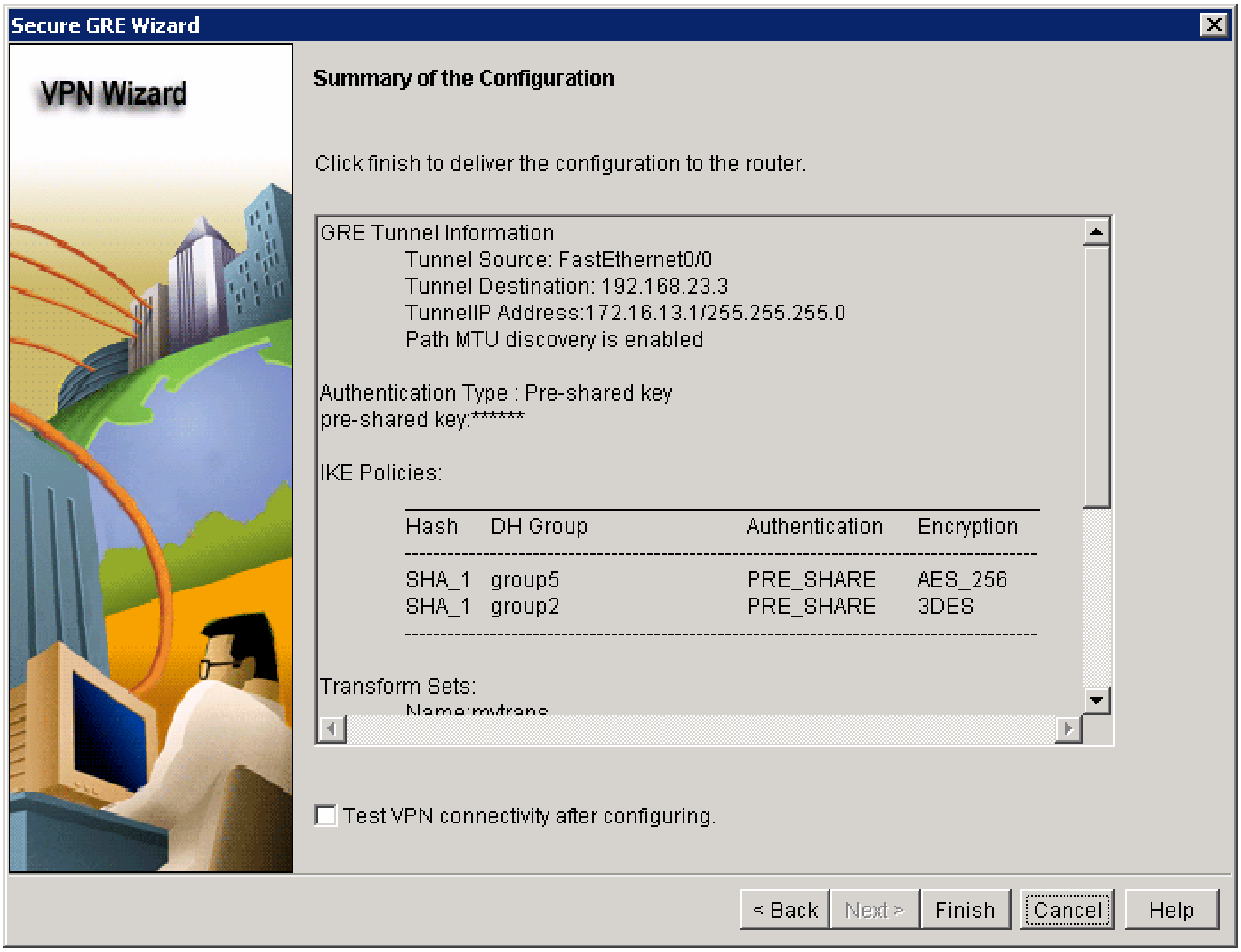
Add the private networks that you want to advertise to the other routers in this GRE over IPSec VPN. Other routers in this GRE over IPSec VPN must be in the same autonomous system.

Private networks advertised using EIGRP

Network	Wildcard Mask
172.16.0.0	

Figure 4-13: Advanced Routing Protocol Configuration

A screen will pop up to confirm the configuration that will be delivered to the router. Click **Finish** to deliver the configuration to the router. Do not test VPN connectivity yet, because the other endpoint of the tunnel is not configured.



Summary of the Configuration

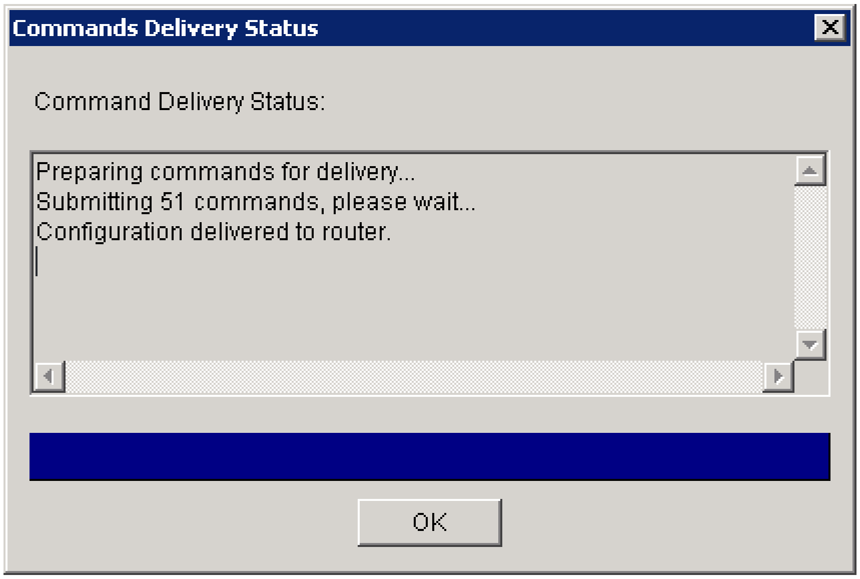
Click finish to deliver the configuration to the router.

```
GRE Tunnel Information
          Tunnel Source: FastEthernet0/0
          Tunnel Destination: 192.168.23.3
          TunnelIP Address:172.16.13.1/255.255.255.0
          Path MTU discovery is enabled

Authentication Type : Pre-shared key
pre-shared key:******

IKE Policies:

          Hash    DH Group                      Authentication   Encryption
          ------------------------------------------------------------------
          SHA_1   group5                        PRE_SHARE        AES_256
          SHA_1   group2                        PRE_SHARE        3DES
          ------------------------------------------------------------------

Transform Sets:
          Name:mytrans
```

□ Test VPN connectivity after configuring.

Figure 4-14: Site-to-Site IPsec GRE Configuration Summary

SDM will deliver the configuration changes to the router. When the configuration changes are completed, click **OK**.



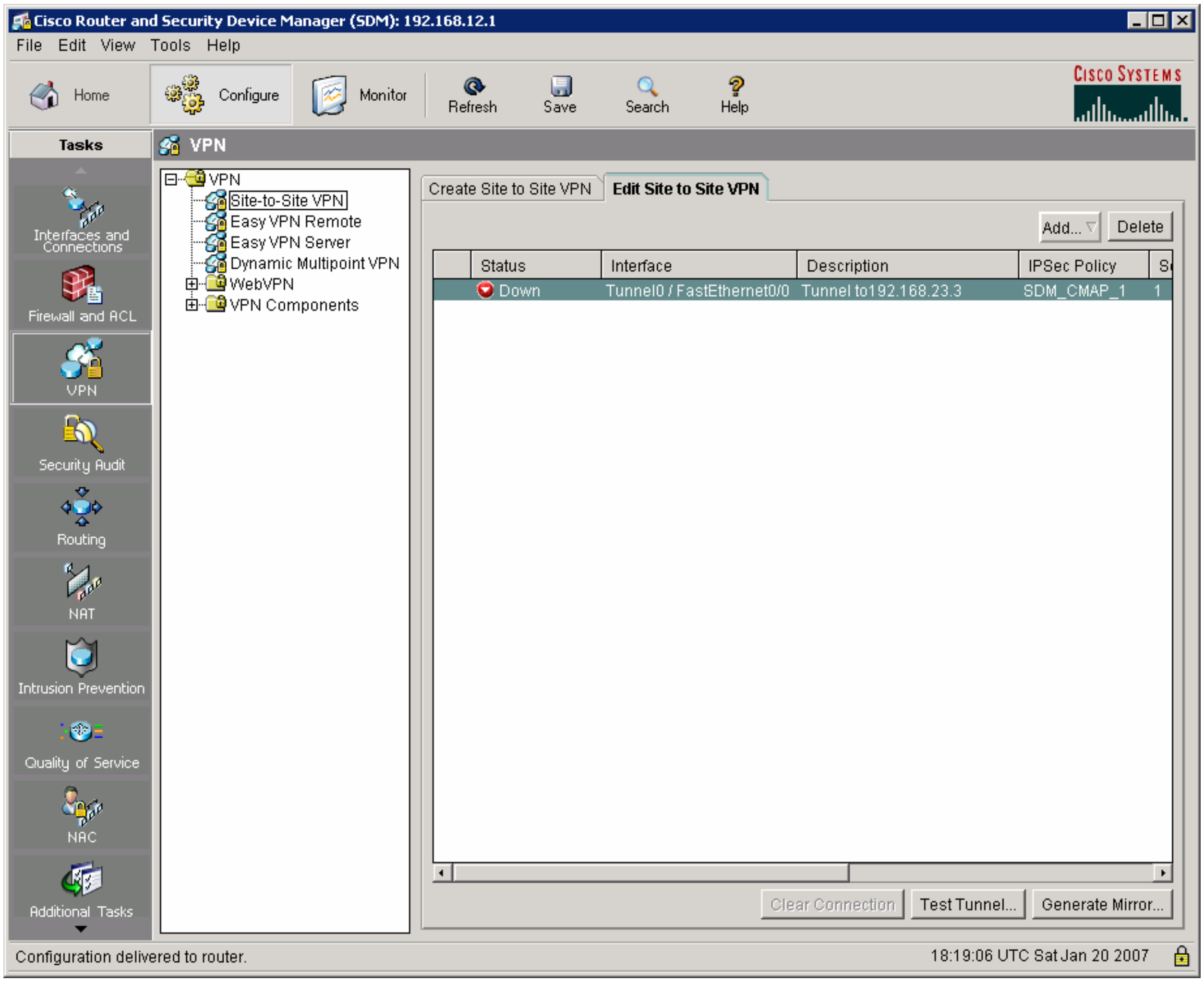
Command Delivery Status:

```
Preparing commands for delivery...
Submitting 51 commands, please wait...
Configuration delivered to router.
```

Figure 4-15: Command Delivery Progress Indicator

Step 5: Generate a Mirror Configuration for R3

In the **Edit Site-to-Site VPN** tab of SDM, click the **Generate Mirror...** button. An incomplete mirror configuration for R3 is generated.



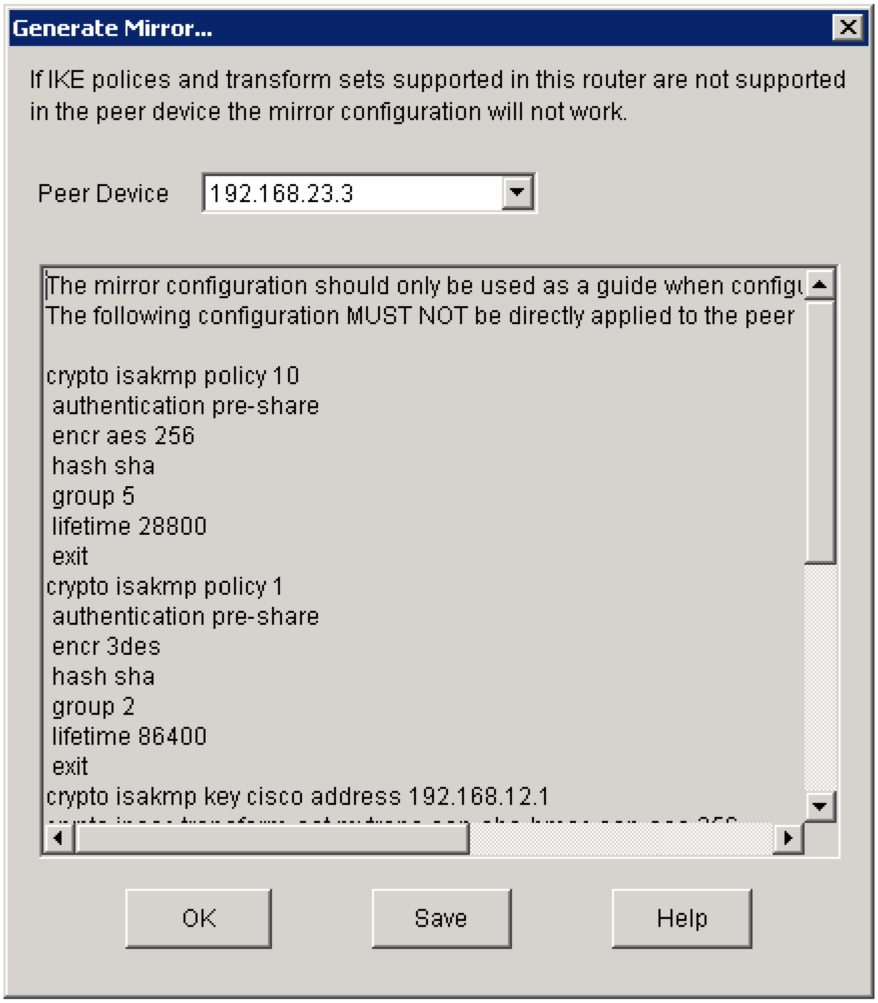
Status	Interface	Description	IPSec Policy	Se
Down	Tunnel0 / FastEthernet0/0	Tunnel to192.168.23.3	SDM_CMAP_1	1

Configuration delivered to router.

18:19:06 UTC Sat Jan 20 2007

Figure 5-1: Edit Site-to-Site VPN Tab

Copy the commands shown in the dimmed text box to the Windows clipboard.



If IKE polices and transform sets supported in this router are not supported in the peer device the mirror configuration will not work.

Peer Device 192.168.23.3

```
The mirror configuration should only be used as a guide when configu
The following configuration MUST NOT be directly applied to the peer

crypto isakmp policy 10
 authentication pre-share
 encr aes 256
 hash sha
 group 5
 lifetime 28800
 exit
crypto isakmp policy 1
 authentication pre-share
 encr 3des
 hash sha
 group 2
 lifetime 86400
 exit
crypto isakmp key cisco address 192.168.12.1
```

Figure 5-2: Mirror Router Configuration Script

In global configuration mode on R3, paste in this configuration.

```
R3(config)# crypto isakmp policy 10
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# encr aes 256
R3(config-isakmp)# hash sha
R3(config-isakmp)# group 5
R3(config-isakmp)# lifetime 28800
R3(config-isakmp)# exit
R3(config)# crypto isakmp policy 1
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# encr 3des
R3(config-isakmp)# hash sha
R3(config-isakmp)# group 2
R3(config-isakmp)# lifetime 86400
R3(config-isakmp)# exit
R3(config)# crypto isakmp key cisco address 192.168.12.1
R3(config)# crypto ipsec transform-set mytrans esp-sha-hmac esp-aes 256
R3(cfg-crypto-trans)# mode tunnel
R3(cfg-crypto-trans)# exit
R3(config)# ip access-list extended SDM_1
R3(config-ext-nacl)# remark SDM_ACL Category=4
R3(config-ext-nacl)# permit gre host 192.168.23.3 host 192.168.12.1
```

```
R3(config-ext-nacl)# exit
R3(config)# crypto map SDM_CMAP_1 1 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R3(config-crypto-map)# description Apply the crypto map on the peer router's
interface having IP address 192.168.23.3 that connects to this router.
R3(config-crypto-map)# set transform-set mytrans
R3(config-crypto-map)# set peer 192.168.12.1
R3(config-crypto-map)# match address SDM_1
R3(config-crypto-map)# set security-association lifetime seconds 3600
R3(config-crypto-map)# set security-association lifetime kilobytes 4608000
R3(config-crypto-map)# exit
```

Unfortunately, the configuration generated from SDM is incomplete. There is no GRE tunnel interface and the crypto map must also be applied to the physical interface on R3. The EIGRP AS 2 routing process is also missing from the configuration. To get a general idea of what the tunnel configuration should look like, look at R1's tunnel interface.

```
R1# show run | interface tunnel 0
Building configuration...

Current configuration : 190 bytes
!
interface Tunnel0
 ip address 172.16.13.1 255.255.255.0
 ip mtu 1420
 tunnel source FastEthernet0/0
 tunnel destination 192.168.23.3
 tunnel path-mtu-discovery
 crypto map SDM_CMAP_1
end
```

Reuse this configuration, but swap the IP addresses and interfaces as necessary. You may see a warning about IKE failing because there is no key for the remote peer with that IP address. This is normal.

```
R3(config)# interface Tunnel 0
R3(config-if)# ip address 172.16.13.3 255.255.255.0
R3(config-if)# ip mtu 1420
R3(config-if)# tunnel source Serial0/0/1
R3(config-if)# tunnel destination 192.168.12.1
R3(config-if)# tunnel path-mtu-discovery
R3(config-if)# crypto map SDM_CMAP_1
```

Apply the crypto map that was created to the serial interface to encrypt GRE traffic.

```
R3(config)# interface serial 0/0/1
R3(config-if)# crypto map SDM_CMAP_1
```

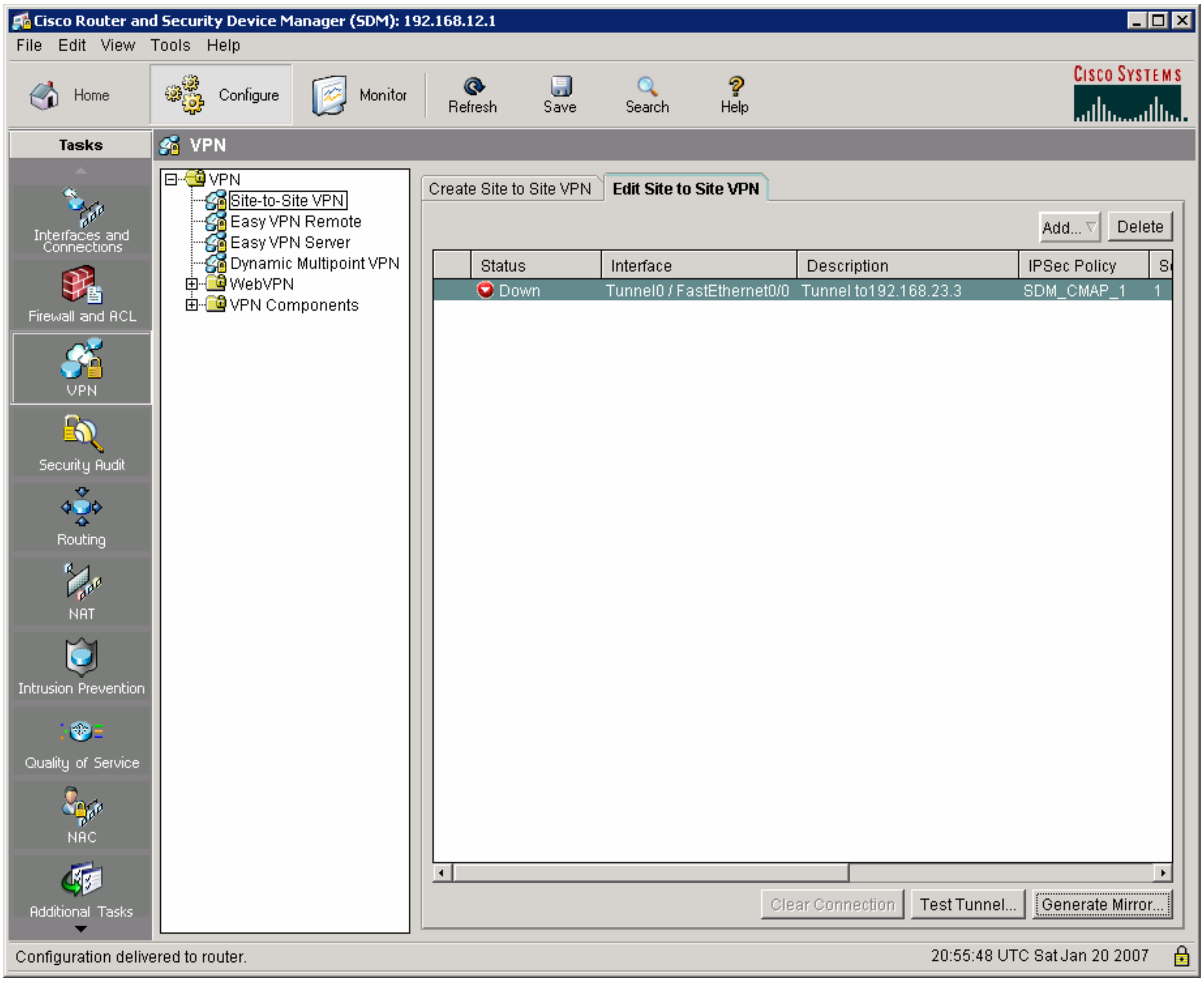
Finally, create the EIGRP AS 2 process on R3. Disable automatic summarization and add the entire 172.16.0.0/16 major network to it. You should see the EIGRP adjacency come up over the tunnel interface.

```
R3(config)# router eigrp 2
R3(config-router)# no auto-summary
R3(config-router)# network 172.16.0.0
```

Will the 172.16.13.0/24 network be reachable from R2?

Step 6: Verify Tunnel Configuration through SDM

You can use SDM to verify the tunnel configuration. To do this, click the **Test Tunnel...** button on the **Edit Site to Site VPN** tab.



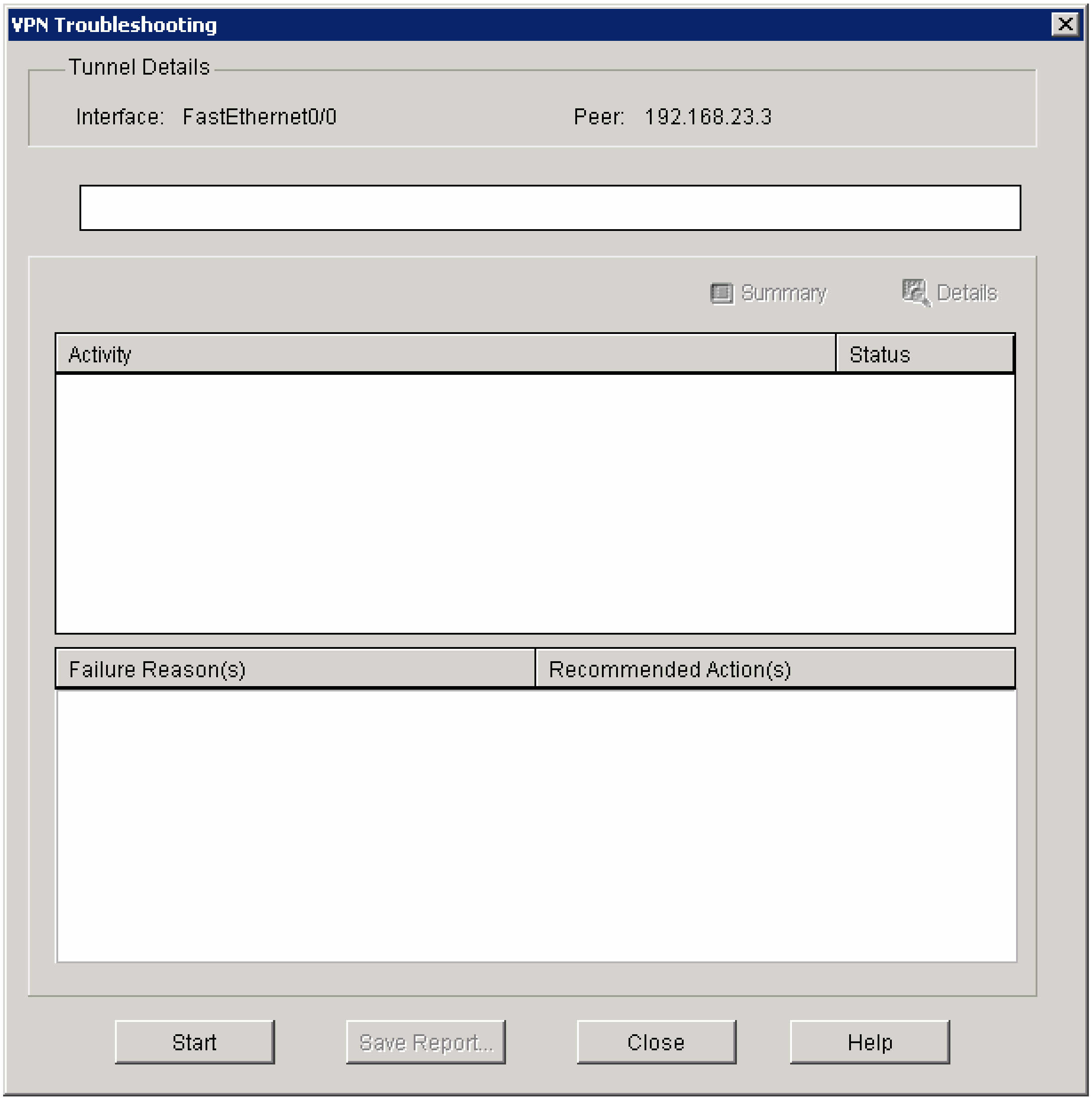
Status	Interface	Description	IPSec Policy	S
Down	Tunnel0 / FastEthernet0/0	Tunnel to192.168.23.3	SDM_CMAP_1	1

Configuration delivered to router.

20:55:48 UTC Sat Jan 20 2007

Figure 6-1: Edit Site-to-Site VPN Tab

Click **Start** and SDM will verify the tunnel status.



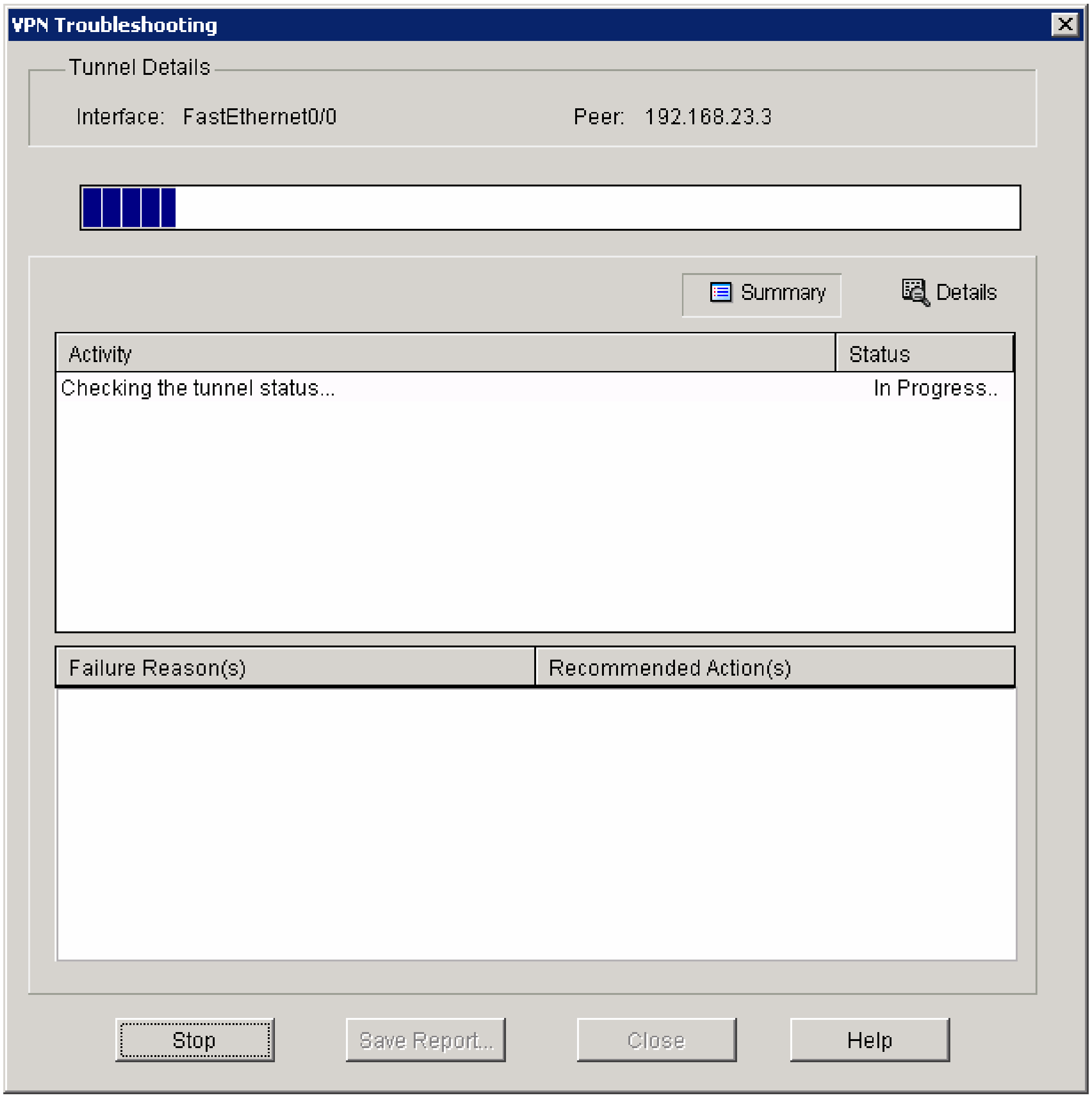
VPN Troubleshooting

Tunnel Details

Interface: FastEthernet0/0 Peer: 192.168.23.3

Figure 6-2: VPN Testing Window

When verification is complete, a success message should appear. Click **OK**.



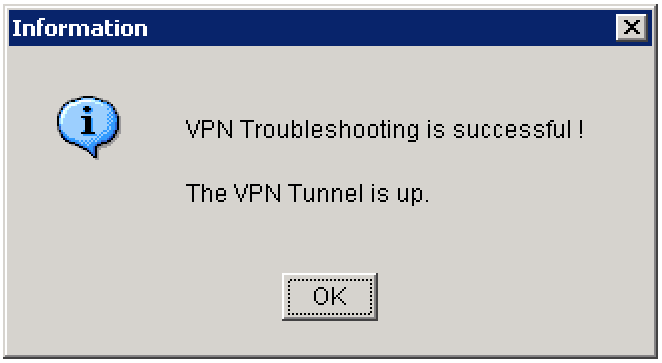
Interface: FastEthernet0/0

Peer: 192.168.23.3

Activity	Status
Checking the tunnel status...	In Progress..

Failure Reason(s)	Recommended Action(s)

Figure 6-3: VPN Test In Progress

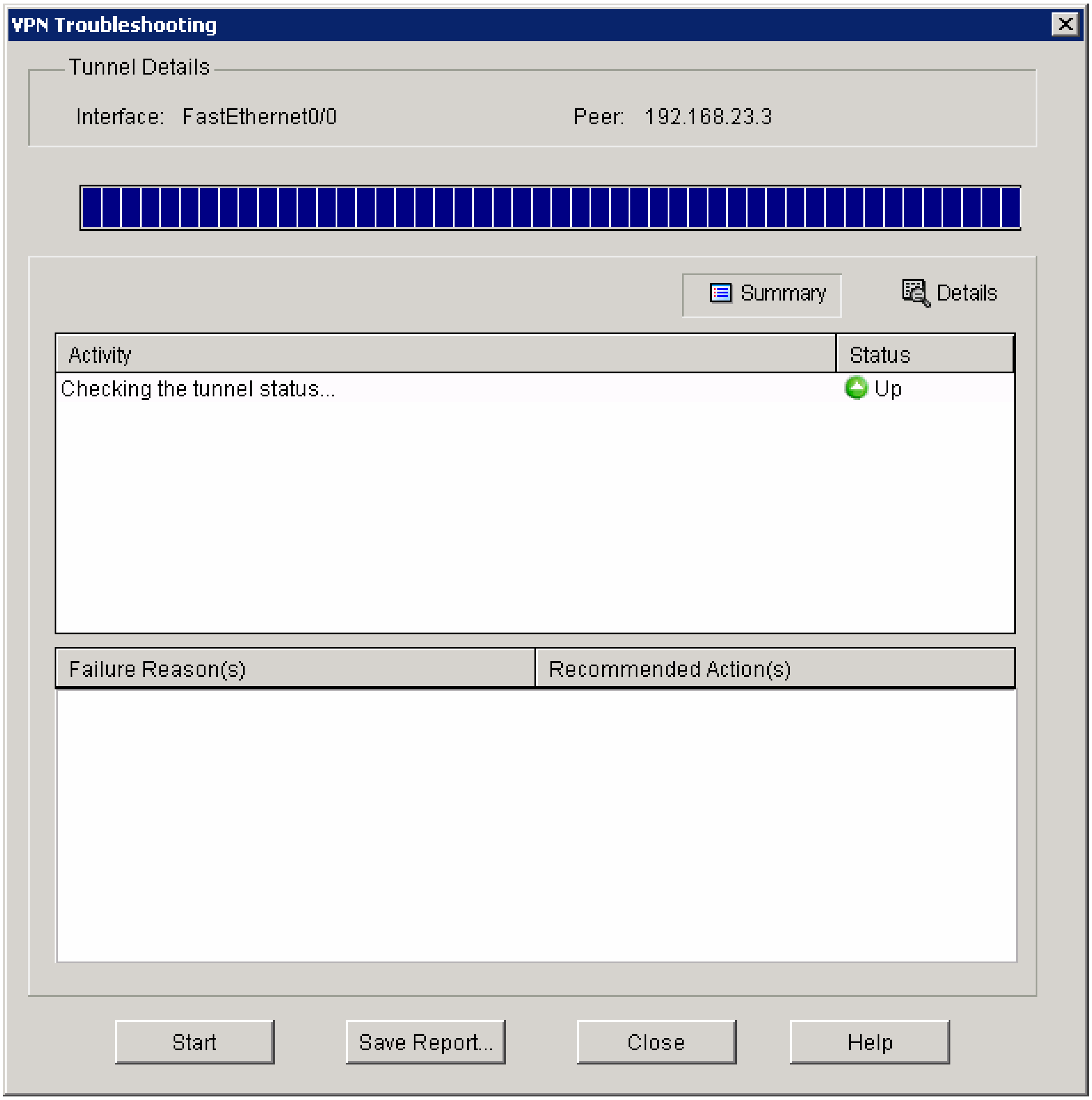


VPN Troubleshooting is successful !

The VPN Tunnel is up.

Figure 6-4: Successful VPN Test Status Window

The status of “up” should be displayed in this window. Click **Close** when you are done reading this window. You will be returned to the main SDM window.



Activity	Status
Checking the tunnel status...	Up

Figure 6-5: Detailed VPN Test Results

Verify that you have partial IP connectivity at this point with the following Toolkit Command Language (TCL) script.

```
tclsh

foreach address {
172.16.1.1
172.16.3.1
172.16.13.1
172.16.13.3
192.168.12.1
192.168.12.2
192.168.23.2
192.168.23.3
} { ping $address }

tclquit
```

Compare your output with the output shown in Appendix A. Troubleshoot as necessary. Remember that R2 should not be able to reach any subnets of the 172.16.0.0/16 network.

Challenge: Use Wireshark to Monitor Encryption of Traffic

You can observe packets on the wire using Wireshark and see how their content looks unencrypted and then encrypted. To do this, first configure a SPAN session on the switch and open up Wireshark on a host attached to the SPAN destination port. You can use the host that you used for SDM because you don't need it anymore to configure the VPNs. If you do not know how to do this, refer to Lab 3.3: Configuring Wireshark and SPAN.

Next, you will remove the **crypto map** statements on R1 and R3. View the current configuration on the FastEthernet0/0 interface on R1 and Serial0/0/1 as shown below.

Then, issue the **no crypto map** *name* command in interface configuration mode to remove the Internet Security Association and Key Management Protocol (ISAKMP) security association. The router may issue a warning that ISAKMP is now off.

```
R1# show run | interface fastethernet 0/0
Building configuration...

Current configuration : 120 bytes
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 duplex auto
 speed auto
 crypto map SDM_CMAP_1
end

R1# configure terminal
R1(config)# interface fastethernet0/0
R1(config-if)# no crypto map SDM_CMAP_1
*Jan 16 06:02:58.999: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF

R3# show run | interface serial 0/0/1
Building configuration...

Current configuration : 91 bytes
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map SDM_CMAP_1
end
```

```
R3# configure terminal
R3(config)# interface serial0/0/1
R3(config-if)# no crypto map SDM_CMAP_1
*Jan 16 06:05:36.038: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
```

The traffic we want to sniff will be telnet traffic, so enable telnet access and an enable password on R3 if you haven't already.

```
R3(config)# enable secret cisco
R3(config)# line vty 0 4
R3(config-line)# password cisco
R3(config-line)# login
```

Have Wireshark start sniffing packets that it receives via the SPAN session.

Choose **Capture > Interfaces...**. Then click the **Start** button associated with the interface connected to the SPAN destination port. SPAN should start capturing packets on the line, so you can now telnet from R1's loopback to R3's loopback. To send telnet traffic, use the **telnet** *destination* command.

Do you need to use the **/source** attribute in the telnet command? Explain.

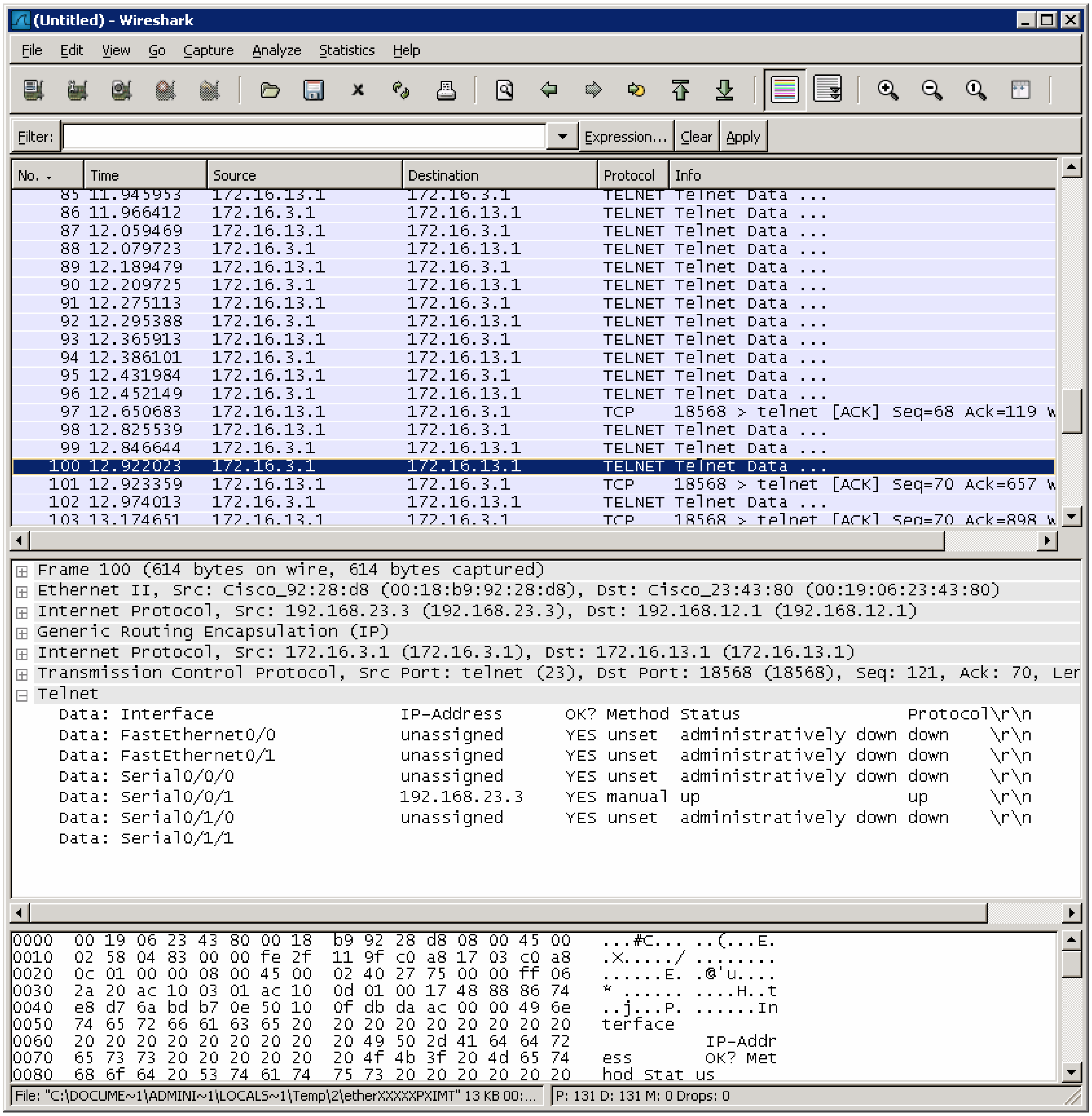
First, begin capturing using Wireshark. Then, begin the telnet session. Once you are connected to R3, try issuing a command or two and then logging out.

The packets will be routed through the tunnel interface towards the loopback on R3, so Wireshark will display the GRE packets. Remember to have Wireshark capturing when you start the telnet session. Once you are connected to the remote router, try issuing a command or two and then logging out.

```
R1# telnet 172.16.3.1
Trying 172.16.3.1 ... Open


User Access Verification

Password:
R3> enable
Password:
R3# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            unassigned      YES unset  administratively down
down
FastEthernet0/1            unassigned      YES unset  administratively down
down
Serial0/0/0                unassigned      YES unset  administratively down
down
```

```
Serial0/0/1                192.168.23.3    YES manual up                    up
Serial0/1/0                unassigned      YES unset  administratively down
down
Serial0/1/1                unassigned      YES unset  administratively down
down
Loopback0                  172.16.3.1      YES manual up                    up
Tunnel0                    172.16.13.3     YES manual up                    up
R3# exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

Now, take a look at the output. Notice that Wireshark is smart enough to classify these packets as telnet traffic, even though the actual packets are GRE. Looking in the middle pane in Wireshark, it will show the multiple layers of encapsulation, including the GRE information. Notice that since we disabled encryption, you can easily read the plaintext strings of the telnet session in Wireshark.



```
No. -  Time       Source        Destination   Protocol Info
  85 11.945953  172.16.13.1   172.16.3.1    TELNET   Telnet Data ...
  86 11.966412  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
  87 12.059469  172.16.13.1   172.16.3.1    TELNET   Telnet Data ...
  88 12.079723  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
  89 12.189479  172.16.13.1   172.16.3.1    TELNET   Telnet Data ...
  90 12.209725  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
  91 12.275113  172.16.13.1   172.16.3.1    TELNET   Telnet Data ...
  92 12.295388  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
  93 12.365913  172.16.13.1   172.16.3.1    TELNET   Telnet Data ...
  94 12.386101  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
  95 12.431984  172.16.13.1   172.16.3.1    TELNET   Telnet Data ...
  96 12.452149  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
  97 12.650683  172.16.13.1   172.16.3.1    TCP      18568 > telnet [ACK] Seq=68 Ack=119 W
  98 12.825539  172.16.13.1   172.16.3.1    TELNET   Telnet Data ...
  99 12.846644  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
 100 12.922023  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
 101 12.923359  172.16.13.1   172.16.3.1    TCP      18568 > telnet [ACK] Seq=70 Ack=657 W
 102 12.974013  172.16.3.1    172.16.13.1   TELNET   Telnet Data ...
 103 13.174651  172.16.13.1   172.16.3.1    TCP      18568 > telnet [ACK] Seq=70 Ack=898 W
```

```
⊞ Frame 100 (614 bytes on wire, 614 bytes captured)
⊞ Ethernet II, Src: Cisco_92:28:d8 (00:18:b9:92:28:d8), Dst: Cisco_23:43:80 (00:19:06:23:43:80)
⊞ Internet Protocol, Src: 192.168.23.3 (192.168.23.3), Dst: 192.168.12.1 (192.168.12.1)
⊞ Generic Routing Encapsulation (IP)
⊞ Internet Protocol, Src: 172.16.3.1 (172.16.3.1), Dst: 172.16.13.1 (172.16.13.1)
⊞ Transmission Control Protocol, Src Port: telnet (23), Dst Port: 18568 (18568), Seq: 121, Ack: 70, Ler
⊟ Telnet
    Data: Interface                  IP-Address      OK? Method Status                Protocol\r\n
    Data: FastEthernet0/0            unassigned      YES unset  administratively down down    \r\n
    Data: FastEthernet0/1            unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/0/0                unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/0/1                192.168.23.3    YES manual up                    up      \r\n
    Data: Serial0/1/0                unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/1/1
```

```
0000  00 19 06 23 43 80 00 18  b9 92 28 d8 08 00 45 00   ...#C... ..(...E.
0010  02 58 04 83 00 00 fe 2f  11 9f c0 a8 17 03 c0 a8   .X...../ ........
0020  0c 01 00 00 08 00 45 00  02 40 27 75 00 00 ff 06   ......E. .@'u....
0030  2a 20 ac 10 03 01 ac 10  0d 01 00 17 48 88 86 74   * ...... ....H..t
0040  e8 d7 6a bd b7 0e 50 10  0f db da ac 00 00 49 6e   ..j...P. ......In
0050  74 65 72 66 61 63 65 20  20 20 20 20 20 20 20 20   terface
0060  20 20 20 20 20 20 20 20  20 49 50 2d 41 64 64 72            IP-Addr
0070  65 73 73 20 20 20 20 20  20 4f 4b 3f 20 4d 65 74   ess       OK? Met
0080  68 6f 64 20 53 74 61 74  75 73 20 20 20 20 20 20   hod Stat us
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\2\etherXXXXXPXIMT" 13 KB 00:... P: 131 D: 131 M: 0 Drops: 0

Figure 7-1: Detailed Packet Data on Telnet String Sent From R1

Based on this output, you can see how easy it is for someone who is in the path of sensitive data to view unencrypted or clear text traffic.

Now, you will reapply the cryptography settings on R1 and R3 and begin a telnet session from R1 to R3 as before.

Begin by reapplying the crypto maps you removed earlier on R1 and R3.

```
R1(config)# interface fastethernet 0/0
R1(config-if)# crypto map SDM_CMAP_1
```

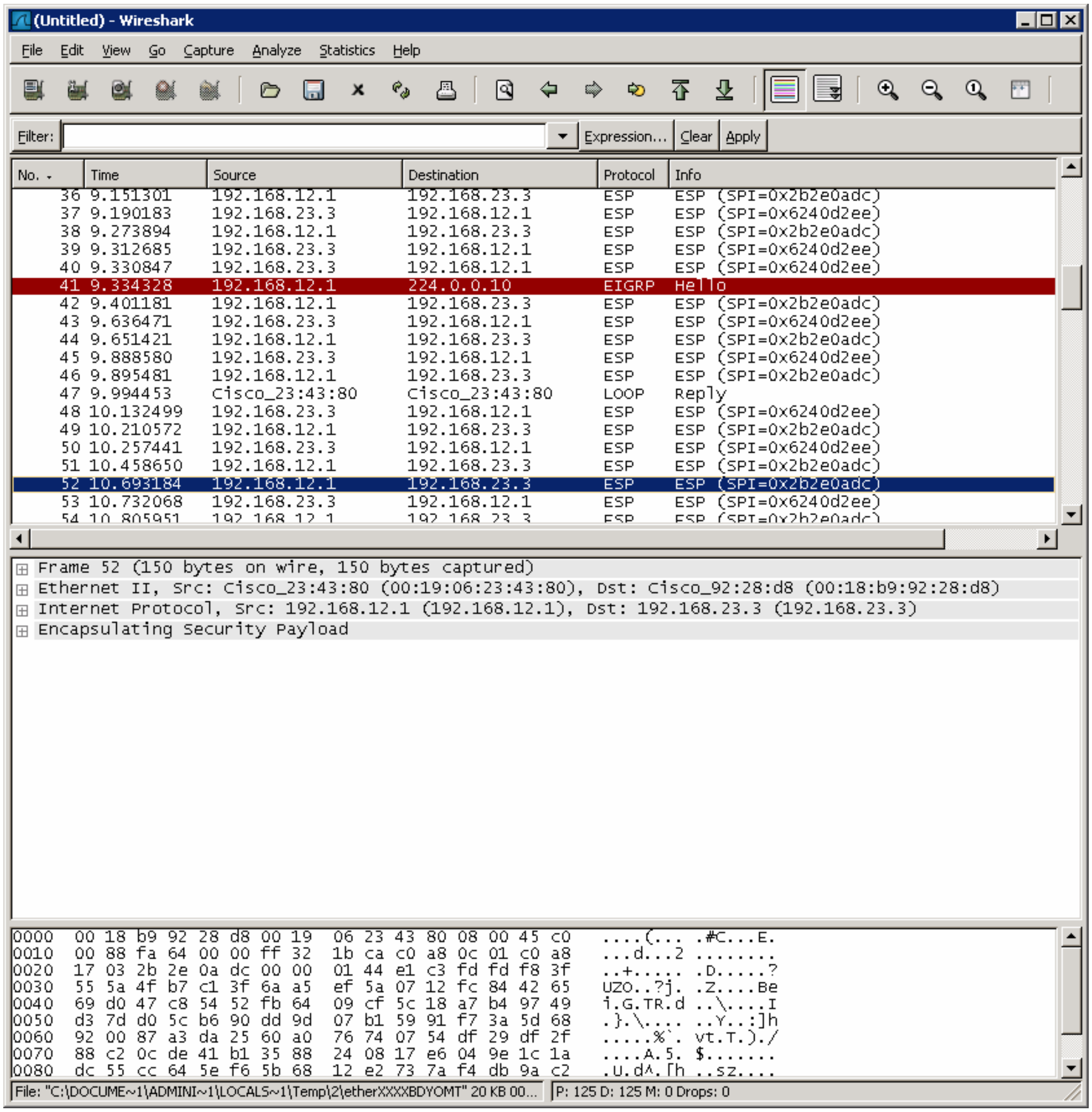
```
R3(config)# interface serial0/0/1
R3(config-if)# crypto map SDM_CMAP_1
```

Start the packet capturing again in Wireshark, and then issue the same telnet sequence that you did previously.

```
R1# telnet 172.16.3.1
Trying 172.16.3.1 ... Open


User Access Verification

Password:
R3> enable
Password:
R3# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            unassigned      YES unset  administratively down
down
FastEthernet0/1            unassigned      YES unset  administratively down
down
Serial0/0/0                unassigned      YES unset  administratively down
down
Serial0/0/1                192.168.23.3    YES manual up                    up
Serial0/1/0                unassigned      YES unset  administratively down
down
Serial0/1/1                unassigned      YES unset  administratively down
down
Loopback0                  172.16.3.1      YES manual up                    up
Tunnel0                    172.16.13.3     YES manual up                    up
R3#exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

End your Wireshark capture when you are finished with the telnet session.

As far as the user is concerned, the telnet session seems the same with and without encryption. However, the packet capture from Wireshark shows that the VPN is actively encapsulating and encrypting packets.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. - Time        Source            Destination       Protocol Info
   36 9.151301    192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   37 9.190183    192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   38 9.273894    192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   39 9.312685    192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   40 9.330847    192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   41 9.334328    192.168.12.1      224.0.0.10        EIGRP    Hello
   42 9.401181    192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   43 9.636471    192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   44 9.651421    192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   45 9.888580    192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   46 9.895481    192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   47 9.994453    Cisco_23:43:80    Cisco_23:43:80    LOOP     Reply
   48 10.132499   192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   49 10.210572   192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   50 10.257441   192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   51 10.458650   192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   52 10.693184   192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
   53 10.732068   192.168.23.3      192.168.12.1      ESP      ESP (SPI=0x6240d2ee)
   54 10.805951   192.168.12.1      192.168.23.3      ESP      ESP (SPI=0x2b2e0adc)
```

```
⊞ Frame 52 (150 bytes on wire, 150 bytes captured)
⊞ Ethernet II, Src: Cisco_23:43:80 (00:19:06:23:43:80), Dst: Cisco_92:28:d8 (00:18:b9:92:28:d8)
⊞ Internet Protocol, Src: 192.168.12.1 (192.168.12.1), Dst: 192.168.23.3 (192.168.23.3)
⊞ Encapsulating Security Payload
```

```
0000  00 18 b9 92 28 d8 00 19  06 23 43 80 08 00 45 c0   ....(... .#C...E.
0010  00 88 fa 64 00 00 ff 32  1b ca c0 a8 0c 01 c0 a8   ...d...2 ........
0020  17 03 2b 2e 0a dc 00 00  01 44 e1 c3 fd fd f8 3f   ..+..... .D.....?
0030  55 5a 4f b7 c1 3f 6a a5  ef 5a 07 12 fc 84 42 65   UZO..?j. .Z....Be
0040  69 d0 47 c8 54 52 fb 64  09 cf 5c 18 a7 b4 97 49   i.G.TR.d ..\....I
0050  d3 7d d0 5c b6 90 dd 9d  07 b1 59 91 f7 3a 5d 68   .}.\.... ..Y..:]h
0060  92 00 87 a3 da 25 60 a0  76 74 07 54 df 29 df 2f   .....%`. vt.T.)./
0070  88 c2 0c de 41 b1 35 88  24 08 17 e6 04 9e 1c 1a   ....A.5. $.......
0080  dc 55 cc 64 5e f6 5b 68  12 e2 73 7a f4 db 9a c2   .U.d^.[h ..sz....
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\2\etherXXXXBDYOMT" 20 KB 00... P: 125 D: 125 M: 0 Drops: 0

Figure 8-2: Detailed Packet Data on Encrypted Telnet String Sent From R1

Notice that the protocol is not telnet (TCP port 23), but the Encapsulating Security Protocol (ESP, IP protocol number 50). Remember, all traffic here matches the IPsec access list.

Also, notice that the source and destination are not the actual source and destination of the addresses participating in this telnet conversation. Rather, they are the endpoints of the VPN.

Finally, and most important, if you look at the contents of these packets in Wireshark, no matter how you try to format or filter them, you will not be able to see what data was originally inside.

The encryption suite provided by IPsec successfully secures data through authentication, encryption, and data-integrity services.

Appendix A: TCL Script Output

```
tclsh

foreach address {
172.16.1.1
172.16.3.1
172.16.13.1
172.16.13.3
192.168.12.1
192.168.12.2
192.168.23.2
192.168.23.3
} { ping $address }

R1# tclsh
R1(tcl)#
R1(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#172.16.3.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
R1(tcl)# tclquit

R2# tclsh
R2(tcl)#
R2(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#172.16.3.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/56/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/36 ms
R2(tcl)# tclquit

R3# tclsh
R3(tcl)#
R3(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#172.16.3.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#192.168.12.1
+>(tcl)#192.168.12.2
+>(tcl)#192.168.23.2
+>(tcl)#192.168.23.3
+>(tcl)#} { ping $address }
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/64 ms
R3(tcl)# tclquit
```

Final Configurations

```
R1# show run
hostname R1
!
crypto pki trustpoint TP-self-signed-1455051929
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1455051929
 revocation-check none
 rsakeypair TP-self-signed-1455051929
!
crypto pki certificate chain TP-self-signed-1455051929
 certificate self-signed 01
  3082023A 308201A3 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 31343535 30353139 3239301E 170D3037 30313139 30303337
  30375A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 34353530
  35313932 3930819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100B2AE D3DF3BE4 D1323EDA B5A4EC54 2E3F3B46 20204095 3FA3FE01 0B3F5C84
  283D08A2 1023886D 6791AD57 DFFD39EE C453D2EF 0555041C A1B9CCCA 82216AAB
  FBD731B8 465F3B57 4E7D76C3 54BE49F3 B82D0AF7 74005E9E 59736B5A 90D63697
  EABA4FE5 973B7F4A D0C2B77A 5B03A5C7 4376DE69 3B784063 726D0E9C 51065FEC
  E4290203 010001A3 62306030 0F060355 1D130101 FF040530 030101FF 300D0603
  551D1104 06300482 02523130 1F060355 1D230418 30168014 976FC125 5539A586
  94800545 D6F943AD A89E2B22 301D0603 551D0E04 16041497 6FC12555 39A58694
  800545D6 F943ADA8 9E2B2230 0D06092A 864886F7 0D010104 05000381 81000E3E
  9C147BD6 EF49FD63 943C943A FD5773A4 559346F8 0F33886E 26A84C33 2FB0AC36
  FF5F849E 782BAB73 D94FFEAB 7BE8F8E1 E72238F9 A70A7709 8854878F 53105BB2
  3996E9E2 CD907377 101D3E5C 62A7CC8B 3C268997 CCF09774 909EE66A F09A9D3E
  BBB99FC4 96E50636 1CEC52CB 9A45E8DB 7317DE15 06350825 9ECCD529 B3A7
  quit
username ciscosdm privilege 15 password 0 ciscosdm
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
 lifetime 28800
crypto isakmp key cisco address 192.168.23.3
!
!
crypto ipsec transform-set mytrans esp-aes 256 esp-sha-hmac
!
crypto map SDM_CMAP_1 1 ipsec-isakmp
 description Tunnel to192.168.23.3
 set peer 192.168.23.3
 set transform-set mytrans
 match address 100
!
interface Tunnel0
 ip address 172.16.13.1 255.255.255.0
 ip mtu 1420
 tunnel source FastEthernet0/0
 tunnel destination 192.168.23.3
 tunnel path-mtu-discovery
 crypto map SDM_CMAP_1
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 crypto map SDM_CMAP_1
 no shut
!
router eigrp 1
 network 192.168.12.0
 no auto-summary
!
router eigrp 2
 network 172.16.13.0 0.0.0.255
 network 172.16.0.0
 no auto-summary
!
ip http server
ip http authentication local
ip http secure-server
!
access-list 100 remark SDM_ACL Category=4
access-list 100 permit gre host 192.168.12.1 host 192.168.23.3
!
line vty 0 4
 login local
 transport input telnet ssh
end
```

```
R2# show run
hostname R2
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 no shut
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 clock rate 64000
 no shut
!
router eigrp 1
 network 192.168.12.0
 network 192.168.23.0
 no auto-summary
!
end

R3# show run
hostname R3
!
enable secret 5 $1$xbvr$6YNBOCZFuWyM3UTmlHK03.
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
 lifetime 28800
crypto isakmp key cisco address 192.168.12.1
!
!
crypto ipsec transform-set mytrans esp-aes 256 esp-sha-hmac
!
crypto map SDM_CMAP_1 1 ipsec-isakmp
 description Apply the crypto map on the peer router's interface having IP
address 192.168.23.3 that connects to this router.
 set peer 192.168.12.1
 set transform-set mytrans
 match address SDM_1
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Tunnel0
 ip address 172.16.13.3 255.255.255.0
 ip mtu 1420
 tunnel source Serial0/0/1
 tunnel destination 192.168.12.1
 tunnel path-mtu-discovery
 crypto map SDM_CMAP_1
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map SDM_CMAP_1
 no shut
!
```

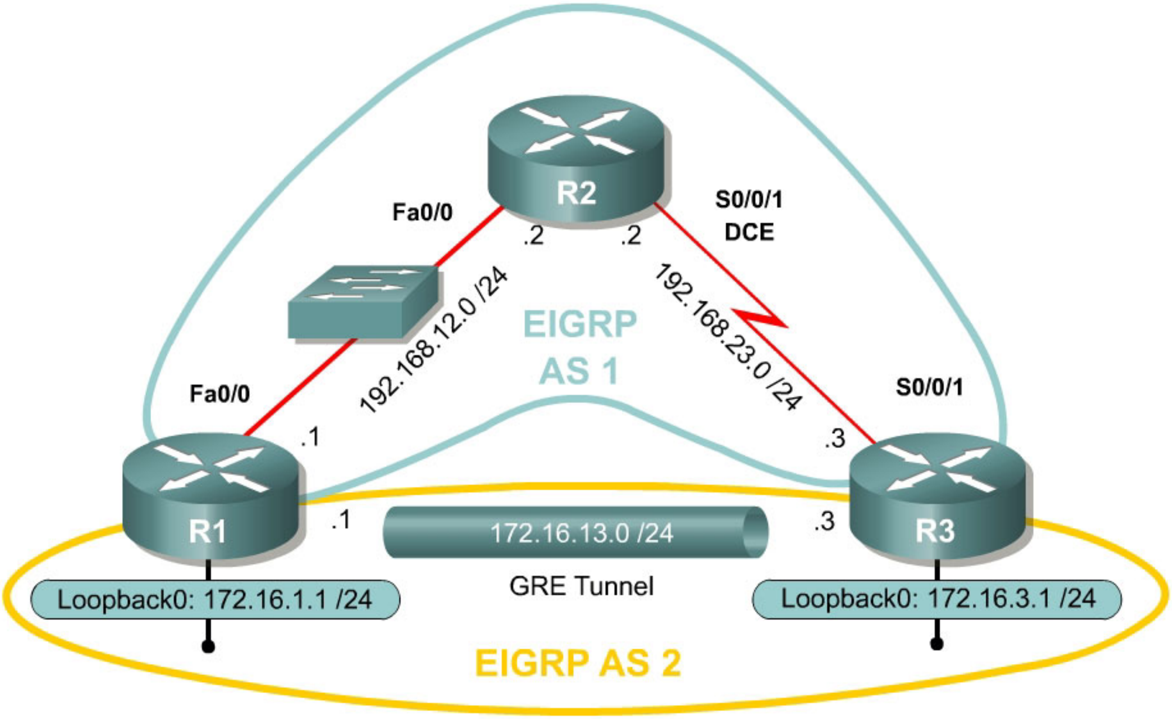
```
router eigrp 1
 network 192.168.23.0
 no auto-summary
!
router eigrp 2
 network 172.16.0.0
 no auto-summary
!
ip access-list extended SDM_1
 remark SDM_ACL Category=4
 permit gre host 192.168.23.3 host 192.168.12.1
!
line vty 0 4
 password ccie
 login
end
```

Lab 3.7 Configuring a Secure GRE Tunnel with the IOS CLI

Learning Objectives

- Configure EIGRP on the routers
- Create a GRE tunnel between two routers
- Use IPsec to secure the GRE tunnel

Topology Diagram



Scenario

In this lab, you will use the Cisco Command Line Interface (CLI) to configure a secure generic routing encapsulation (GRE) tunnel using IPsec. You will also use IPsec to secure traffic going through the tunnel. It will help you to have previously completed Labs 3.2 and 3.5 since this lab is a combination of the two. Lab 3.8 also addresses a newer way to configure this type of tunnel, in the configuring IPsec VTIs lab. This newer method combines encryption into the tunnel configuration.

Step 1: Configure Addressing

Configure the interfaces with the addresses as shown in the topology above. Set the clock rate on the appropriate interfaces and issue the **no shutdown** command as necessary. Verify that you have connectivity across the local subnet with the **ping** command. Do not set up the tunnel interface until the next step.

```
R1# configure terminal
R1(config)# interface loopback0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface fastethernet0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown

R2# configure terminal
R2(config)# interface fastethernet0/0
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/0/1
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3# configure terminal
R3(config)# interface loopback0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface serial0/0/1
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Configure EIGRP AS 1

Configure EIGRP AS 1 for the major networks 192.168.12.0/24 and 192.168.23.0/24. Do not include the networks in the diagram falling in the 172.16.0.0/16 range. The Class C networks will serve as the transit networks for the tunnel network. Make sure you disable EIGRP automatic summarization.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 192.168.12.0

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 192.168.12.0
R2(config-router)# network 192.168.23.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
R3(config-router)# network 192.168.23.0
Verify that R1 and R3 can see the remote transit network with show ip route
```

Step 3: Configure the GRE Tunnel

To configure a GRE tunnel, enter interface configuration mode with the **interface tunnel** *number* command from global configuration mode. For simplicity, use tunnel number 0 on both routers. Next, configure an IP address with **ip address** *address mask* the way you would on any other interface. Finally, assign a source and destination address for the tunnel with **tunnel source** *address* and **tunnel destination** *address*, respectively. The source can also be specified by interface. These addresses specify the endpoints of the tunnel, so our GRE traffic will be encapsulated with the source address and decapsulated at the destination address. You will not need to configure a tunnel mode because the default tunnel mode is GRE.

```
R1(config)# interface tunnel 0
R1(config-if)# ip address 172.16.13.1 255.255.255.0
R1(config-if)# tunnel source fastethernet0/0
R1(config-if)# tunnel destination 192.168.23.3

R3(config)# interface tunnel0
R3(config-if)# ip address 172.16.13.3 255.255.255.0
R3(config-if)# tunnel source serial0/0/1
R3(config-if)# tunnel destination 192.168.12.1
```

Verify that you can **ping** across the tunnel to the other side. If you can do this, you have successfully set up the tunnel.

```
R1# ping 172.16.13.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms

R3# ping 172.16.13.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/72 ms
```

With what source and destination IP address are these packets sent out of the FastEthernet0/0 interface on R1? Why?

What IP protocol number do these packets have?

Step 4: Configure EIGRP AS 2 over the Tunnel

Now that you set up the GRE tunnel, implement routing through it the way you would any other interface. Configure EIGRP AS 2 to route the entire 172.16.0.0/16 major network. Disable automatic summarization. Remember that R2 is not participating in this routing process and will not need to be configured with EIGRP AS 2.

```
R1(config)# router eigrp 2
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.0.0

R3(config)# router eigrp 2
R3(config-router)# no auto-summary
R3(config-router)# network 172.16.0.0
```

You should observe EIGRP neighbor adjacencies become active with messages logged to the console. If not, troubleshoot by ensuring that you can ping from 192.168.12.1 to 192.168.23.3 and vice versa. Also check that you have configured the tunnel interfaces above correctly.

If you have configured this step correctly, you should be able to ping from R1’s loopback interface to R3’s loopback successfully.

Step 5: Create IKE Policies and Peers

Configure an Internet Key Exchange (IKE) policy and peer key. Create an IKE policy using the information that follows. If your IOS image doesn’t support all of the settings, configure what you can. Just make sure your VPN settings match on both ends of the connection.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# hash sha
R1(config-isakmp)# group 5
R1(config-isakmp)# lifetime 3600

R3(config)# crypto isakmp policy 10
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# encryption aes 256
R3(config-isakmp)# hash sha
R3(config-isakmp)# group 5
R3(config-isakmp)# lifetime 3600
```

Of the three authentication methods available, which is considered the weakest?

What is currently the most secure encryption algorithm?

What is currently the most secure hash algorithm?

Which of the Diffie-Hellman groups is considered weakest?

Next, configure each peer using the key “cisco” for Internet Security Association and Key Management Protocol (ISAKMP).

```
R1(config)# crypto isakmp key cisco address 192.168.23.3

R3(config)# crypto isakmp key cisco address 192.168.12.1
```

Step 7: Create IPsec Transform Sets

On both endpoint routers, create an IPsec transform set with the following settings. If your routers do not support these settings, use whichever settings you can. Just keep it consistent on both routers.

```
R1(config)# crypto ipsec transform-set mytrans esp-aes 256 esp-sha-hmac ah-
sha-hmac
R1(cfg-crypto-trans)# exit
R1(config)#

R3(config)# crypto ipsec transform-set mytrans esp-aes 256 esp-sha-hmac ah-
sha-hmac
R3(cfg-crypto-trans)# exit
R3(config)#
```

Step 8: Define the Traffic to be Encrypted

On both endpoint routers, define traffic to be encrypted by IPsec to be GRE traffic with the source and destination as the tunnel endpoint addresses. Remember to keep the correct order of these networks on each router.

```
R1(config)# access-list 101 permit gre host 192.168.12.1 host 192.168.23.3

R3(config)# access-list 101 permit gre host 192.168.23.3 host 192.168.12.1
```

Step 9: Create and Apply Crypto Maps

On both endpoint routers, you will need to create and apply an IPsec crypto map to the outgoing interfaces to encrypt the GRE tunnel traffic. The EIGRP neighbor adjacency may “flap” (go down and then come back up) while the crypto map is configured on one router and not the other.

```
R1(config)# crypto map mymap 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R1(config-crypto-map)# match address 101
R1(config-crypto-map)# set peer 192.168.23.3
R1(config-crypto-map)# set transform-set mytrans
R1(config-crypto-map)# exit
R1(config)# interface fastethernet 0/0
R1(config-if)# crypto map mymap
*Jan 22 07:01:30.147: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON

R3(config)# crypto map mymap 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R3(config-crypto-map)# match address 101
R3(config-crypto-map)# set peer 192.168.12.1
R3(config-crypto-map)# set transform-set mytrans
R3(config-crypto-map)# interface serial 0/0/1
R3(config-if)# crypto map mymap
*Jan 22 07:02:47.726: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

NOTE: On certain older IOS releases, you may also need to apply the crypto map to the tunnel interface.

Step 10: Verify Crypto Operation

Verify that the number of packets is increasing by issuing the command **show crypto ipsec sa**, and monitoring the number of packet differences after issuing the command on a router.

```
R1# show crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: mymap, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.12.1/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (192.168.23.3/255.255.255.255/47/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 8, #pkts encrypt: 8, #pkts digest: 8
    #pkts decaps: 8, #pkts decrypt: 8, #pkts verify: 8
...
```

Wait a few seconds, then issue the **show crypto ipsec sa** command again.

```
R1# show crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: mymap, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.12.1/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (192.168.23.3/255.255.255.255/47/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
```

```
...
```

Although you have not issued another ping, packets are still being encrypted in the GRE tunnel and in the IPsec VPN.

Based on your knowledge of the configuration on R1 and R3, what packets are causing the packet count to increment as time passes?

For more crypto verification commands, consult Lab 3.5.

Challenge: Use Wireshark to Monitor Encryption of Traffic

You can observe packets on the wire using Wireshark and see how their content looks unencrypted and then encrypted. To do this, first configure a SPAN session on the switch and open up Wireshark on a host attached to the SPAN destination port. You can use the host that you used for SDM because you don’t need it anymore to configure the VPNs. If you do not know how to do this, refer to Lab 3.3: Configuring Wireshark and SPAN.

Next, you will remove the **crypto map** statements on R1 and R3. View the current configuration on the FastEthernet0/0 interface on R1 and Serial0/0/1 as shown below.

Then, issue the **no crypto map** *name* command in interface configuration mode to remove the ISAKMP security association. The router may issue a warning that ISAKMP is now off.

```
R1# show run interface fastethernet 0/0
Building configuration...

Current configuration : 120 bytes
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 duplex auto
 speed auto
 crypto map mymap
end

R1# configure terminal
R1(config)# interface fastethernet0/0
R1(config-if)# no crypto map mymap
*Jan 16 06:02:58.999: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF

R3# show run interface serial 0/0/1
Building configuration...

Current configuration : 91 bytes
```

```
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map mymap
end

R3# configure terminal
R3(config)# interface serial0/0/1
R3(config-if)# no crypto map mymap
*Jan 16 06:05:36.038: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
```

The traffic we want to sniff will be telnet traffic, so enable telnet access and an enable password on R3 if you haven't already.

```
R3(config)# enable secret cisco
R3(config)# line vty 0 4
R3(config-line)# password cisco
R3(config-line)# login
```

Have Wireshark start sniffing packets that it receives via the SPAN session.

Choose **Capture > Interfaces...**. Then click the **Start** button associated with the interface connected to the SPAN destination port. SPAN should start capturing packets on the line, so you can now telnet from R1's loopback to R3's loopback. To send telnet traffic, use the **telnet** *destination* command.

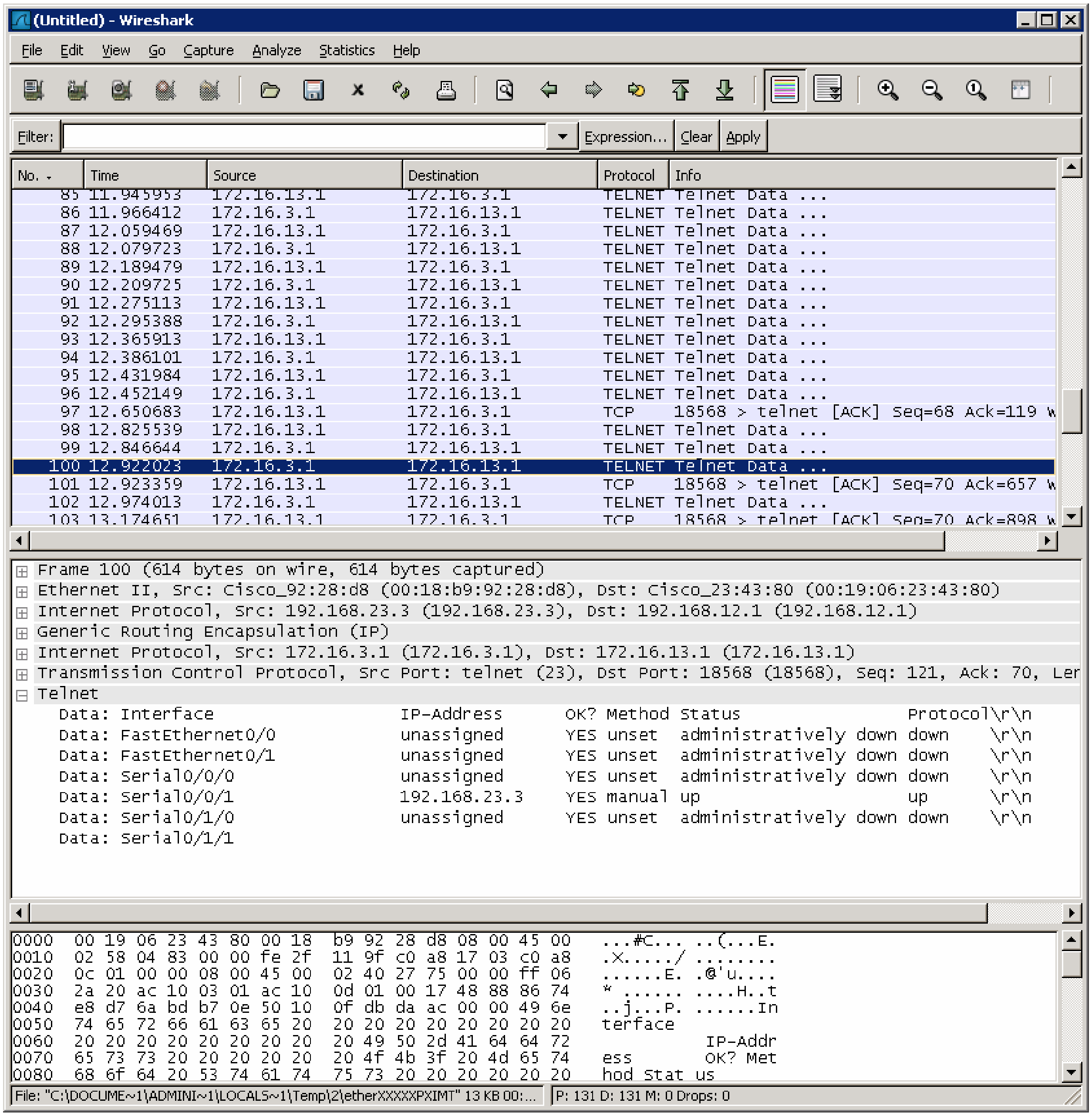
Do you need to use the **/source** attribute in the telnet command? Explain.

First, begin capturing packets using Wireshark. Then, begin the telnet session. Once you are connected to R3, issue a command or two and then log out. The packets will be routed through the tunnel interface towards the loopback on R3, so Wireshark will display the GRE packets.

```
R1# telnet 172.16.3.1
Trying 172.16.3.1 ... Open


User Access Verification

Password:
R3> enable
Password:
R3# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            unassigned      YES unset  administratively down
down
```

```
FastEthernet0/1            unassigned      YES unset  administratively down
down
Serial0/0/0                unassigned      YES unset  administratively down
down
Serial0/0/1                192.168.23.3    YES manual up                    up
Serial0/1/0                unassigned      YES unset  administratively down
down
Serial0/1/1                unassigned      YES unset  administratively down
down
Loopback0                  172.16.3.1      YES manual up                    up
Tunnel0                    172.16.13.3     YES manual up                    up
R3# exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

Now, take a look at the output. Notice that Wireshark is smart enough to classify these packets as telnet traffic, even though the actual packets are GRE. Looking in the middle pane in Wireshark, it will show the multiple layers of encapsulation, including the GRE information. Notice that since you disabled encryption, you can easily read the plaintext strings of the telnet session in Wireshark.



```
No. -   Time        Source         Destination    Protocol Info
  85 11.945953   172.16.13.1    172.16.3.1     TELNET Telnet Data ...
  86 11.966412   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
  87 12.059469   172.16.13.1    172.16.3.1     TELNET Telnet Data ...
  88 12.079723   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
  89 12.189479   172.16.13.1    172.16.3.1     TELNET Telnet Data ...
  90 12.209725   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
  91 12.275113   172.16.13.1    172.16.3.1     TELNET Telnet Data ...
  92 12.295388   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
  93 12.365913   172.16.13.1    172.16.3.1     TELNET Telnet Data ...
  94 12.386101   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
  95 12.431984   172.16.13.1    172.16.3.1     TELNET Telnet Data ...
  96 12.452149   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
  97 12.650683   172.16.13.1    172.16.3.1     TCP    18568 > telnet [ACK] Seq=68 Ack=119 W
  98 12.825539   172.16.13.1    172.16.3.1     TELNET Telnet Data ...
  99 12.846644   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
 100 12.922023   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
 101 12.923359   172.16.13.1    172.16.3.1     TCP    18568 > telnet [ACK] Seq=70 Ack=657 W
 102 12.974013   172.16.3.1     172.16.13.1    TELNET Telnet Data ...
 103 13.174651   172.16.13.1    172.16.3.1     TCP    18568 > telnet [ACK] Seq=70 Ack=898 W
```

```
⊞ Frame 100 (614 bytes on wire, 614 bytes captured)
⊞ Ethernet II, Src: Cisco_92:28:d8 (00:18:b9:92:28:d8), Dst: Cisco_23:43:80 (00:19:06:23:43:80)
⊞ Internet Protocol, Src: 192.168.23.3 (192.168.23.3), Dst: 192.168.12.1 (192.168.12.1)
⊞ Generic Routing Encapsulation (IP)
⊞ Internet Protocol, Src: 172.16.3.1 (172.16.3.1), Dst: 172.16.13.1 (172.16.13.1)
⊞ Transmission Control Protocol, Src Port: telnet (23), Dst Port: 18568 (18568), Seq: 121, Ack: 70, Ler
⊟ Telnet
    Data: Interface                  IP-Address      OK? Method Status                Protocol\r\n
    Data: FastEthernet0/0            unassigned      YES unset  administratively down down    \r\n
    Data: FastEthernet0/1            unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/0/0                unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/0/1                192.168.23.3    YES manual up                    up      \r\n
    Data: Serial0/1/0                unassigned      YES unset  administratively down down    \r\n
    Data: Serial0/1/1
```

```
0000  00 19 06 23 43 80 00 18  b9 92 28 d8 08 00 45 00   ...#C... ..(...E.
0010  02 58 04 83 00 00 fe 2f  11 9f c0 a8 17 03 c0 a8   .X...../ ........
0020  0c 01 00 00 08 00 45 00  02 40 27 75 00 00 ff 06   ......E. .@'u....
0030  2a 20 ac 10 03 01 ac 10  0d 01 00 17 48 88 86 74   * ...... ....H..t
0040  e8 d7 6a bd b7 0e 50 10  0f db da ac 00 00 49 6e   ..j...P. ......In
0050  74 65 72 66 61 63 65 20  20 20 20 20 20 20 20 20   terface
0060  20 20 20 20 20 20 20 20  20 49 50 2d 41 64 64 72            IP-Addr
0070  65 73 73 20 20 20 20 20  20 4f 4b 3f 20 4d 65 74   ess      OK? Met
0080  68 6f 64 20 53 74 61 74  75 73 20 20 20 20 20 20   hod Stat us
```

```
File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\2\etherXXXXXPXIMT" 13 KB 00:...   P: 131 D: 131 M: 0 Drops: 0
```

Figure 11-1: Detailed Packet Data on Telnet String Sent From R1

Based on this output, you can see how easy it is for someone who is in the path of sensitive data to view unencrypted or clear text traffic.

Now, you will reapply the cryptography settings on R1 and R3 and begin a telnet session from R1 to R3 as before.

Begin by reapplying the crypto maps you removed earlier on R1 and R3.

```
R1(config)# interface fastethernet 0/0
R1(config-if)# crypto map mymap
```

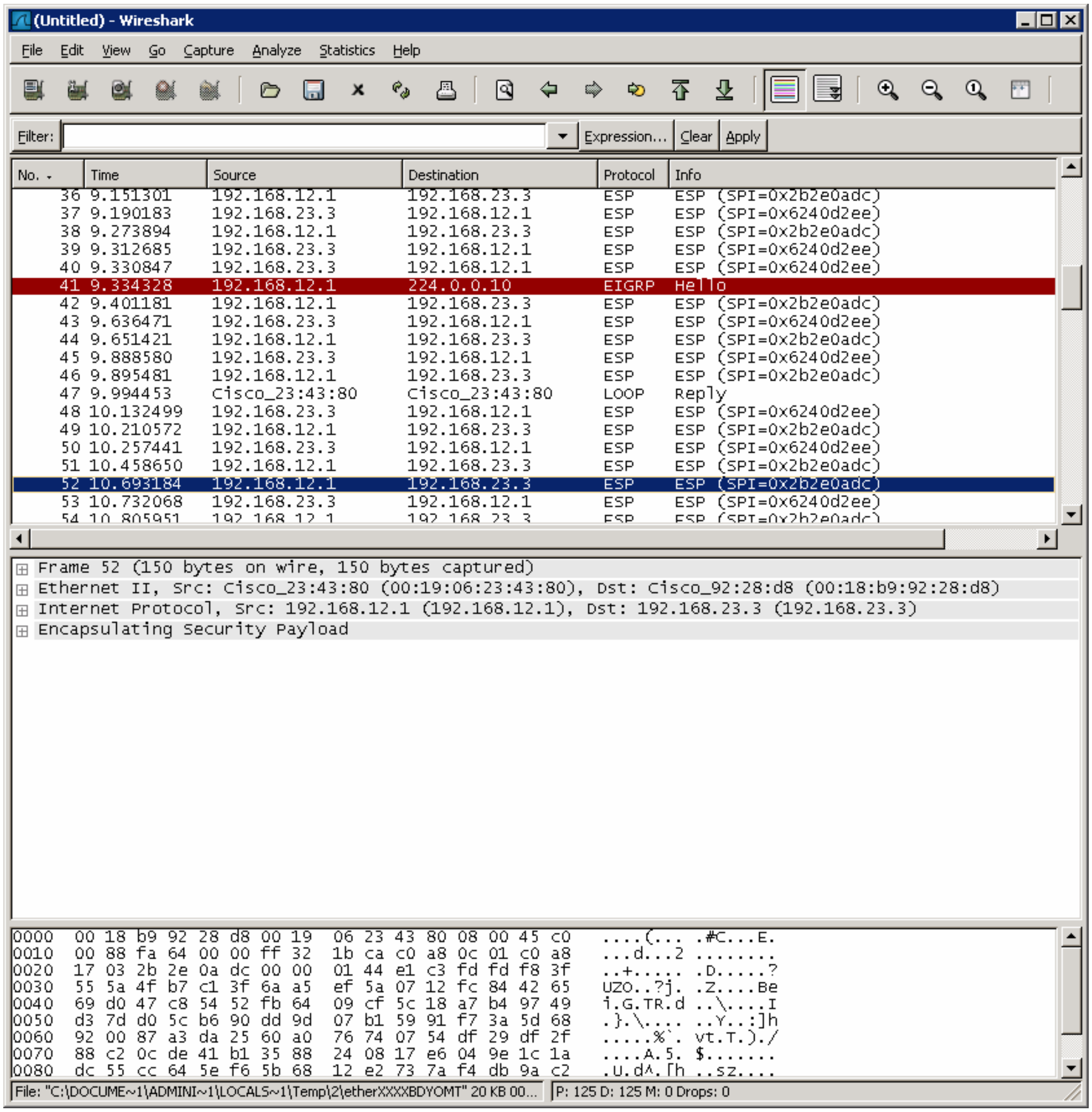
```
R3(config)# interface serial0/0/1
R3(config-if)# crypto map mymap
```

Start the packet capturing again in Wireshark, and then issue the same telnet sequence that you did previously.

```
R1# telnet 172.16.3.1
Trying 172.16.3.1 ... Open


User Access Verification

Password:
R3> enable
Password:
R3# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            unassigned      YES unset  administratively down
down
FastEthernet0/1            unassigned      YES unset  administratively down
down
Serial0/0/0                unassigned      YES unset  administratively down
down
Serial0/0/1                192.168.23.3    YES manual up                    up
Serial0/1/0                unassigned      YES unset  administratively down
down
Serial0/1/1                unassigned      YES unset  administratively down
down
Loopback0                  172.16.3.1      YES manual up                    up
Tunnel0                    172.16.13.3     YES manual up                    up
R3#exit

[Connection to 172.16.3.1 closed by foreign host]
R1#
```

End your Wireshark capture when you are finished with the telnet session.

As far as the user is concerned, the telnet session seems the same with and without encryption. However, the packet capture from Wireshark shows that the VPN is actively encapsulating and encrypting packets.



```
No. -  Time       Source          Destination     Protocol Info
   36 9.151301   192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   37 9.190183   192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   38 9.273894   192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   39 9.312685   192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   40 9.330847   192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   41 9.334328   192.168.12.1    224.0.0.10      EIGRP    Hello
   42 9.401181   192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   43 9.636471   192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   44 9.651421   192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   45 9.888580   192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   46 9.895481   192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   47 9.994453   Cisco_23:43:80  Cisco_23:43:80  LOOP     Reply
   48 10.132499  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   49 10.210572  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   50 10.257441  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   51 10.458650  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   52 10.693184  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
   53 10.732068  192.168.23.3    192.168.12.1    ESP      ESP (SPI=0x6240d2ee)
   54 10.805951  192.168.12.1    192.168.23.3    ESP      ESP (SPI=0x2b2e0adc)
```

```
⊞ Frame 52 (150 bytes on wire, 150 bytes captured)
⊞ Ethernet II, Src: Cisco_23:43:80 (00:19:06:23:43:80), Dst: Cisco_92:28:d8 (00:18:b9:92:28:d8)
⊞ Internet Protocol, Src: 192.168.12.1 (192.168.12.1), Dst: 192.168.23.3 (192.168.23.3)
⊞ Encapsulating Security Payload
```

```
0000  00 18 b9 92 28 d8 00 19  06 23 43 80 08 00 45 c0   ....(... .#C...E.
0010  00 88 fa 64 00 00 ff 32  1b ca c0 a8 0c 01 c0 a8   ...d...2 ........
0020  17 03 2b 2e 0a dc 00 00  01 44 e1 c3 fd fd f8 3f   ..+..... .D.....?
0030  55 5a 4f b7 c1 3f 6a a5  ef 5a 07 12 fc 84 42 65   UZO..?j. .Z....Be
0040  69 d0 47 c8 54 52 fb 64  09 cf 5c 18 a7 b4 97 49   i.G.TR.d ..\....I
0050  d3 7d d0 5c b6 90 dd 9d  07 b1 59 91 f7 3a 5d 68   .}.\.... ..Y..:]h
0060  92 00 87 a3 da 25 60 a0  76 74 07 54 df 29 df 2f   .....%`. vt.T.)./
0070  88 c2 0c de 41 b1 35 88  24 08 17 e6 04 9e 1c 1a   ....A.5. $.......
0080  dc 55 cc 64 5e f6 5b 68  12 e2 73 7a f4 db 9a c2   .U.d^.[h ..sz....
```

File: "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\2\etherXXXXBDYOMT" 20 KB 00... | P: 125 D: 125 M: 0 Drops: 0

Figure 11-2: Detailed Packet Data on Encrypted Telnet String Sent From R1

Notice that the protocol is not telnet (TCP port 23), but the Encapsulating Security Protocol (ESP, IP protocol number 50). Remember, all traffic here matches the IPSec access list.

Also, notice that the source and destination are not the actual source and destination of the addresses participating in this telnet conversation. Rather, they are the endpoints of the VPN.

Finally, and most important, if you look at the contents of these packets in Wireshark, no matter how you try to format or filter them, you will not be able to see what data was originally inside.

The encryption suite provided by IPSec successfully secures data through authentication, encryption, and data-integrity services.

Final Configurations

```
R1# show run
!
hostname R1
!
crypto isakmp policy 10
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.23.3
!
crypto ipsec transform-set mytrans ah-sha-hmac esp-aes 256 esp-sha-hmac
!
crypto map mymap 10 ipsec-isakmp
 set peer 192.168.23.3
 set transform-set mytrans
 match address 101
!
interface Tunnel0
 ip address 172.16.13.1 255.255.255.0
 tunnel source FastEthernet0/0
 tunnel destination 192.168.23.3
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 crypto map mymap
 no shutdown
!
router eigrp 1
 network 192.168.12.0
 no auto-summary
!
router eigrp 2
 network 172.16.0.0
 no auto-summary
!
access-list 101 permit gre host 192.168.12.1 host 192.168.23.3
end

R2# show run
hostname R2
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 192.168.12.0
 network 192.168.23.0
 no auto-summary
```

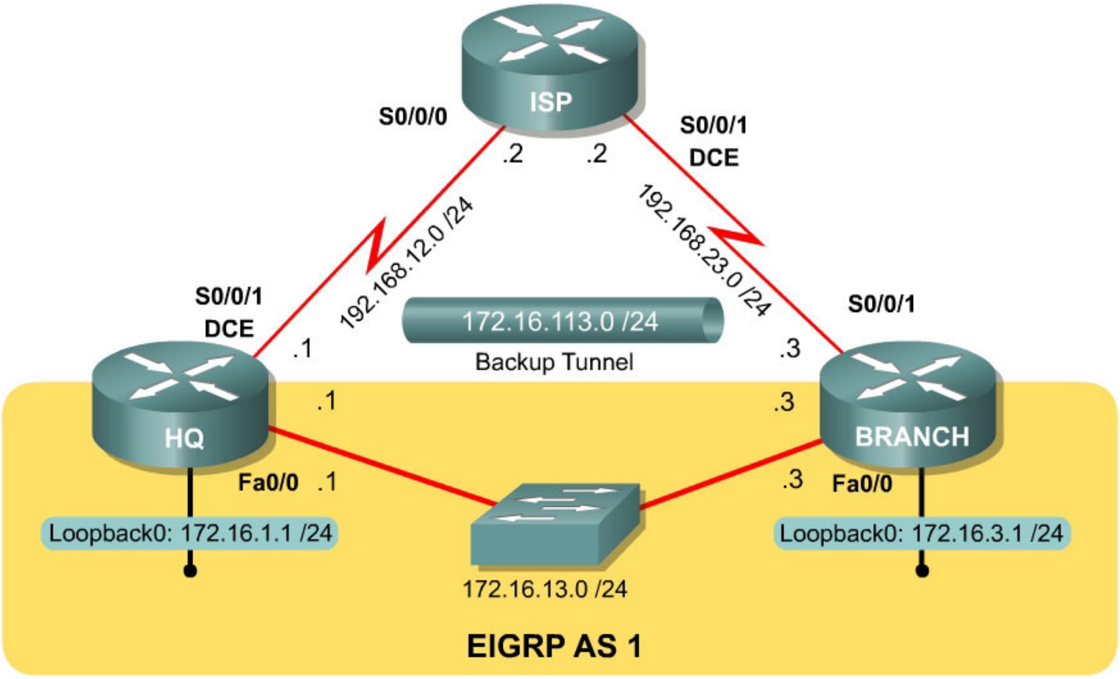
```
!
end

R3# show run
hostname R3
!
enable secret 5 $1$kkTj$cIYDuP2yz3vA1ARGVwxd11
!
crypto isakmp policy 10
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.12.1
!
crypto ipsec transform-set mytrans ah-sha-hmac esp-aes 256 esp-sha-hmac
!
crypto map mymap 10 ipsec-isakmp
 set peer 192.168.12.1
 set transform-set mytrans
 match address 101
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Tunnel0
 ip address 172.16.13.3 255.255.255.0
 tunnel source Serial0/0/1
 tunnel destination 192.168.12.1
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 crypto map mymap
 no shutdown
!
router eigrp 1
 network 192.168.23.0
 no auto-summary
!
router eigrp 2
 network 172.16.0.0
 no auto-summary
!
access-list 101 permit gre host 192.168.23.3 host 192.168.12.1
!
line vty 0 4
 password cisco
 login
end
```

Lab 3.8 Configuring IPsec VTIs

Learning Objectives

- Configure EIGRP on a router
- Configure an IPsec Virtual Tunnel Interface
- Configure the VTI to be used for backup purposes only

Topology Diagram



Scenario

The headquarters and branch office routers of International Travel Agency are connected over a leased line, which they receive as an Ethernet connection. They also both have slower, serial links connecting them to the Internet. This lab will cover configuring an IPsec Virtual Tunnel Interface, or VTI, to be used as a backup connection, running over the public internet.

If you have previously completed Lab 3.7 in which you created secure GRE tunnels, the results are similar; however, this newer method is the preferred method. If you are running an older IOS release, this feature may not be supported. It is highly recommended that you have previously configured GRE tunnels and IPsec VPNs to understand the commands being used in this lab.

Step 1: Configure Addressing

Configure the loopback interfaces with the addresses shown in the diagram. Also configure the serial interfaces shown in the diagram. Set the clock rates on the appropriate interfaces and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet with the **ping** command. Do not set up the tunnel interface until Step 7.

```
HQ# configure terminal
HQ(config)# interface loopback 0
HQ(config-if)# ip address 172.16.1.1 255.255.255.0
HQ(config-if)# interface fastethernet 0/0
HQ(config-if)# ip address 172.16.13.1 255.255.255.0
HQ(config-if)# no shutdown
HQ(config-if)# interface serial 0/0/0
HQ(config-if)# ip address 192.168.12.1 255.255.255.0
HQ(config-if)# clockrate 64000
HQ(config-if)# no shutdown

ISP# configure terminal
ISP(config-if)# interface serial 0/0/0
ISP(config-if)# ip address 192.168.12.2 255.255.255.0
ISP(config-if)# no shutdown
ISP(config-if)# interface serial 0/0/1
ISP(config-if)# ip address 192.168.23.2 255.255.255.0
ISP(config-if)# clockrate 64000
ISP(config-if)# no shutdown

BRANCH# configure terminal
BRANCH(config)# interface loopback 0
BRANCH(config-if)# ip address 172.16.3.1 255.255.255.0
BRANCH(config-if)# interface fastethernet 0/0
BRANCH(config-if)# ip address 172.16.13.3 255.255.255.0
BRANCH(config-if)# no shutdown
BRANCH(config-if)# interface serial 0/0/1
BRANCH(config-if)# ip address 192.168.23.3 255.255.255.0
BRANCH(config-if)# no shutdown
```

Step 2: Configure EIGRP AS 1

Configure EIGRP for AS1 on HQ and BRANCH. Add the whole major network 172.16.0.0 and disable automatic summarization. ISP will not participate in this routing process.

```
HQ(config)# router eigrp 1
HQ(config-router)# no auto-summary
HQ(config-router)# network 172.16.0.0

BRANCH(config)# router eigrp 1
BRANCH(config-router)# no auto-summary
BRANCH(config-router)# network 172.16.0.0
```

EIGRP neighbor adjacencies should form. If not, troubleshoot by checking your interface configuration, EIGRP configuration, and physical connectivity.

Step 3: Configure Static Routing

On the HQ and BRANCH routers, add a static default route pointing towards ISP through the serial interfaces. This will simulate a connection to the internet. Remember that this route will not be preferred over routes learned via Enhanced Interior Gateway Routing Protocol (EIGRP), because the EIGRP routes will be more specific and therefore preferred. We also do not need to configure any static routes on ISP because it has connected routes to route between the 192.168.X.0/24 networks HQ and BRANCH.

```
HQ(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.2

BRANCH(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2
```

Now everything is in place for configuring the backup tunnel to run over the Internet. Before starting, be sure that both HQ and BRANCH can successfully ping each other’s Internet-facing interface (which should be routed via ISP), and the other’s loopback interface. Compare the output of the **show ip route** command to the following output.

```
HQ# ping 192.168.23.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms

HQ# ping 172.16.3.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

HQ# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.12.2 to network 0.0.0.0

C    192.168.12.0/24 is directly connected, Serial0/0/0
     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.13.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, Loopback0
D       172.16.3.0 [90/156160] via 172.16.13.3, 00:01:56, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 192.168.12.2

BRANCH# ping 192.168.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
```

```
BRANCH# ping 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

BRANCH# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.23.2 to network 0.0.0.0

     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.13.0 is directly connected, FastEthernet0/0
D       172.16.1.0 [90/156160] via 172.16.13.1, 00:02:32, FastEthernet0/0
C       172.16.3.0 is directly connected, Loopback0
C    192.168.23.0/24 is directly connected, Serial0/0/1
S*   0.0.0.0/0 [1/0] via 192.168.23.2
```

Step 4: Create IKE Policies and Peers

Since you will be using Internet Security Association and Key Management Protocol (ISAKMP) with IPsec, configure Internet Key Exchange (IKE) policies and IKE peers on both tunnel endpoints. Create an IKE policy as shown in the following output. If your IOS image doesn’t support all of the settings, configure what you can. Just make sure your VPN settings match on both ends of the connection.

```
HQ(config)# crypto isakmp policy 10
HQ(config-isakmp)# authentication pre-share
HQ(config-isakmp)# encryption aes 256
HQ(config-isakmp)# hash sha
HQ(config-isakmp)# group 5
HQ(config-isakmp)# lifetime 3600

BRANCH(config)# crypto isakmp policy 10
BRANCH(config-isakmp)# authentication pre-share
BRANCH(config-isakmp)# encryption aes 256
BRANCH(config-isakmp)# hash sha
BRANCH(config-isakmp)# group 5
BRANCH(config-isakmp)# lifetime 3600
```

Which of the options ensures data confidentiality in the tunnel?

Which of the options ensures data integrity in the tunnel?

Which of the options controls the strength of keying information during the ISAKMP exchange?

Now, configure each peer using the key “cisco” for ISAKMP.

```
HQ(config)# crypto isakmp key cisco address 192.168.23.3

BRANCH(config)# crypto isakmp key cisco address 192.168.12.1
```

Step 5: Create IPsec Transform Sets

On both endpoint routers, create an IPsec transform set using the settings that follow. An IPsec profile is a set of parameters used to negotiate an IPsec VPN tunnel between two endpoints, including data encapsulation, authentication, and integrity.

```
HQ(config)# crypto ipsec transform-set mytrans esp-aes 256 esp-sha-hmac ah-
sha-hmac
HQ(cfg-crypto-trans)# exit
HQ(config)#

BRANCH(config)# crypto ipsec transform-set mytrans esp-aes 256 esp-sha-hmac
ah-sha-hmac
BRANCH(cfg-crypto-trans)# exit
BRANCH(config)#
```

If your routers do not support these settings, use whichever settings you can. Just keep it consistent on both routers.

Step 6: Create an IPsec Profile

Now that you have created the transform set, create an IPsec profile. An IPsec profile is similar to a crypto map, in that it binds the set of independent parameters and associations negotiated in IKE Phase I with the transform sets for Phase II. It also creates a structure that can, like a crypto map, be applied to an interface. However, an IPsec profile differs from a crypto map in that there is no **match** clause, only **set** statements, because it is applied to an interface. All traffic sent into or out of the tunnel interface will be encrypted.

To begin configuring an IPsec profile, use the global configuration command **crypto ipsec profile** *name*. In this configuration, use “myprof” as the profile name.

```
HQ(config)# crypto ipsec profile myprof
```

In the IPsec profile configuration submode, type **set ?** to find out various attributes you can set with an IPsec profile.

```
HQ(ipsec-profile)# set ?
  identity               Identity restriction.
  isakmp-profile         Specify isakmp Profile
  pfs                    Specify pfs settings
  security-association  Security association parameters
  transform-set          Specify list of transform sets in priority order
```

As you can see, the **set** parameters are very similar to the parameters you can **set** in a crypto map. In this case, we will only set the transform set to the transform set we configured earlier. Apply the same configuration to BRANCH.

```
HQ(ipsec-profile)# set transform-set mytrans

BRANCH(config)# crypto ipsec profile myprof
BRANCH(ipsec-profile)# set transform-set mytrans
```

Step 7: Create the IPsec VTI

HQ and BRANCH will need to have a tunnel interface on them, which will have the standard tunnel IP address and source and destination. The source and destination should be the serial connections to ISP. In addition to this, you should change the mode of the tunnel using the interface level command **tunnel mode** *mode*, and in this case, the mode will be ipsec with IPv4. To apply the IPsec profile created earlier, use the interface level command **tunnel protection ipsec profile** *name*.

```
HQ(config)# interface tunnel 0
HQ(config-if)# ip address 172.16.113.1 255.255.255.0
HQ(config-if)# tunnel source serial 0/0/0
HQ(config-if)# tunnel destination 192.168.23.3
HQ(config-if)# tunnel mode ipsec ipv4
HQ(config-if)# tunnel protection ipsec profile myprof

BRANCH(config)# interface tunnel 0
BRANCH(config-if)# ip address 172.16.113.3 255.255.255.0
BRANCH(config-if)# tunnel source serial 0/0/1
BRANCH(config-if)# tunnel destination 192.168.12.1
BRANCH(config-if)# tunnel mode ipsec ipv4
BRANCH(config-if)# tunnel protection ipsec profile myprof
```

If the endpoints of the tunnel are the serial interfaces facing ISP, then when will routed traffic to the private 172.16.0.0/24 network be sent through the tunnel?

Use the **show crypto ipsec sa** command to verify that the packet counters are incrementing with EIGRP hello packets across the tunnel. Also verify that the EIGRP neighbor adjacency is up with the **show ip eigrp neighbors** command.

```
HQ# show crypto ipsec sa

interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 14, #pkts encrypt: 14, #pkts digest: 14
    #pkts decaps: 16, #pkts decrypt: 16, #pkts verify: 16
<OUTPUT OMITTED>

HQ# show crypto ipsec sa

interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 192.168.12.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   current_peer 192.168.23.3 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 15, #pkts encrypt: 15, #pkts digest: 15
    #pkts decaps: 17, #pkts decrypt: 17, #pkts verify: 17
<OUTPUT OMITTED>

HQ# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.113.3            Tu0               11 00:03:40  118  5000  0  15
0   172.16.13.3             Fa0/0             10 01:04:20    1   200  0  13

BRANCH# show crypto ipsec sa

interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 192.168.23.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   current_peer 192.168.12.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 40, #pkts encrypt: 40, #pkts digest: 40
    #pkts decaps: 39, #pkts decrypt: 39, #pkts verify: 39
<OUTPUT OMITTED>

BRANCH# show crypto ipsec sa

interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 192.168.23.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   current_peer 192.168.12.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 41, #pkts encrypt: 41, #pkts digest: 41
    #pkts decaps: 41, #pkts decrypt: 41, #pkts verify: 41
<OUTPUT OMITTED>

BRANCH# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.113.1            Tu0               11 00:03:48  118  5000  0  12
0   172.16.13.1             Fa0/0             12 01:04:28  333  1998  0  11
```

Step 8: Verify Proper EIGRP Behavior

On HQ, issue a **show ip route** command and make sure that the preferred route to the BRANCH loopback is through the leased line (FastEthernet0/0).

```
HQ# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.12.2 to network 0.0.0.0

C    192.168.12.0/24 is directly connected, Serial0/0/0
     172.16.0.0/24 is subnetted, 4 subnets
C       172.16.13.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, Loopback0
D       172.16.3.0 [90/156160] via 172.16.13.3, 00:13:29, FastEthernet0/0
C       172.16.113.0 is directly connected, Tunnel0
S*   0.0.0.0/0 [1/0] via 192.168.12.2
```

Now, shut down the leased line connection on BRANCH.

```
BRANCH(config)# interface fastethernet 0/0
BRANCH(config-if)# shutdown
```

On HQ, try issuing a **show ip route** command again after the neighbor adjacency expires.

```
*Jan 23 02:14:17.931: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.3
(FastEthernet0/0) is down: holding time expired

HQ# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.12.2 to network 0.0.0.0
```

```
C    192.168.12.0/24 is directly connected, Serial0/0/0
     172.16.0.0/24 is subnetted, 4 subnets
C       172.16.13.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, Loopback0
D       172.16.3.0 [90/297372416] via 172.16.113.3, 00:00:44, Tunnel0
C       172.16.113.0 is directly connected, Tunnel0
S*   0.0.0.0/0 [1/0] via 192.168.12.2
```

Shutting down the Fast Ethernet interface on BRANCH simulates the leased line being disconnected. As you can see, the network reconverges to use the protected tunnel through the ISP router. Perform another **traceroute** to verify that the path now goes through the tunnel.

What happens to IP traffic passing through the tunnel?

Of course, this transport path is slower than the leased line, and is only preferred as a temporary backup, not a permanent solution. Open the FastEthernet0/0 interface that you shutdown earlier, and verify on HQ that the transit path is back to the way it was.

```
BRANCH(config)# interface fastethernet 0/0
BRANCH(config-if)# no shutdown

*Jan 23 02:18:56.959: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.3
(FastEthernet0/0) is up: new adjacency

HQ# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.12.2 to network 0.0.0.0

C    192.168.12.0/24 is directly connected, Serial0/0/0
     172.16.0.0/24 is subnetted, 4 subnets
C       172.16.13.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, Loopback0
D       172.16.3.0 [90/156160] via 172.16.13.3, 00:00:29, FastEthernet0/0
C       172.16.113.0 is directly connected, Tunnel0
S*   0.0.0.0/0 [1/0] via 192.168.12.2
```

You can understand why EIGRP prefers the path through the Fast Ethernet network if you look at the default bandwidth and delay on the Fast Ethernet and Tunnel interfaces with the **show interfaces** *interface-type interface-number* command. Remember that EIGRP will prefer the path with the minimum composite metric of minimum path bandwidth and lowest total delay. As the output shows, there are radical differences in these attributes between Fast Ethernet and Tunnel interfaces. If you needed to change these to make the routes preferred in a certain way, you could modify the attributes with the interface level commands **bandwidth** *bandwidth* and **delay** *delay*, although these commands are outside the scope of this lab.

```
HQ# show interfaces fastethernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 0019.0623.4380 (bia 0019.0623.4380)
  Internet address is 172.16.13.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,

HQ# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.113.1/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
```

Final Configurations

```
HQ# show run
!
hostname HQ
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.23.3
!
crypto ipsec transform-set mytrans ah-sha-hmac esp-aes 256 esp-sha-hmac
!
crypto ipsec profile myprof
 set transform-set mytrans
!
interface Tunnel0
 ip address 172.16.113.1 255.255.255.0
 tunnel source Serial0/0/0
 tunnel destination 192.168.23.3
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile myprof
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 172.16.13.1 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 ip address 192.168.12.1 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
!
ip route 0.0.0.0 0.0.0.0 192.168.12.2
```

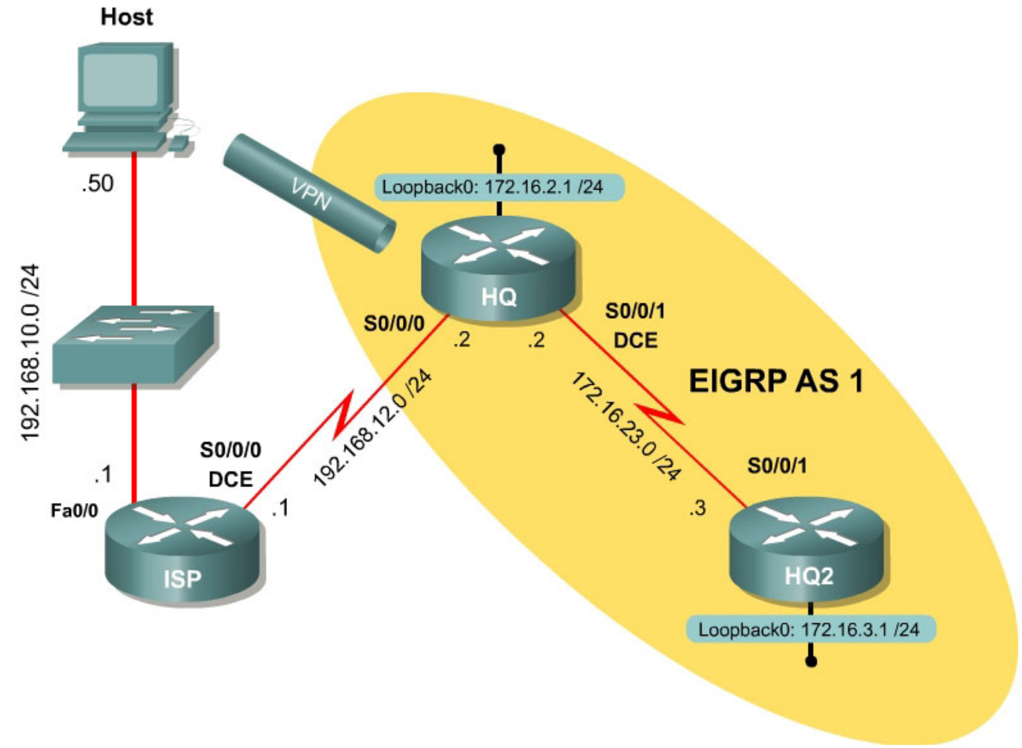
```
!
end

ISP# show run
!
hostname ISP
!
interface Serial0/0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 clock rate 64000
 no shutdown
!
end

BRANCH# show run
hostname BRANCH
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.12.1
!
crypto ipsec transform-set mytrans ah-sha-hmac esp-aes 256 esp-sha-hmac
!
crypto ipsec profile myprof
 set transform-set mytrans
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Tunnel0
 ip address 172.16.113.3 255.255.255.0
 tunnel source Serial0/0/1
 tunnel destination 192.168.12.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile myprof
!
interface FastEthernet0/0
 ip address 172.16.13.3 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
!
ip route 0.0.0.0 0.0.0.0 192.168.23.2
!
end
```

Lab 3.9 Configuring Easy VPN with SDM

Learning Objectives

- Configure EIGRP on a router
- Configure Easy VPN using SDM
- Install the Cisco VPN Client to a host
- Connect to the VPN using Cisco VPN client
- Verify VPN operation using SDM

Topology Diagram



Scenario

In this lab, you will set up Easy VPN using SDM for the International Travel Agency. The host will simulate an employee connecting from home over the Internet. The router ISP will simulate an Internet router representing the Internet connection for both the home user and the company headquarters.

Step 1: Configure Addressing

Configure the loopback interfaces with the addresses shown in the diagram. Also configure the serial interfaces shown in the diagram. Set the clockrate on the appropriate interfaces and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command. Do not set up the tunnel interface.

```
ISP# configure terminal
ISP(config)# interface fastethernet0/0
ISP(config-if)# ip address 192.168.10.1 255.255.255.0
ISP(config-if)# no shutdown
ISP(config-if)# interface serial 0/0/0
ISP(config-if)# ip address 192.168.12.1 255.255.255.0
ISP(config-if)# clockrate 64000
ISP(config-if)# no shutdown

HQ# configure terminal
HQ(config)# interface loopback 0
HQ(config-if)# ip address 172.16.2.1 255.255.255.0
HQ(config-if)# interface serial0/0/0
HQ(config-if)# ip address 192.168.12.2 255.255.255.0
HQ(config-if)# no shutdown
HQ(config-if)# interface serial 0/0/1
HQ(config-if)# ip address 172.16.23.2 255.255.255.0
HQ(config-if)# clockrate 64000
HQ(config-if)# no shutdown

HQ2# configure terminal
HQ2(config)# interface loopback 0
HQ2(config-if)# ip address 172.16.3.1 255.255.255.0
HQ2(config-if)# interface serial 0/0/1
HQ2(config-if)# ip address 172.16.23.3 255.255.255.0
HQ2(config-if)# no shutdown
```

Step 2: Configure EIGRP AS 1

Configure EIGRP for AS1 on HQ and HQ2. Add the entire 172.16.0.0/16 major network and disable automatic summarization. The router ISP will not participate in this routing process.

```
HQ(config)# router eigrp 1
HQ(config-router)# no auto-summary
HQ(config-router)# network 172.16.0.0

HQ2(config)# router eigrp 1
HQ2(config-router)# no auto-summary
HQ2(config-router)# network 172.16.0.0
```

An EIGRP neighbor adjacency should form between HQ and HQ2. If not, troubleshoot by checking your interface configuration, EIGRP configuration, and physical connectivity.

Step 3: Configure a Static Default Route

Since the router ISP represents a connection to the Internet, send all traffic whose destination network does not exist in the routing tables at company headquarters out this connection via a default route. This route can be statically created on HQ, but will need to be redistributed into EIGRP so HQ2 will learn the route too.

```
HQ(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.1
HQ(config)# router eigrp 1
HQ(config-router)# redistribute static
```

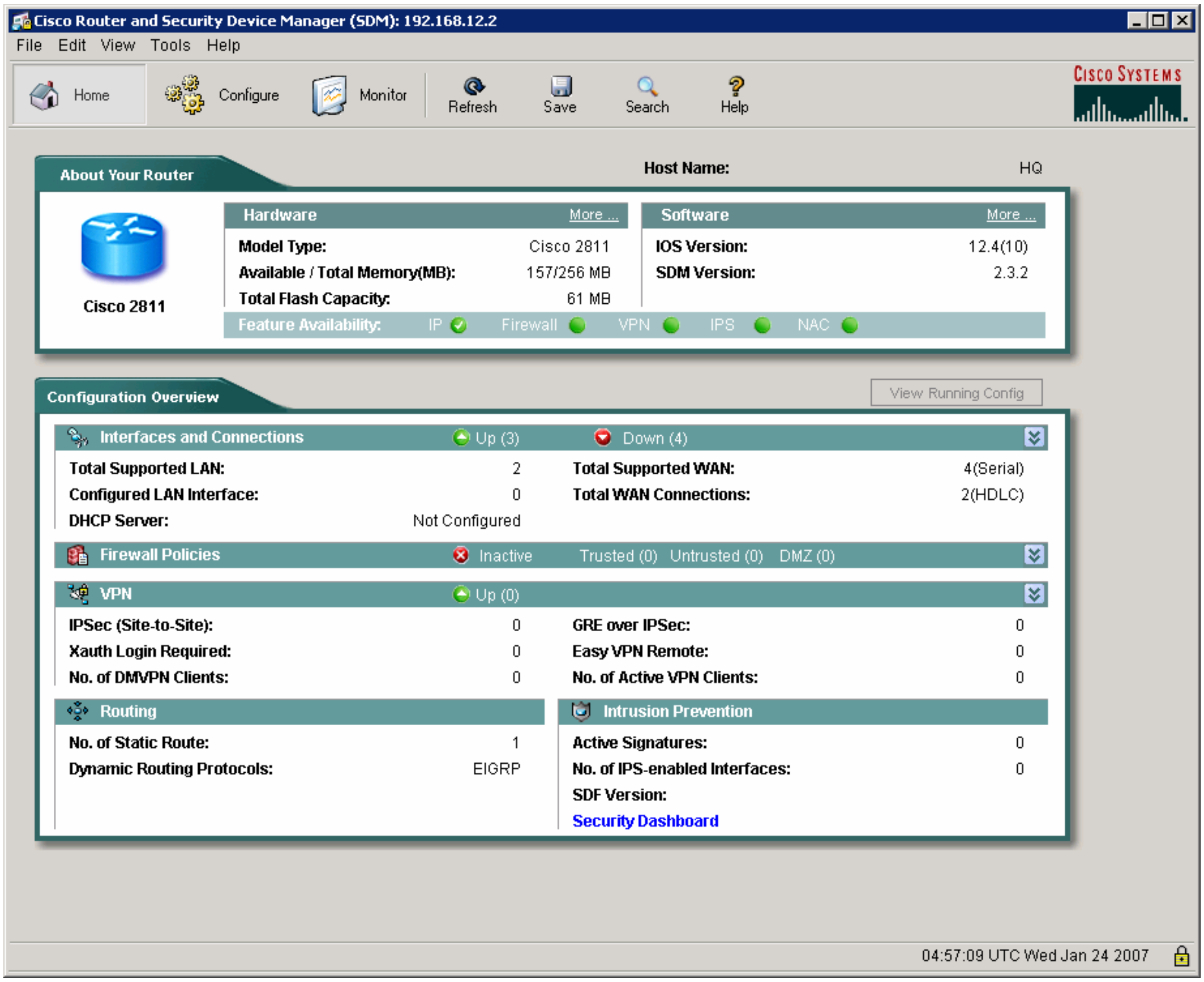
For which types of routes is it unnecessary to assign a default/seed metric when redistributing into EIGRP?

How else could you configure HQ to advertise the default route?

Step 4: Connect to HQ through SDM

Prepare HQ to allow connection and configuration via SDM as you did in Lab 3.1.

Configure the host to connect to HQ using SDM. Configure the host with the IP address shown in the topology diagram, and ensure that its default gateway is set to ISP so that traffic from the host to HQ will get routed properly. Remember that you should only be able to connect to HQ's outside interface (192.168.12.2) using SDM because the interfaces inside the EIGRP domain are not reachable from ISP and the PC. If you do not know how to configure the host IP address and connect using SDM, refer to Lab 3.1.



Host Name: HQ

Cisco 2811

Model Type: Cisco 2811
Available / Total Memory(MB): 157/256 MB
Total Flash Capacity: 61 MB

IOS Version: 12.4(10)
SDM Version: 2.3.2

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Interfaces and Connections Up (3) Down (4)

Total Supported LAN: 2
Configured LAN Interface: 0
DHCP Server: Not Configured
Total Supported WAN: 4(Serial)
Total WAN Connections: 2(HDLC)

Firewall Policies Inactive Trusted (0) Untrusted (0) DMZ (0)

VPN Up (0)

IPSec (Site-to-Site): 0
Xauth Login Required: 0
No. of DMVPN Clients: 0
GRE over IPSec: 0
Easy VPN Remote: 0
No. of Active VPN Clients: 0

Routing

No. of Static Route: 1
Dynamic Routing Protocols: EIGRP

Intrusion Prevention

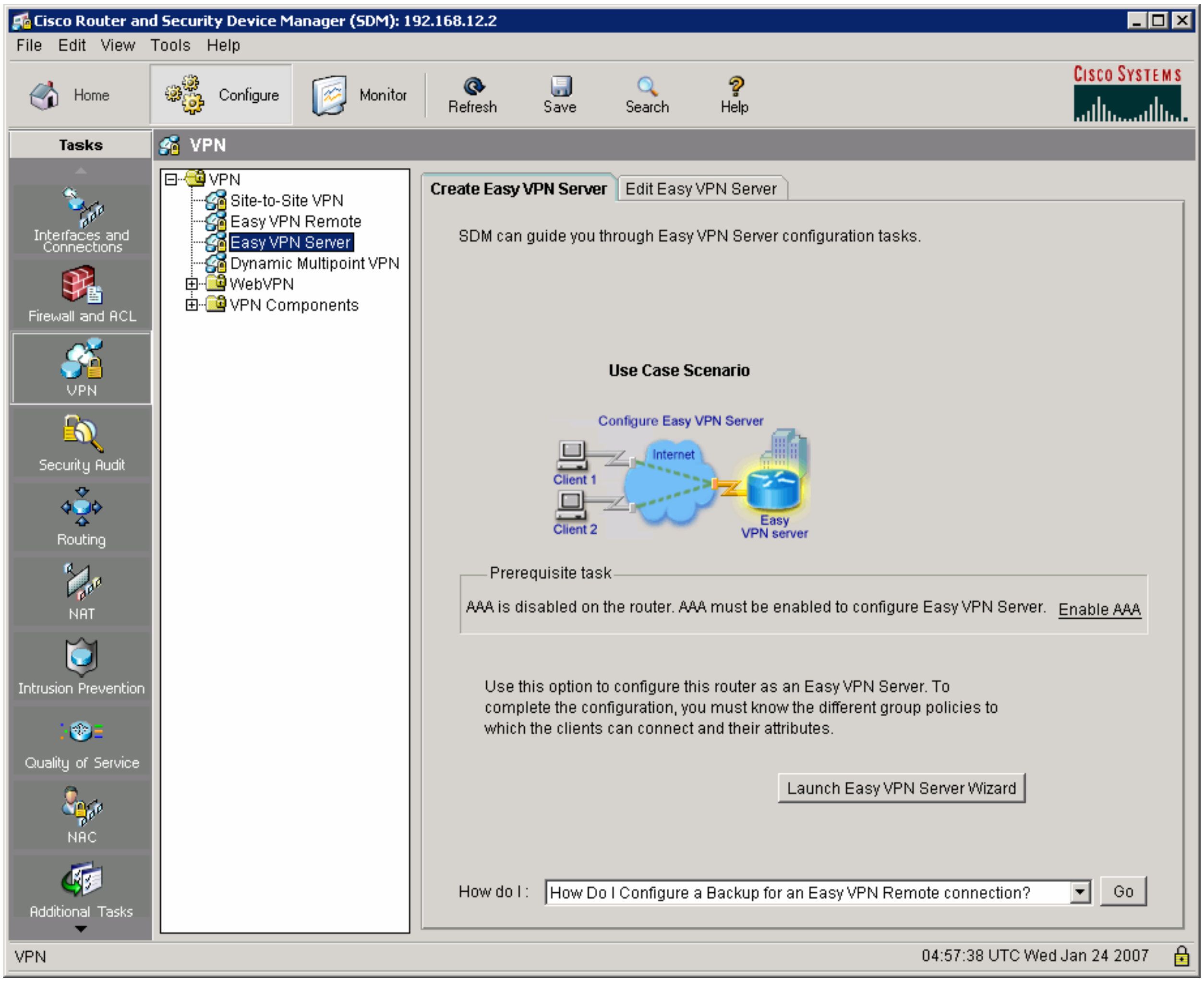
Active Signatures: 0
No. of IPS-enabled Interfaces: 0
SDF Version:

04:57:09 UTC Wed Jan 24 2007

Figure 4-1: SDM Home Screen

Step 5: Configure Easy VPN Server through SDM

Once you are at the SDM home screen for HQ, click the **Configure** icon at the top and choose **VPN** on the left side bar. Choose **Easy VPN Server** in the VPN types list. Notice that there is a prerequisite task to configure AAA. Click **Enable AAA** to allow SDM to fulfill this task for you.



SDM can guide you through Easy VPN Server configuration tasks.

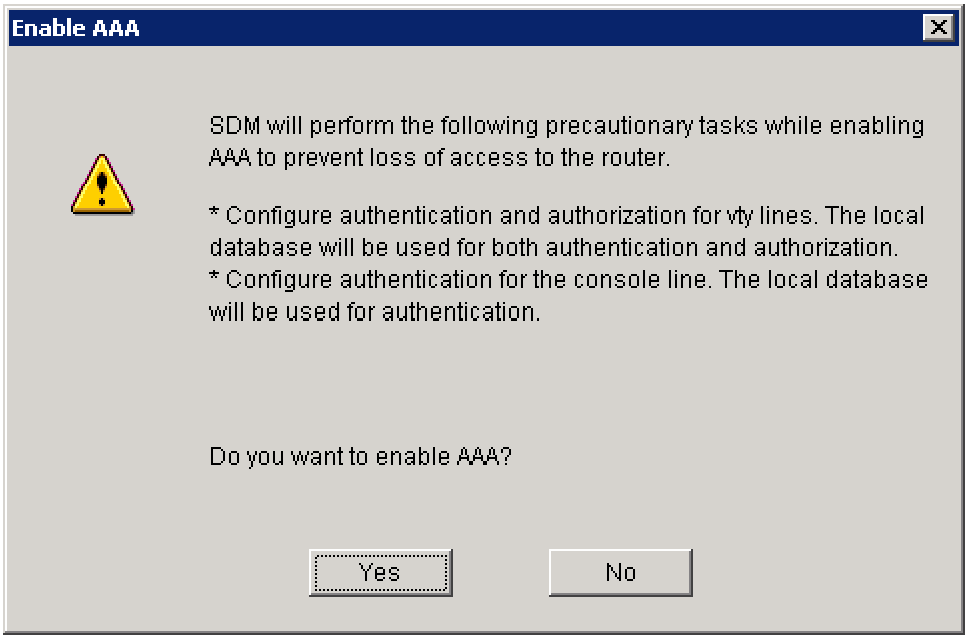
AAA is disabled on the router. AAA must be enabled to configure Easy VPN Server.

Use this option to configure this router as an Easy VPN Server. To complete the configuration, you must know the different group policies to which the clients can connect and their attributes.

How do I : How Do I Configure a Backup for an Easy VPN Remote connection?

Figure 5-1: Create Easy VPN Server Tab

SDM gives you a warning about the changes it will make in addition to enabling AAA (this is to prevent you from getting locked out of the router). When you understand the implications of acknowledgement, click **Yes** to continue. Note that now when accessing HQ you need to use a username/password pair configured on the router. You already have configured one for use with SDM, so you can reuse it.



Enable AAA

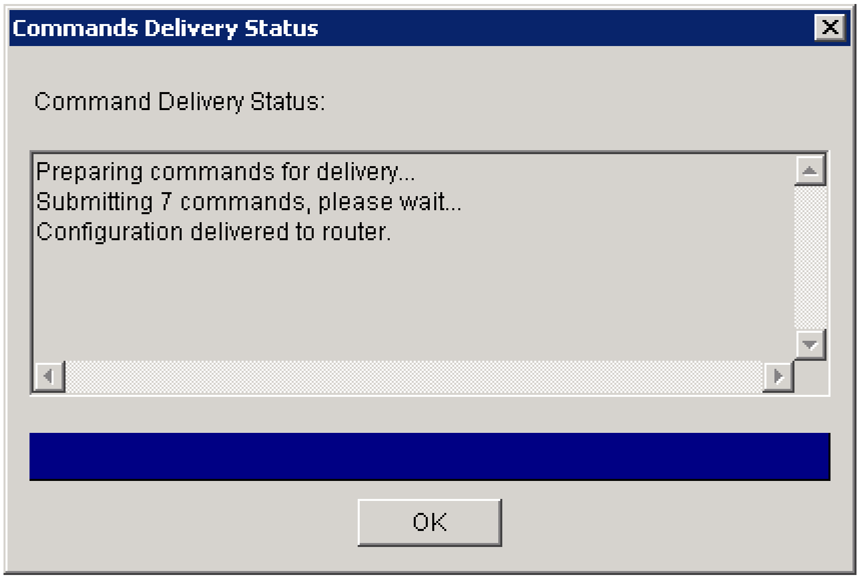
SDM will perform the following precautionary tasks while enabling AAA to prevent loss of access to the router.

* Configure authentication and authorization for vty lines. The local database will be used for both authentication and authorization.
* Configure authentication for the console line. The local database will be used for authentication.

Do you want to enable AAA?

Figure 5-2: AAA Configuration Prompt

Click **Yes** and the SDM will deliver the AAA commands to the router. Click **OK** when the delivery process is complete.



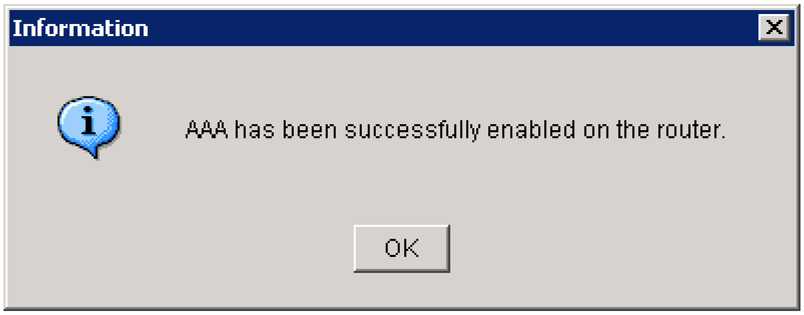
Commands Delivery Status

Command Delivery Status:

```
Preparing commands for delivery...
Submitting 7 commands, please wait...
Configuration delivered to router.
```

Figure 5-3: Command Delivery Progress Indicator

Once delivery is complete, SDM notifies you that enabling AAA was successful. Click **OK** to continue.

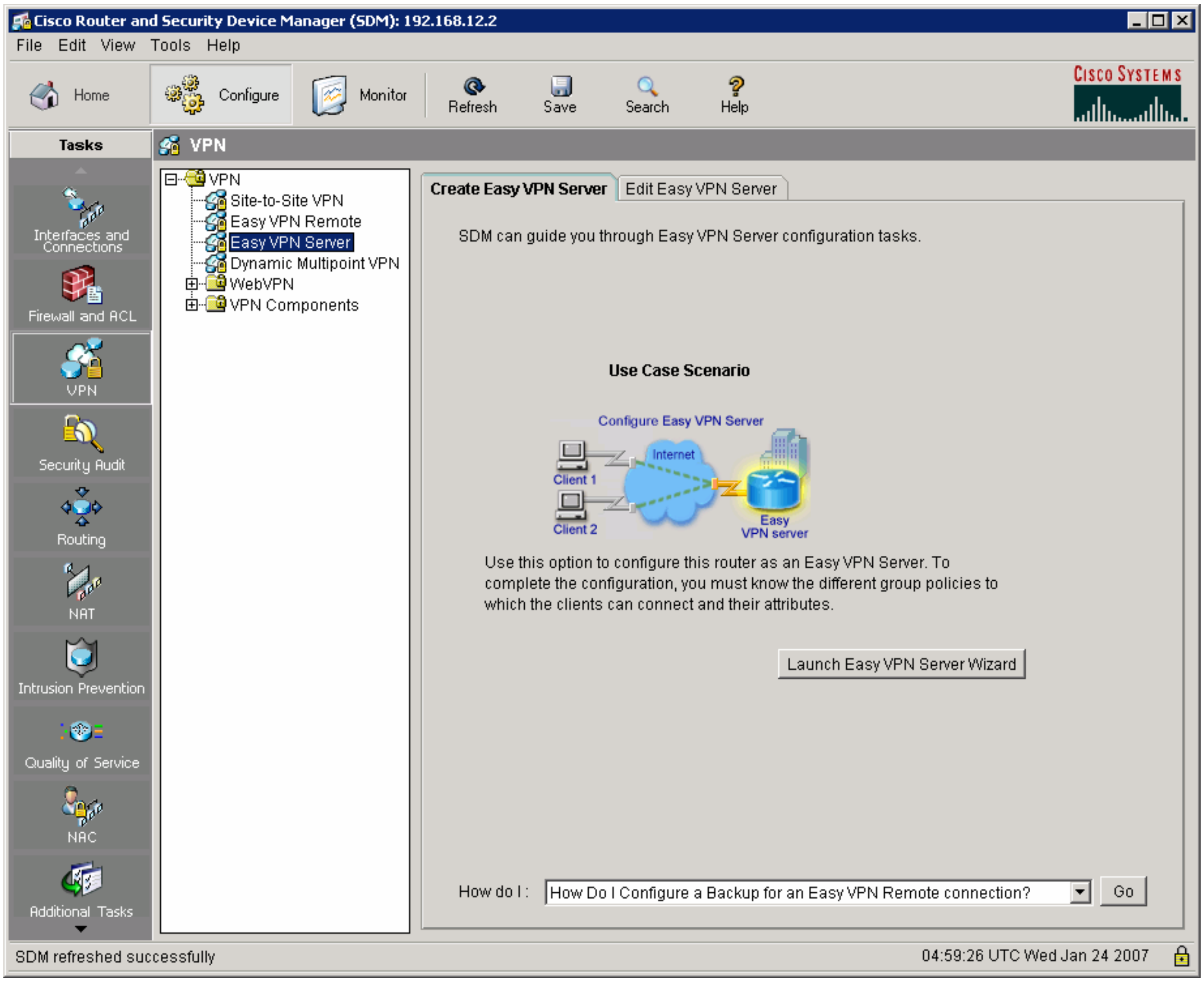


Information

AAA has been successfully enabled on the router.

Figure 5-4: Successful AAA Configuration Report

Now that AAA is enabled, you can start the Easy VPN Server Wizard by clicking the **Launch Easy VPN Server Wizard** button.



Cisco Router and Security Device Manager (SDM): 192.168.12.2

SDM can guide you through Easy VPN Server configuration tasks.

Use Case Scenario

Use this option to configure this router as an Easy VPN Server. To complete the configuration, you must know the different group policies to which the clients can connect and their attributes.

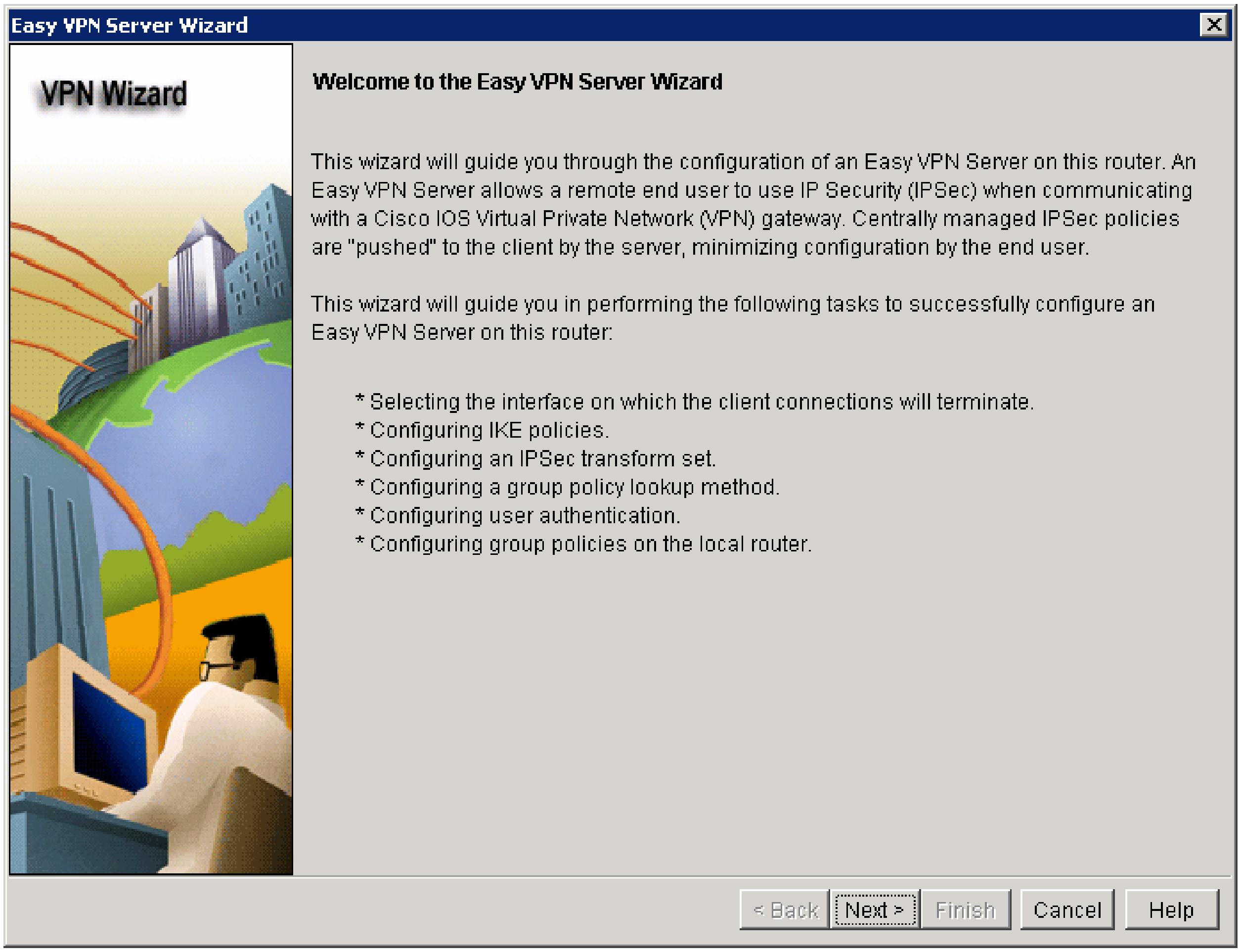
How do I : How Do I Configure a Backup for an Easy VPN Remote connection?

SDM refreshed successfully

04:59:26 UTC Wed Jan 24 2007

Figure 5-5: Create Easy VPN Server Tab

After reading the brief introduction to the wizard, click **Next**.



Easy VPN Server Wizard

Welcome to the Easy VPN Server Wizard

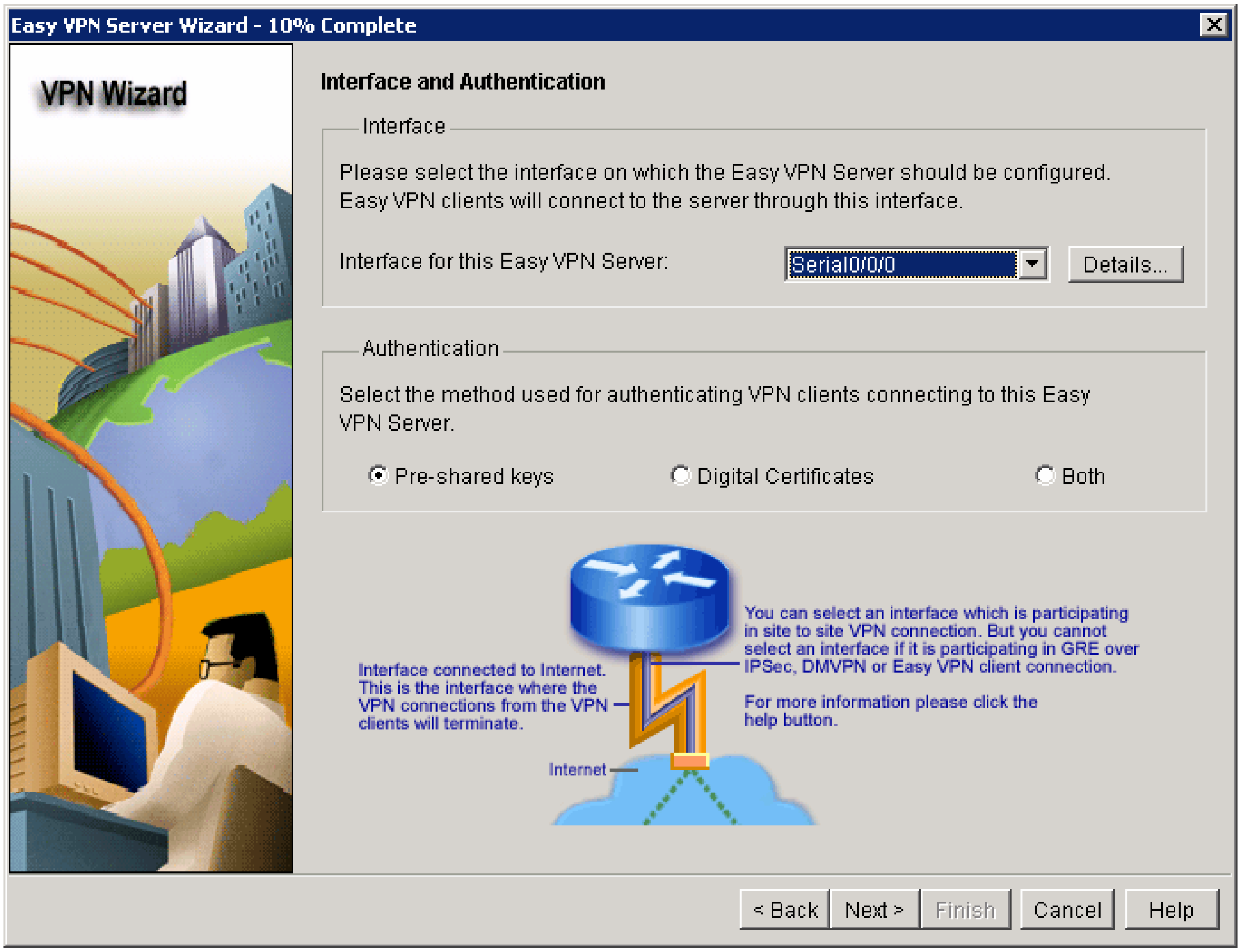
This wizard will guide you through the configuration of an Easy VPN Server on this router. An Easy VPN Server allows a remote end user to use IP Security (IPSec) when communicating with a Cisco IOS Virtual Private Network (VPN) gateway. Centrally managed IPSec policies are "pushed" to the client by the server, minimizing configuration by the end user.

This wizard will guide you in performing the following tasks to successfully configure an Easy VPN Server on this router:

* Selecting the interface on which the client connections will terminate.
* Configuring IKE policies.
* Configuring an IPSec transform set.
* Configuring a group policy lookup method.
* Configuring user authentication.
* Configuring group policies on the local router.

Figure 5-6: Easy VPN Server Wizard

Choose to run the Easy VPN Server on the ISP-facing interface of HQ. Use pre-shared keys as the authentication type since we will not be using a certificate server. Click **Next** when you are finished.



Easy VPN Server Wizard - 10% Complete

VPN Wizard

Interface and Authentication

Interface

Please select the interface on which the Easy VPN Server should be configured. Easy VPN clients will connect to the server through this interface.

Interface for this Easy VPN Server: Serial0/0/0

Authentication

Select the method used for authenticating VPN clients connecting to this Easy VPN Server.

● Pre-shared keys ○ Digital Certificates ○ Both

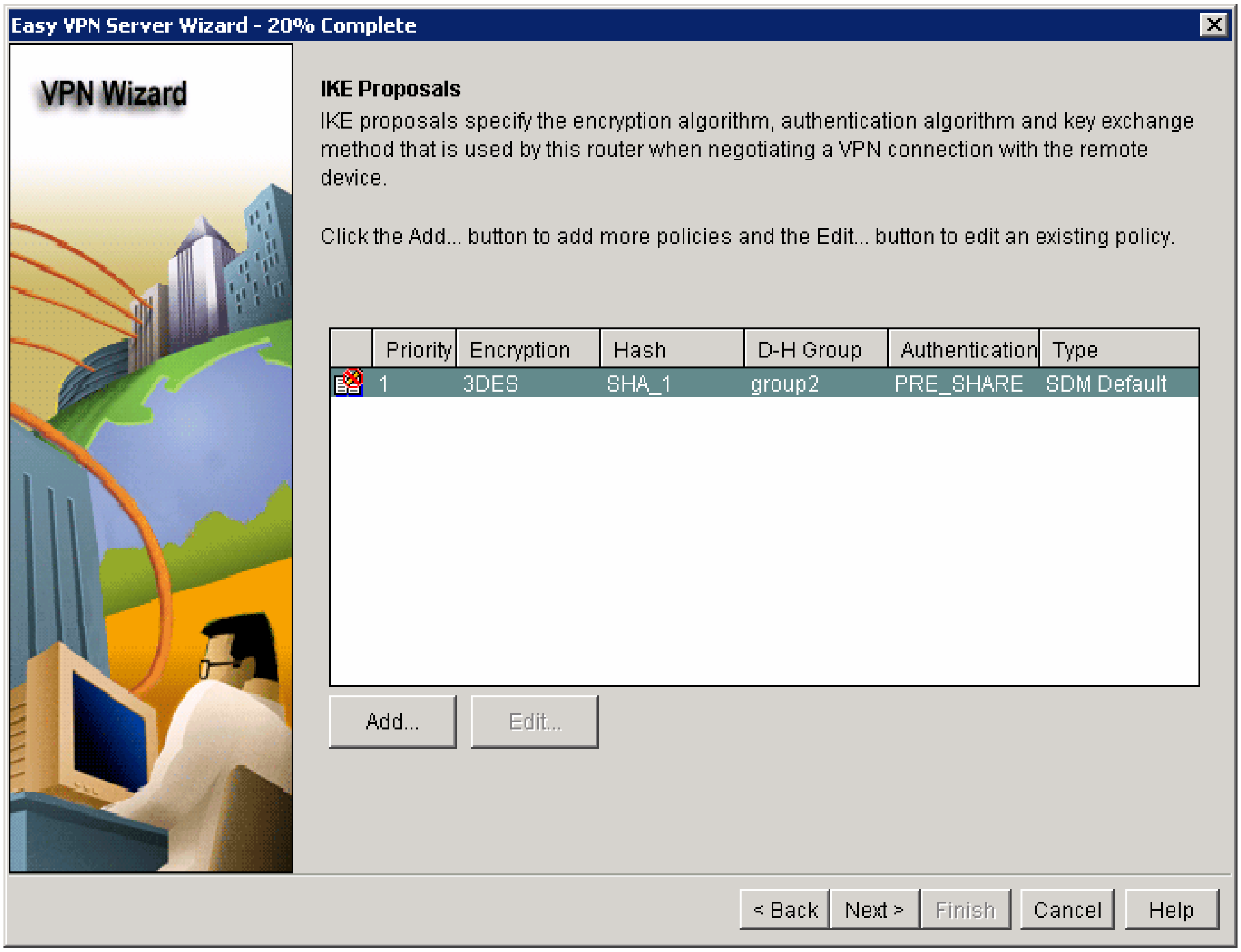
Interface connected to Internet. This is the interface where the VPN connections from the VPN clients will terminate.

You can select an interface which is participating in site to site VPN connection. But you cannot select an interface if it is participating in GRE over IPSec, DMVPN or Easy VPN client connection.

For more information please click the help button.

Figure 5-7: Interface and Authentication Options

Use the default SDM IKE proposal and click **Next**.



Easy VPN Server Wizard - 20% Complete

IKE Proposals

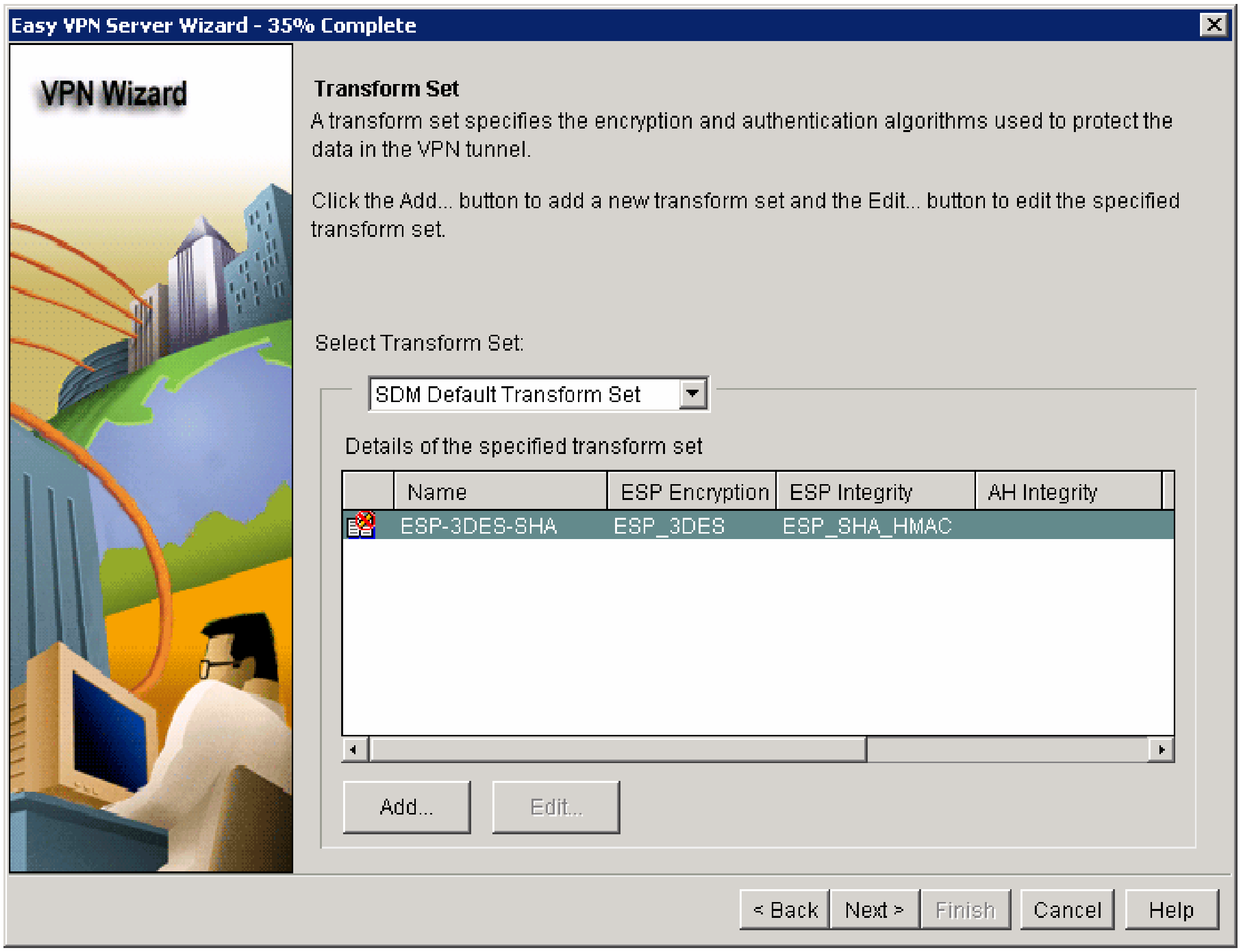
IKE proposals specify the encryption algorithm, authentication algorithm and key exchange method that is used by this router when negotiating a VPN connection with the remote device.

Click the Add... button to add more policies and the Edit... button to edit an existing policy.

	Priority	Encryption	Hash	D-H Group	Authentication	Type
	1	3DES	SHA_1	group2	PRE_SHARE	SDM Default

Figure 5-8: IKE Proposals List

Use the default SDM IPsec transform set and click **Next**.



Easy VPN Server Wizard - 35% Complete

Transform Set

A transform set specifies the encryption and authentication algorithms used to protect the data in the VPN tunnel.

Click the Add... button to add a new transform set and the Edit... button to edit the specified transform set.

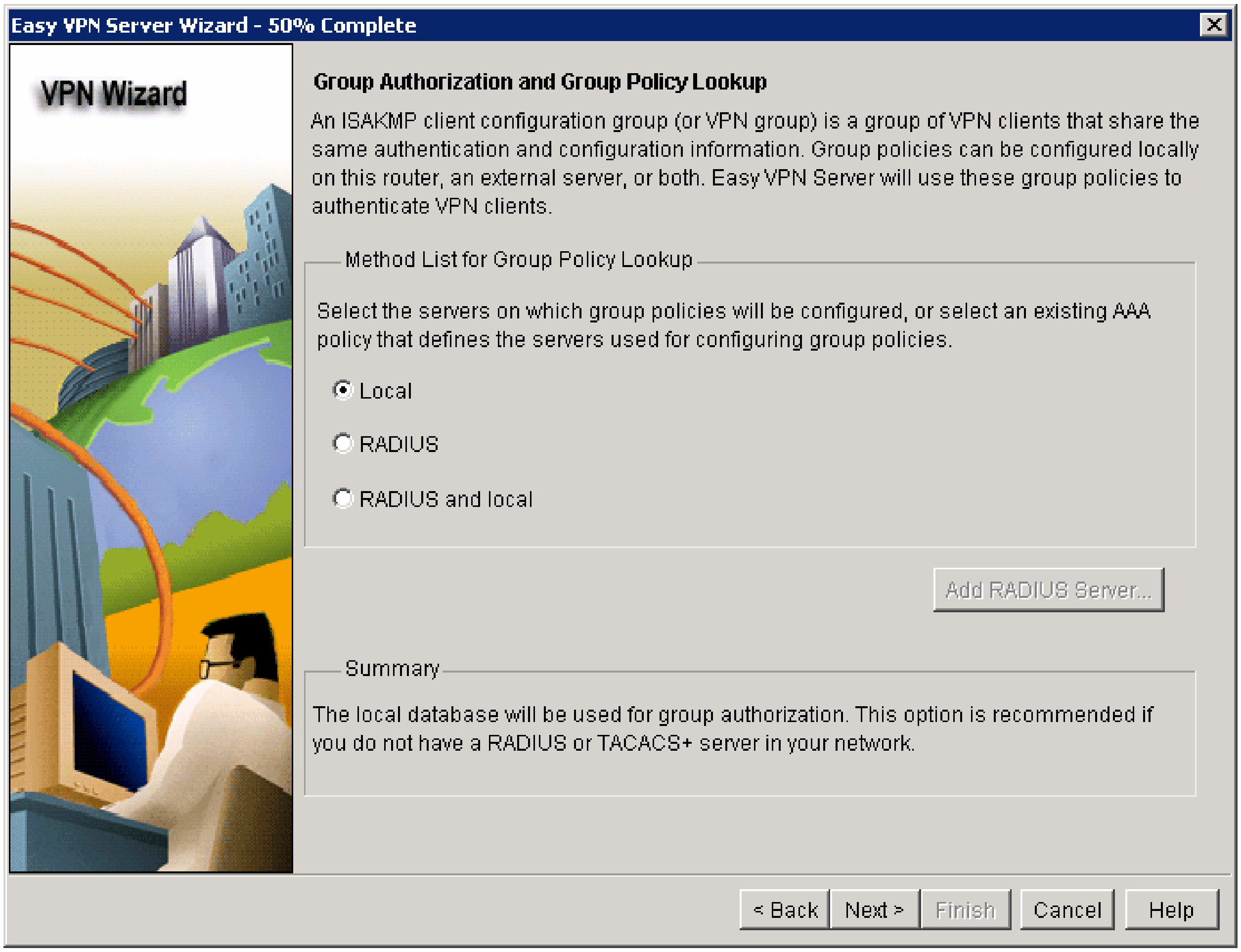
Select Transform Set: SDM Default Transform Set

Details of the specified transform set

	Name	ESP Encryption	ESP Integrity	AH Integrity
	ESP-3DES-SHA	ESP_3DES	ESP_SHA_HMAC	

Figure 5-9: IPsec Transform Set List

Choose **Local** in **Method List for Group Policy Lookup**, and then click **Next**.



Group Authorization and Group Policy Lookup

An ISAKMP client configuration group (or VPN group) is a group of VPN clients that share the same authentication and configuration information. Group policies can be configured locally on this router, an external server, or both. Easy VPN Server will use these group policies to authenticate VPN clients.

Method List for Group Policy Lookup

Select the servers on which group policies will be configured, or select an existing AAA policy that defines the servers used for configuring group policies.

● Local

○ RADIUS

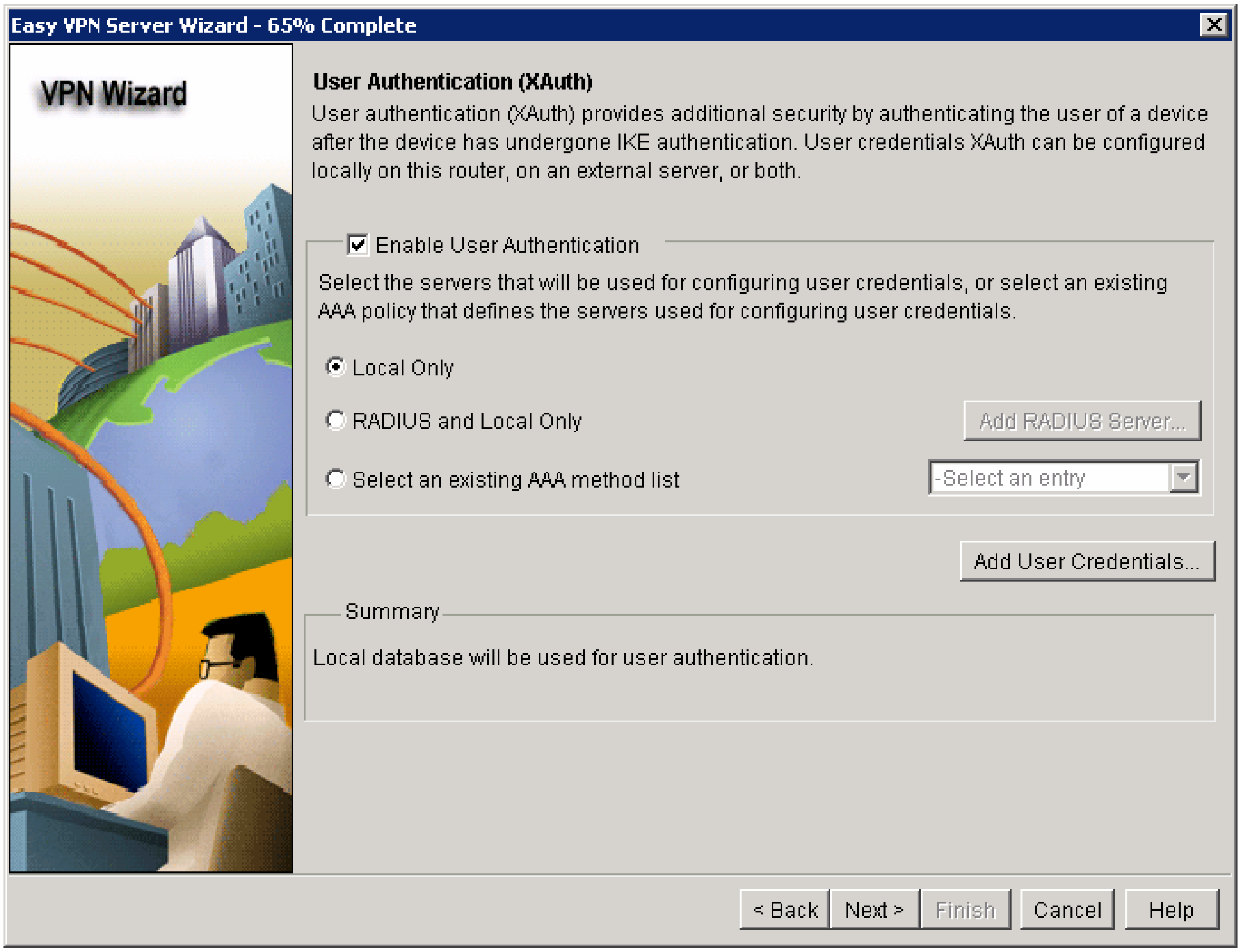
○ RADIUS and local

Summary

The local database will be used for group authorization. This option is recommended if you do not have a RADIUS or TACACS+ server in your network.

Figure 5-10: Authorization and Policy Options

Enable user authentication from a local database. Click **Add User Credentials...** to add a username for VPN access.



User Authentication (XAuth)

User authentication (XAuth) provides additional security by authenticating the user of a device after the device has undergone IKE authentication. User credentials XAuth can be configured locally on this router, on an external server, or both.

☑ Enable User Authentication

Select the servers that will be used for configuring user credentials, or select an existing AAA policy that defines the servers used for configuring user credentials.

● Local Only

○ RADIUS and Local Only

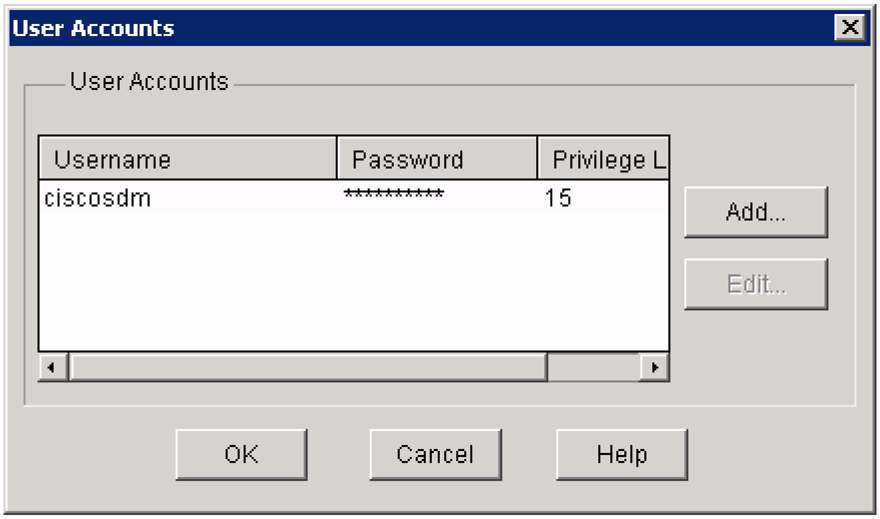
○ Select an existing AAA method list

Summary

Local database will be used for user authentication.

Figure 5-11: User Authentication Options

Click **Add...** to create a new user.



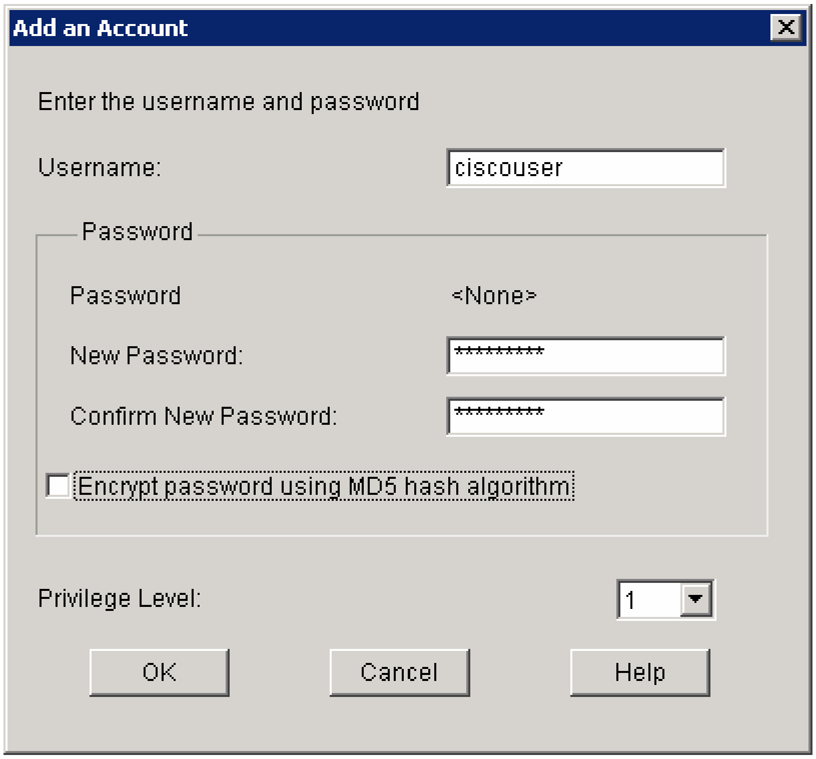
Username	Password	Privilege L
ciscosdm	**********	15

Figure 5-12: Local User Accounts

Create a username of “ciscouser” with a password of “ciscouser.” You can leave this user at privilege level 1 since it is only going to be used for VPN access. Encrypting this password is optional and not required.

If you clicked Encrypt password using MD5 hash algorithm, how would the password be stored?

Click **OK** twice when you are done, and then click **Next** in the user authentication window.



Add an Account

Enter the username and password

Username: ciscouser

Password

Password <None>

New Password: *********

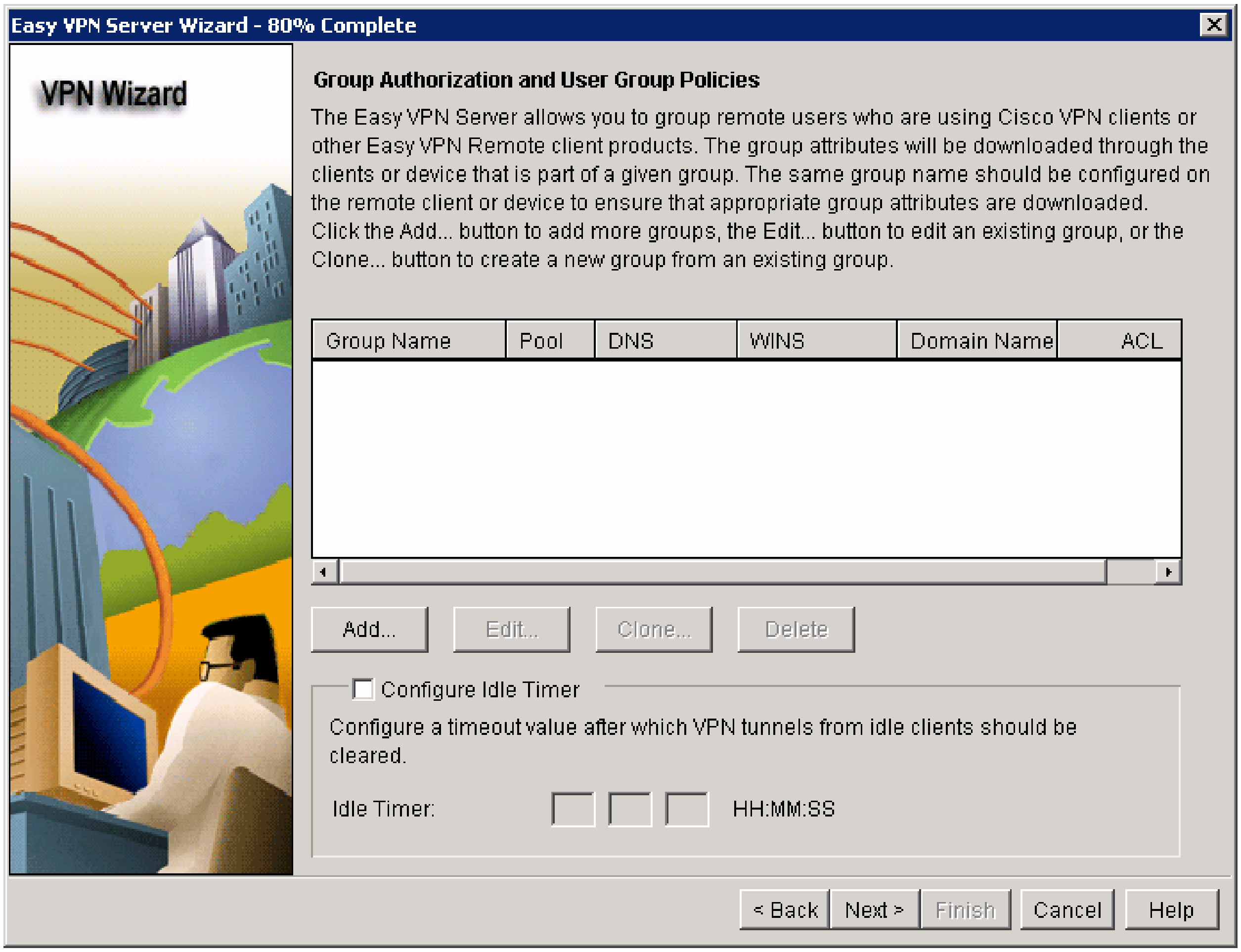
Confirm New Password: *********

□ Encrypt password using MD5 hash algorithm

Privilege Level: 1

Figure 5-13: Add User Account Dialog

We will need to create a group for our Easy VPN clients. To do this, click **Add...**.



Easy VPN Server Wizard - 80% Complete

Group Authorization and User Group Policies

The Easy VPN Server allows you to group remote users who are using Cisco VPN clients or other Easy VPN Remote client products. The group attributes will be downloaded through the clients or device that is part of a given group. The same group name should be configured on the remote client or device to ensure that appropriate group attributes are downloaded. Click the Add... button to add more groups, the Edit... button to edit an existing group, or the Clone... button to create a new group from an existing group.

Group Name	Pool	DNS	WINS	Domain Name	ACL

□ Configure Idle Timer

Configure a timeout value after which VPN tunnels from idle clients should be cleared.

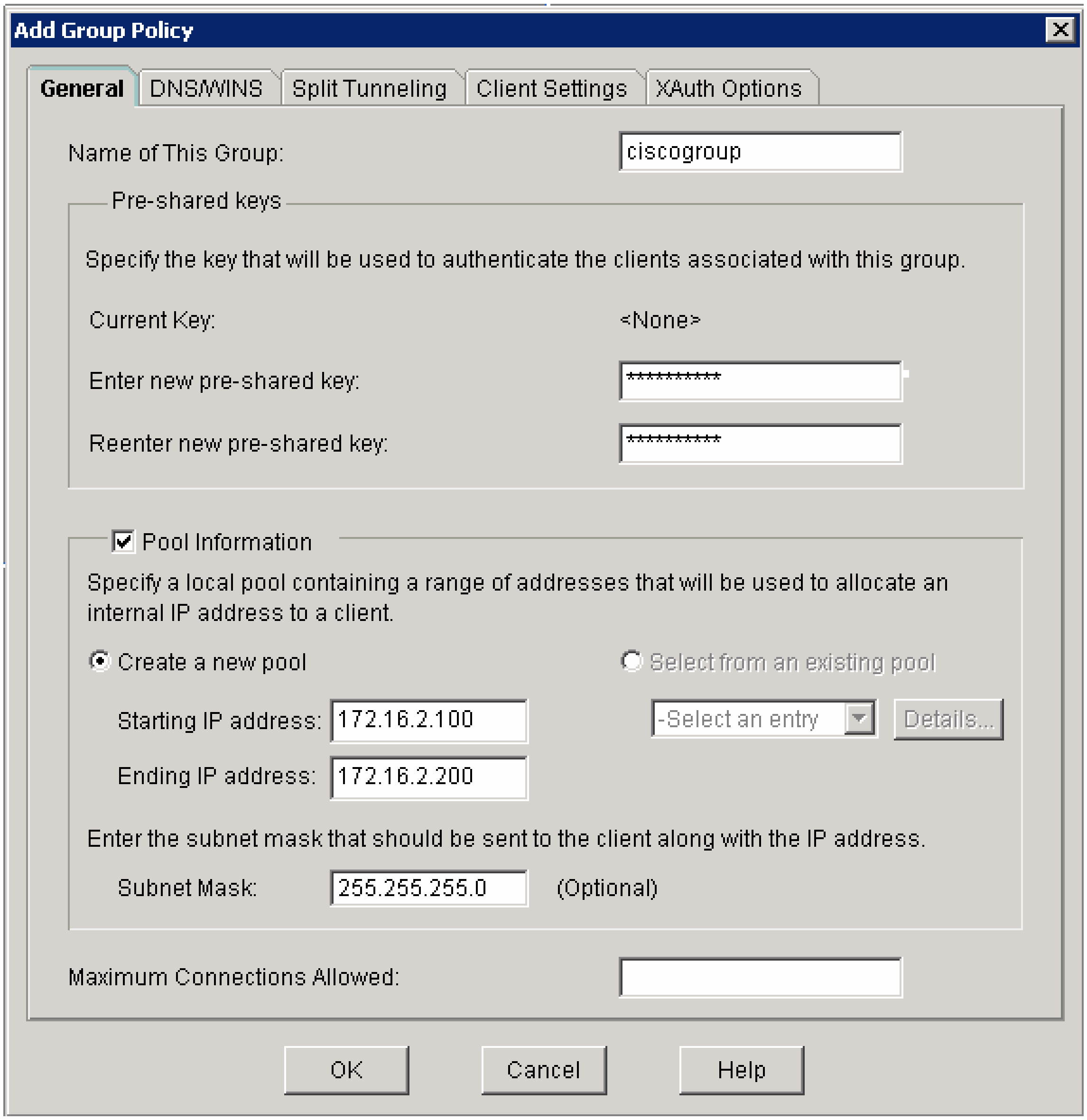
Idle Timer: ___ ___ ___ HH:MM:SS

Figure 5-14: VPN Client Authorization Configuration

Make the group name and pre-shared key “ciscogroup.” Create an IP pool for clients and use the range 172.16.2.100 – 172.16.2.200, with a subnet mask of 24 bits. Notice that this range falls under HQ’s loopback network. Click the **Split Tunneling** tab after completing these fields.

Why would you want to use an IP network associated with a loopback interface for your VPN pool?

How will HQ2 route traffic to the VPN clients?



Add Group Policy

Name of This Group: ciscogroup

Pre-shared keys

Specify the key that will be used to authenticate the clients associated with this group.

Current Key: <None>

Enter new pre-shared key: ***********

Reenter new pre-shared key: ***********

☑ Pool Information

Specify a local pool containing a range of addresses that will be used to allocate an internal IP address to a client.

● Create a new pool ○ Select from an existing pool

Starting IP address: 172.16.2.100

Ending IP address: 172.16.2.200

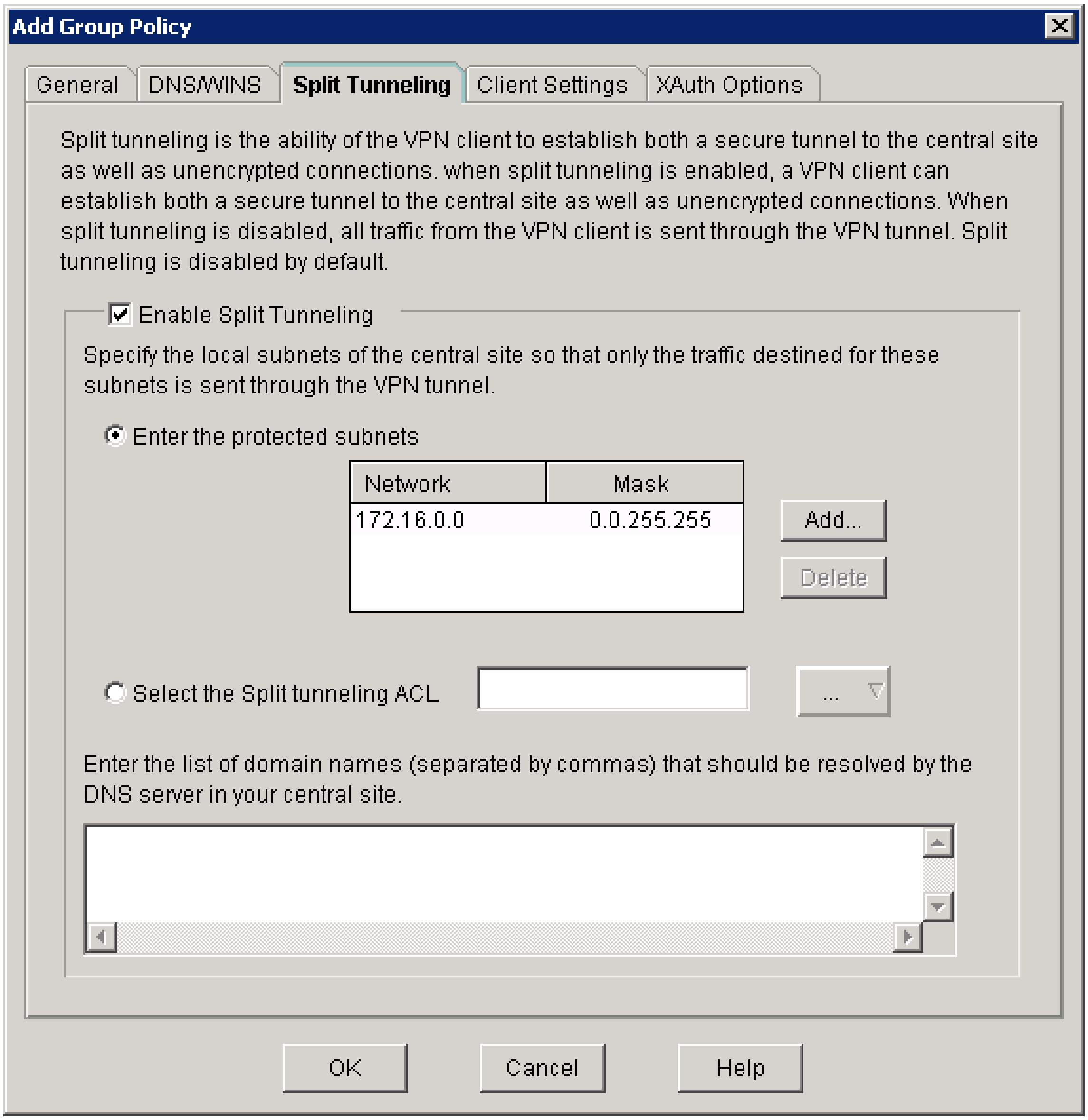
Enter the subnet mask that should be sent to the client along with the IP address.

Subnet Mask: 255.255.255.0 (Optional)

Maximum Connections Allowed:

Figure 5-15: VPN Group Policy Configuration

Enable split tunneling to advertise the entire 172.16.0.0 network into the route table of VPN clients. Click the **Add...** button and add the network with the appropriate wildcard mask. When complete, click **OK**.



Split tunneling is the ability of the VPN client to establish both a secure tunnel to the central site as well as unencrypted connections. when split tunneling is enabled, a VPN client can establish both a secure tunnel to the central site as well as unencrypted connections. When split tunneling is disabled, all traffic from the VPN client is sent through the VPN tunnel. Split tunneling is disabled by default.

☑ Enable Split Tunneling

Specify the local subnets of the central site so that only the traffic destined for these subnets is sent through the VPN tunnel.

● Enter the protected subnets

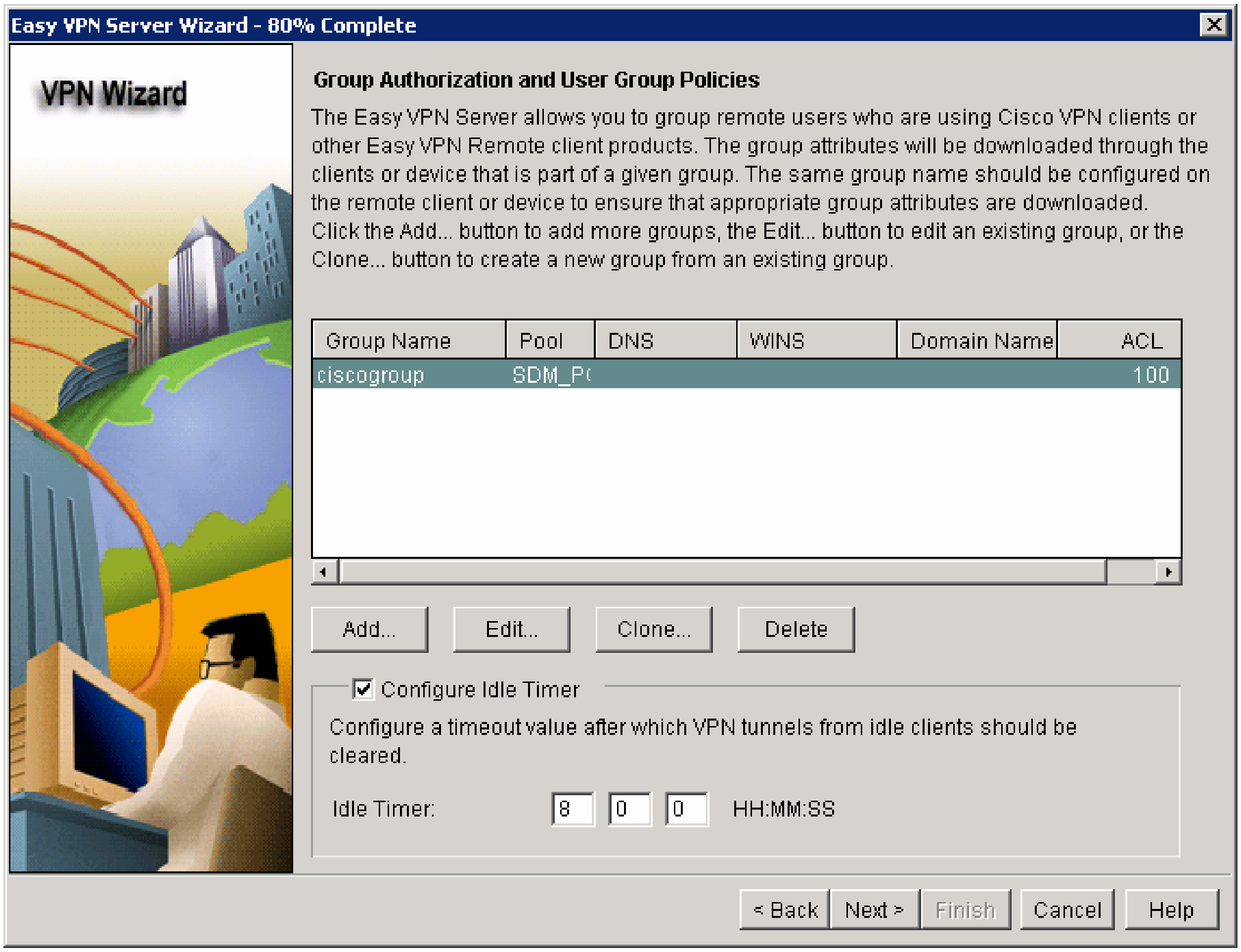
Network	Mask
172.16.0.0	0.0.255.255

○ Select the Split tunneling ACL

Enter the list of domain names (separated by commas) that should be resolved by the DNS server in your central site.

Figure 5-16: Split Tunneling Tab

You should see the new group information added. Configure an idle timer of 8 hours and click **Next**.



Easy VPN Server Wizard - 80% Complete

Group Authorization and User Group Policies

The Easy VPN Server allows you to group remote users who are using Cisco VPN clients or other Easy VPN Remote client products. The group attributes will be downloaded through the clients or device that is part of a given group. The same group name should be configured on the remote client or device to ensure that appropriate group attributes are downloaded. Click the Add... button to add more groups, the Edit... button to edit an existing group, or the Clone... button to create a new group from an existing group.

Group Name	Pool	DNS	WINS	Domain Name	ACL
ciscogroup	SDM_P(				100

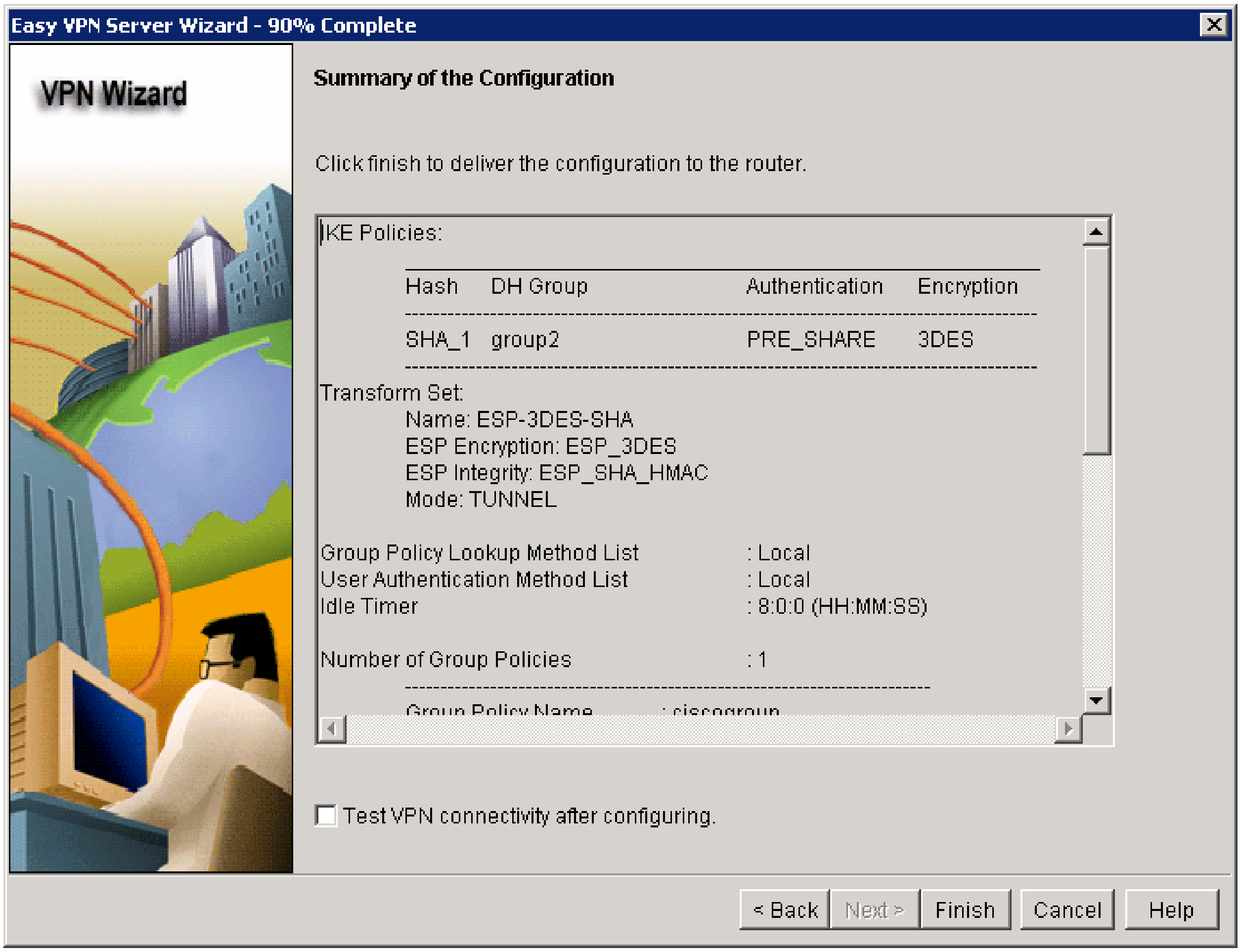
☑ Configure Idle Timer

Configure a timeout value after which VPN tunnels from idle clients should be cleared.

Idle Timer: 8 0 0 HH:MM:SS

Figure 5-17: VPN Client Authorization Configuration with Changes Applied

Review what SDM will send to the router and click **Finish**.



Summary of the Configuration

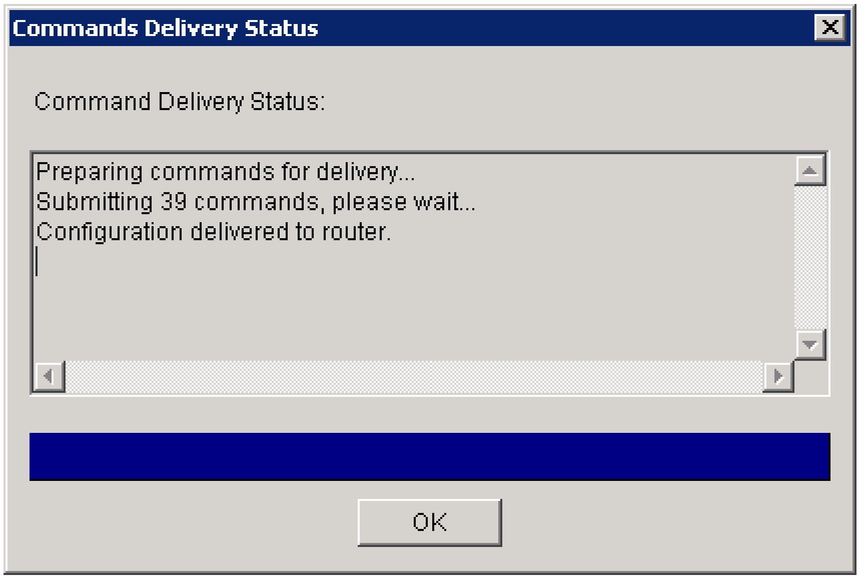
Click finish to deliver the configuration to the router.

```
IKE Policies:

          Hash     DH Group                     Authentication     Encryption
          -------------------------------------------------------------------
          SHA_1    group2                       PRE_SHARE          3DES
          -------------------------------------------------------------------
Transform Set:
          Name: ESP-3DES-SHA
          ESP Encryption: ESP_3DES
          ESP Integrity: ESP_SHA_HMAC
          Mode: TUNNEL

Group Policy Lookup Method List              : Local
User Authentication Method List              : Local
Idle Timer                                   : 8:0:0 (HH:MM:SS)

Number of Group Policies                     : 1
          ------------------------------------------------------------
          Group Policy Name          : ciscogroup
```

□ Test VPN connectivity after configuring.

Figure 5-18: Summary of Easy VPN Configuration

Click **Finish** and SDM will deliver the configuration to the router. Click **OK** when delivery is complete.



```
Preparing commands for delivery...
Submitting 39 commands, please wait...
Configuration delivered to router.
```

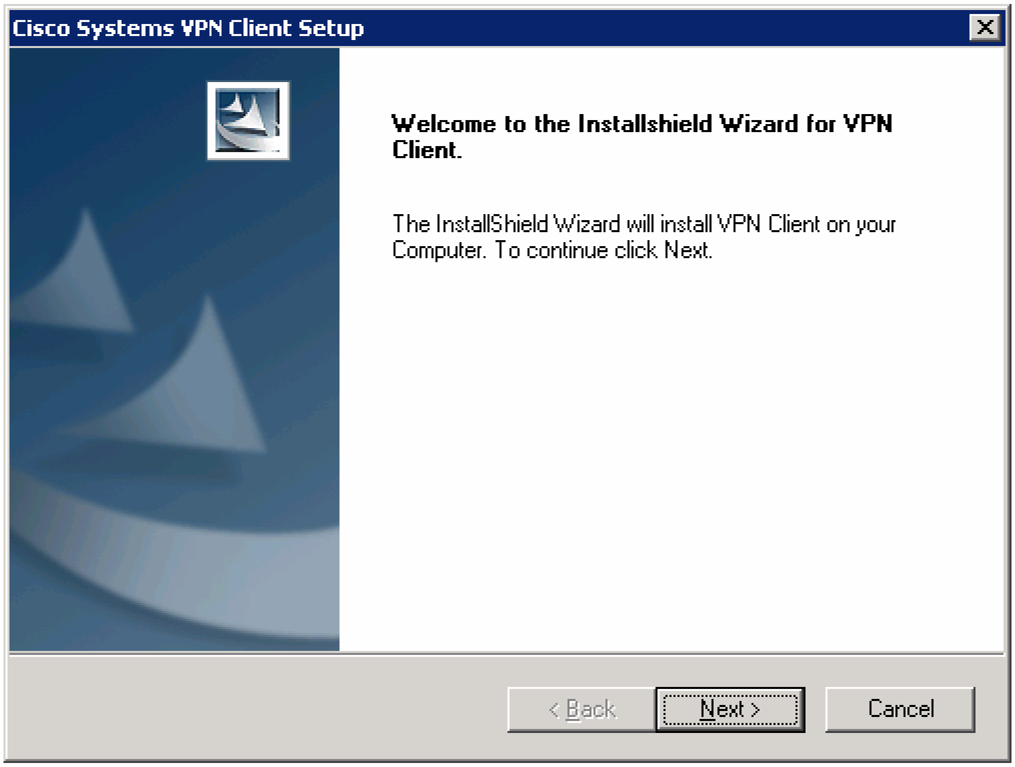
Figure 5-19: Command Delivery Progress Indicator

You have now successfully configured Easy VPN server.

Step 6: Install the Cisco VPN Client

Now that HQ has been set up as an Easy VPN Server, the host will change its role from management host to a VPN client connecting across the Internet to HQ. Before you can connect, you must install the Cisco VPN Client if you haven’t already. If you have already installed the VPN Client, skip this step and move on to Step 7.

To begin the installation, download the VPN Client from Cisco, and extract it to a temporary directory. Run the setup.exe file in the temporary directory to start installation. Click **Next** when the installer welcomes you.

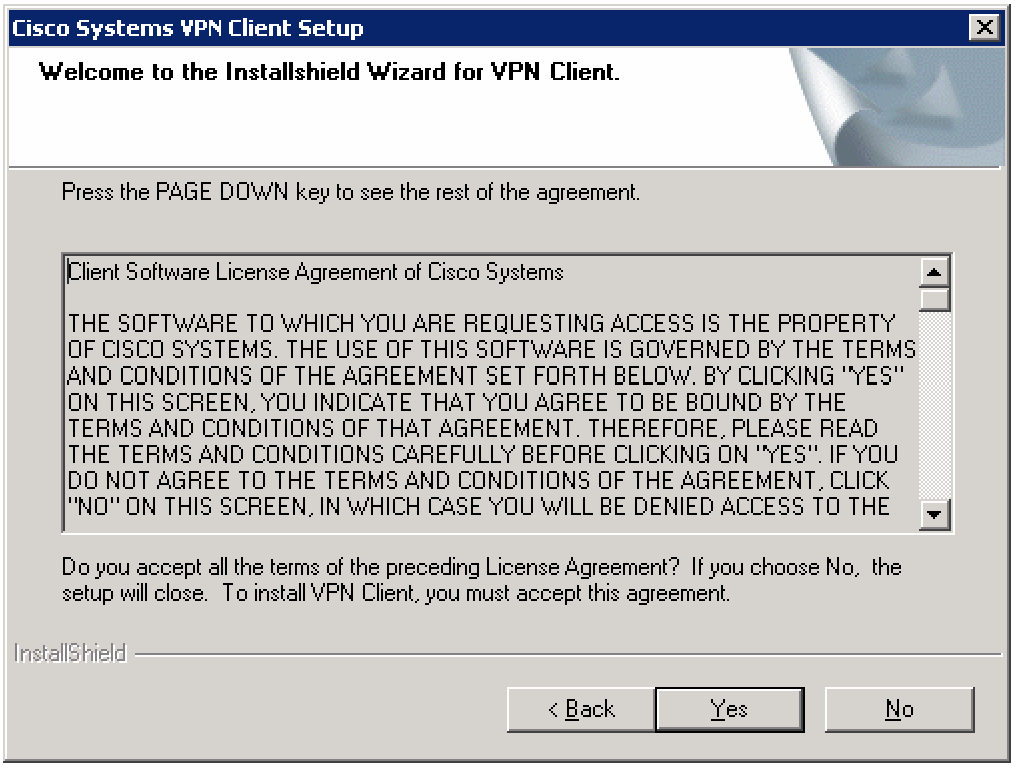

Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

The InstallShield Wizard will install VPN Client on your Computer. To continue click Next.

Figure 6-1: VPN Client Installation Wizard

Click **Yes** after reading the software license agreement.


Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

Press the PAGE DOWN key to see the rest of the agreement.

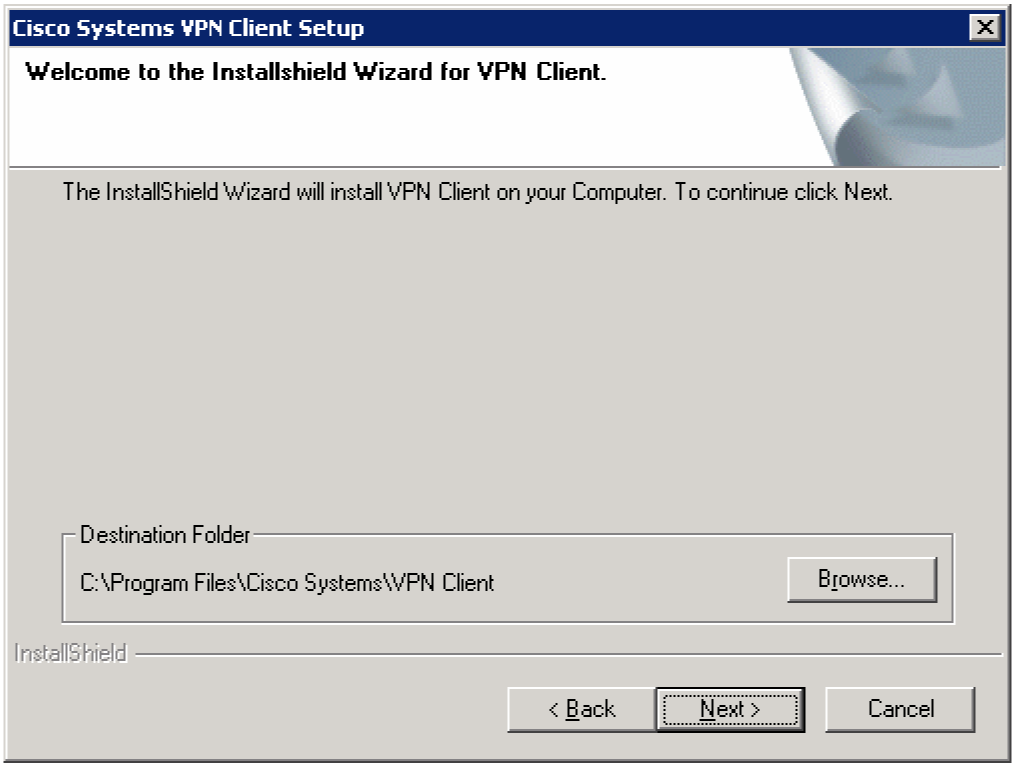
Client Software License Agreement of Cisco Systems

THE SOFTWARE TO WHICH YOU ARE REQUESTING ACCESS IS THE PROPERTY OF CISCO SYSTEMS. THE USE OF THIS SOFTWARE IS GOVERNED BY THE TERMS AND CONDITIONS OF THE AGREEMENT SET FORTH BELOW. BY CLICKING "YES" ON THIS SCREEN, YOU INDICATE THAT YOU AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THAT AGREEMENT. THEREFORE, PLEASE READ THE TERMS AND CONDITIONS CAREFULLY BEFORE CLICKING ON "YES". IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THE AGREEMENT, CLICK "NO" ON THIS SCREEN, IN WHICH CASE YOU WILL BE DENIED ACCESS TO THE

Do you accept all the terms of the preceding License Agreement? If you choose No, the setup will close. To install VPN Client, you must accept this agreement.

Figure 6-2: Cisco VPN Client License Agreement

Click **Next** to use the default installation.



Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

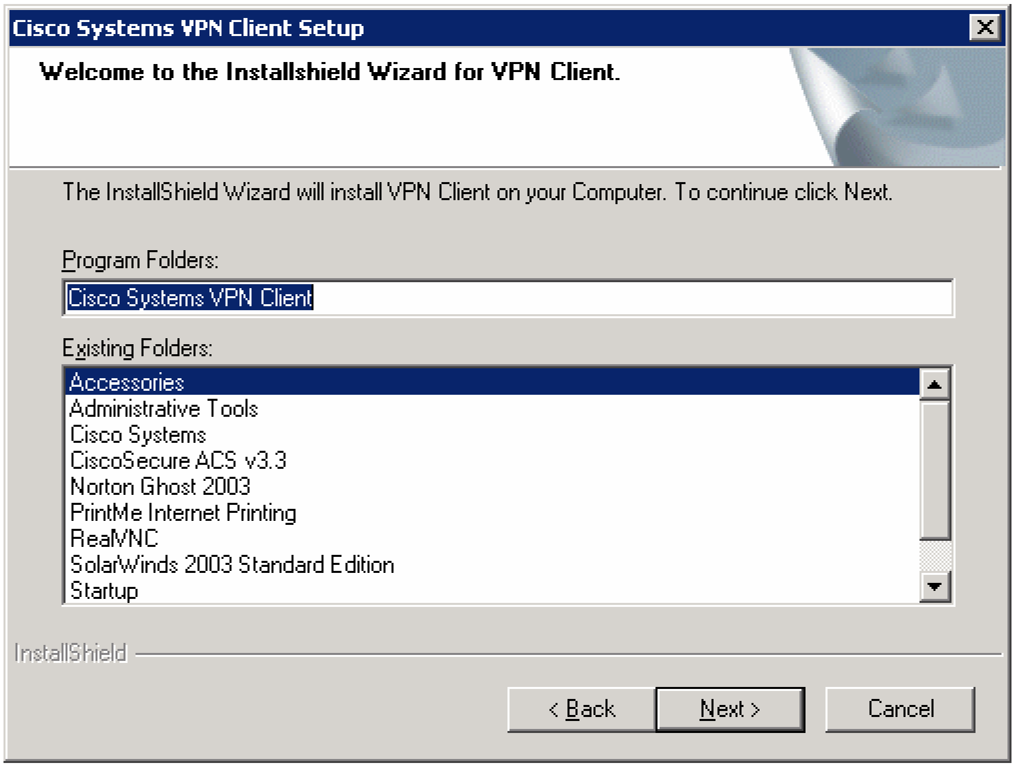
The InstallShield Wizard will install VPN Client on your Computer. To continue click Next.

Destination Folder

C:\Program Files\Cisco Systems\VPN Client

Figure 6-3: VPN Client Installation Location

Choose the default program group and click **Next**.



Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

The InstallShield Wizard will install VPN Client on your Computer. To continue click Next.

Program Folders:

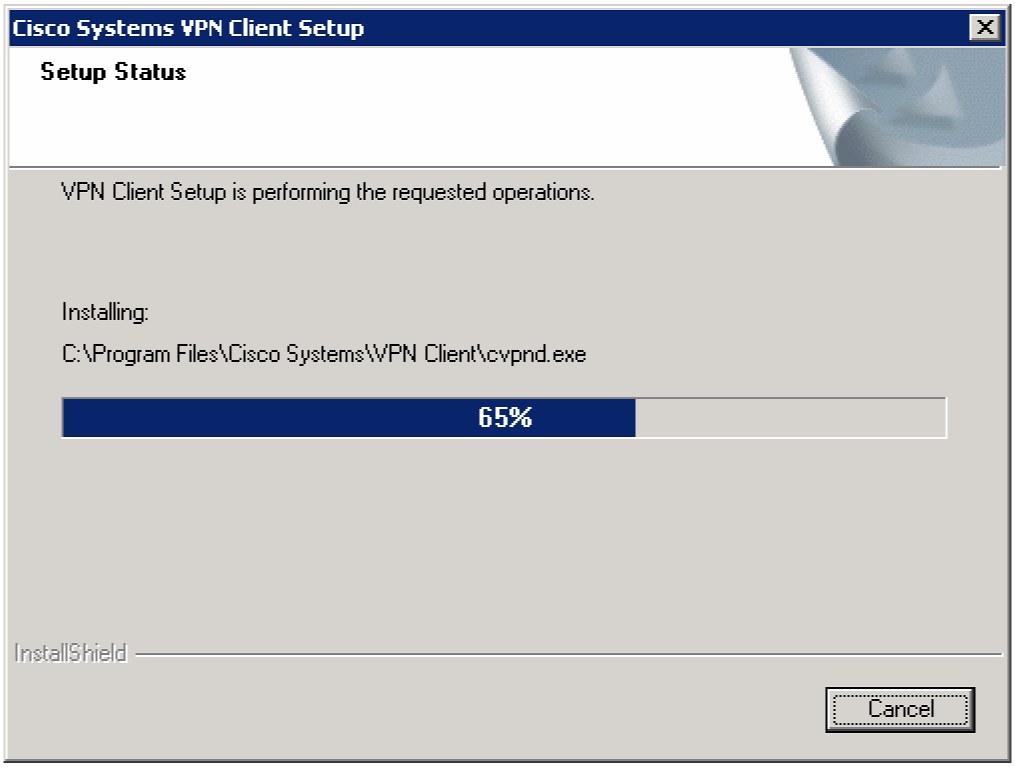
Cisco Systems VPN Client

Existing Folders:

Accessories
Administrative Tools
Cisco Systems
CiscoSecure ACS v3.3
Norton Ghost 2003
PrintMe Internet Printing
RealVNC
SolarWinds 2003 Standard Edition
Startup

Figure 6-4: Start Menu Program Folder Selection

Allow the wizard to install all necessary files. At the end of the process, the wizard will add the virtual network interfaces required for VPN use. This may take some time.



Cisco Systems VPN Client Setup

Setup Status

VPN Client Setup is performing the requested operations.

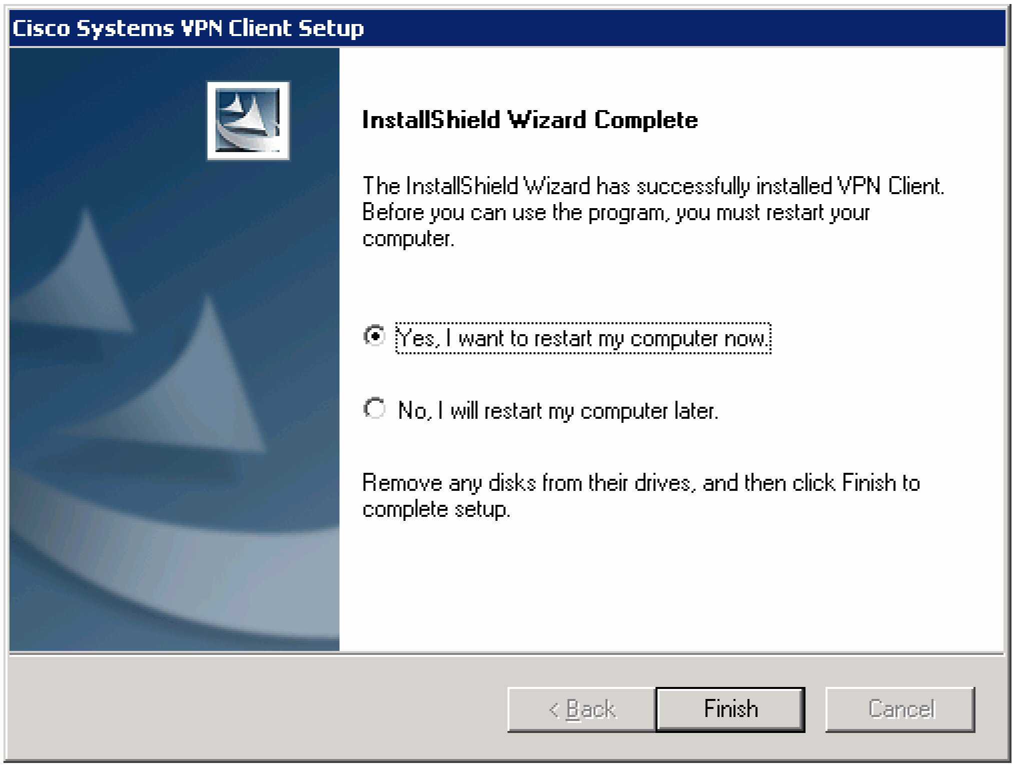
Installing:

C:\Program Files\Cisco Systems\VPN Client\cvpnd.exe

65%

Figure 6-5: VPN Client Installation Progress Indicator

At the end of the installer, you will be required to restart. Click **Finish** to let your computer restart.



Cisco Systems VPN Client Setup

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed VPN Client. Before you can use the program, you must restart your computer.

● Yes, I want to restart my computer now.

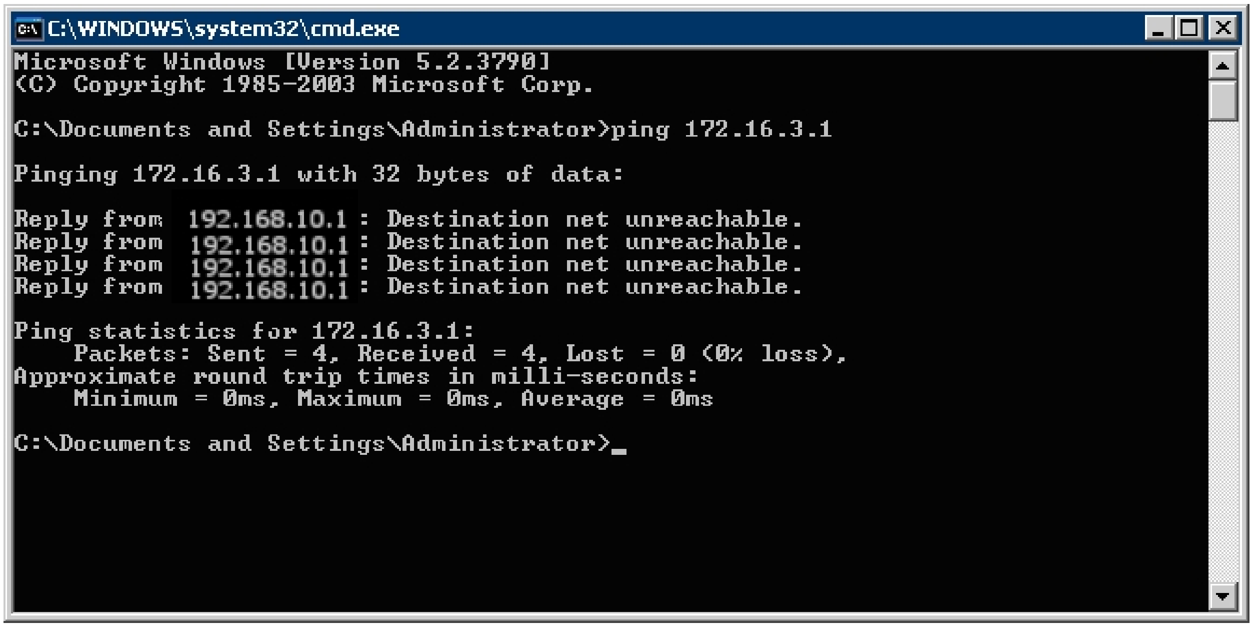
○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

Figure 6-6: Final Installation Wizard Window

Step 7: Test Access from Client without VPN Connection

After restarting the host with the VPN client installed, open up a command prompt. Click the **Start** button, choose **Run...** and type **cmd,** and then click **OK**. Try pinging HQ2's loopback address. The pings should fail.



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 172.16.3.1

Pinging 172.16.3.1 with 32 bytes of data:

Reply from 192.168.10.1: Destination net unreachable.
Reply from 192.168.10.1: Destination net unreachable.
Reply from 192.168.10.1: Destination net unreachable.
Reply from 192.168.10.1: Destination net unreachable.

Ping statistics for 172.16.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator>_
```

Figure 7-1: Unsuccessful Pings Without VPN

Step 8: Connect to the VPN

Start the Cisco VPN Client by clicking the **Start** button, and choosing **Programs > Cisco Systems VPN Client > VPN Client**.

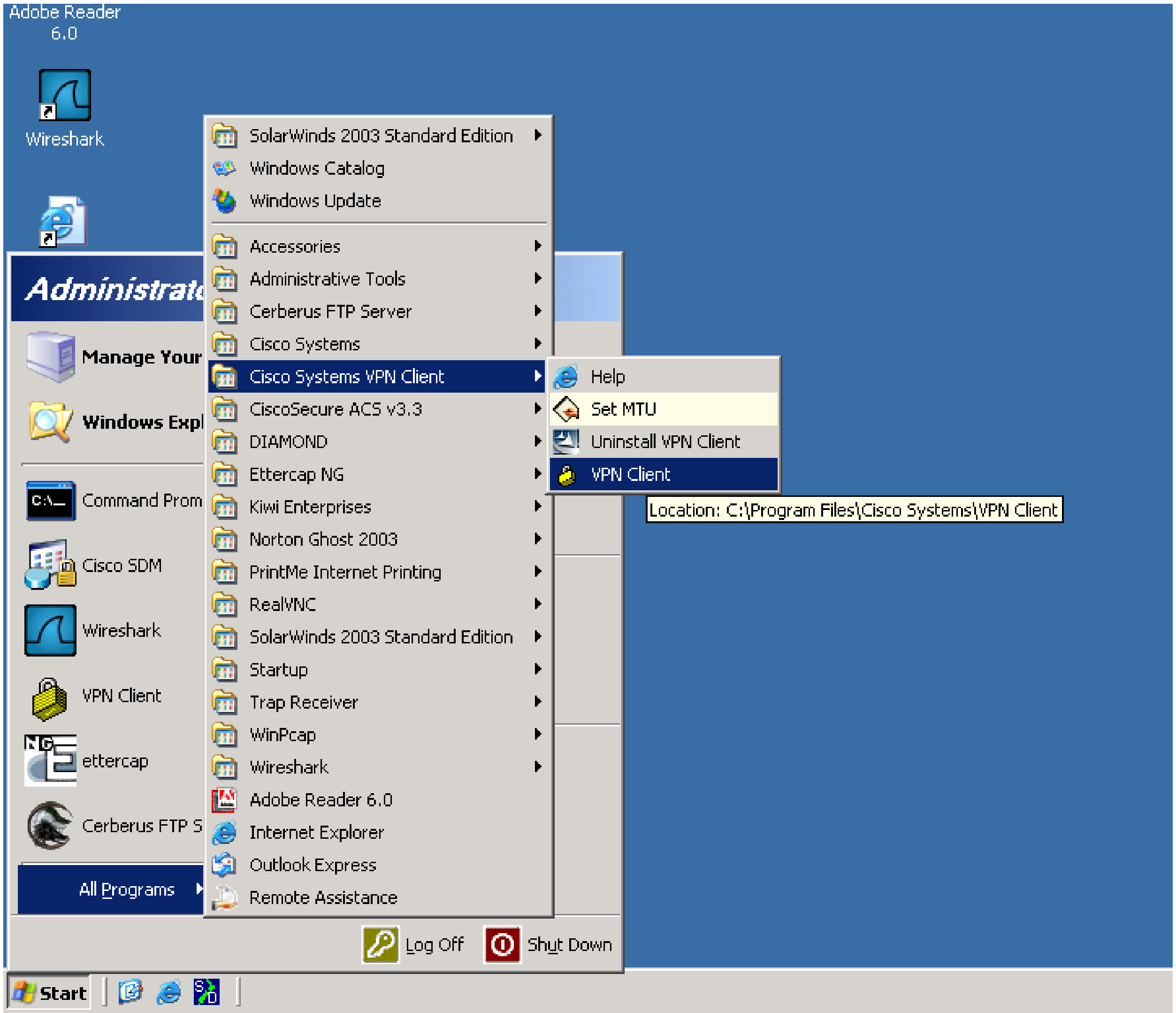


Figure 8-1: Launching the VPN Client

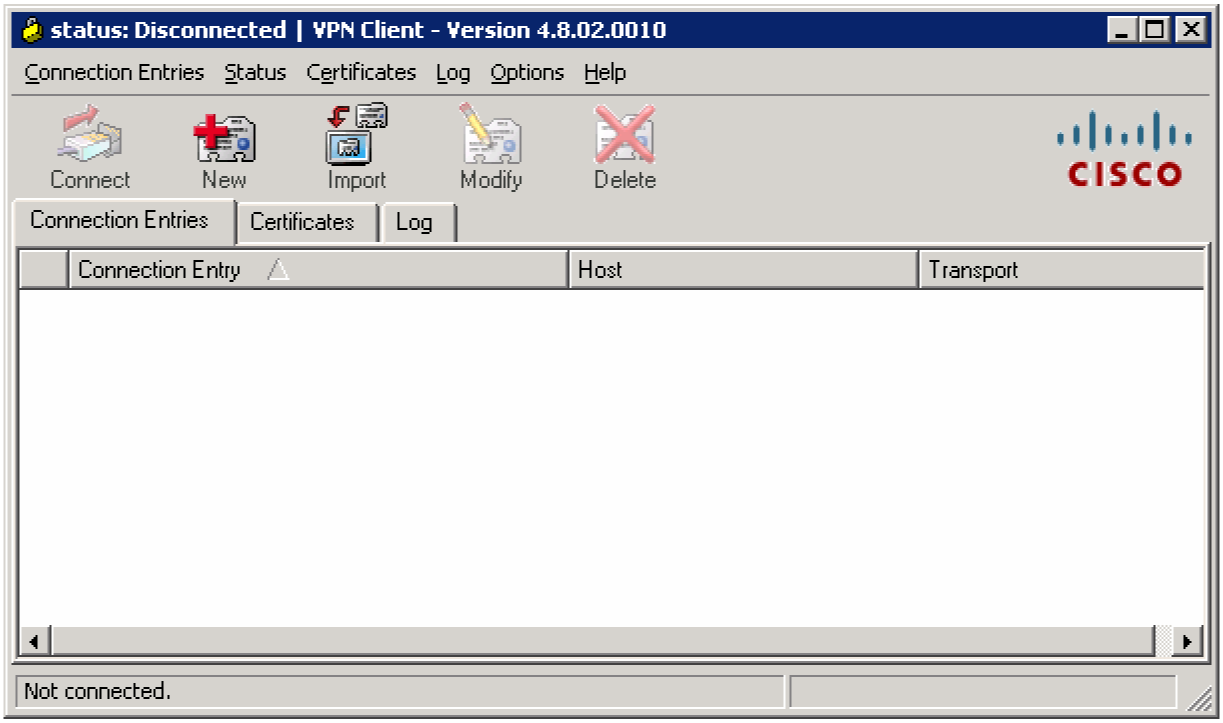
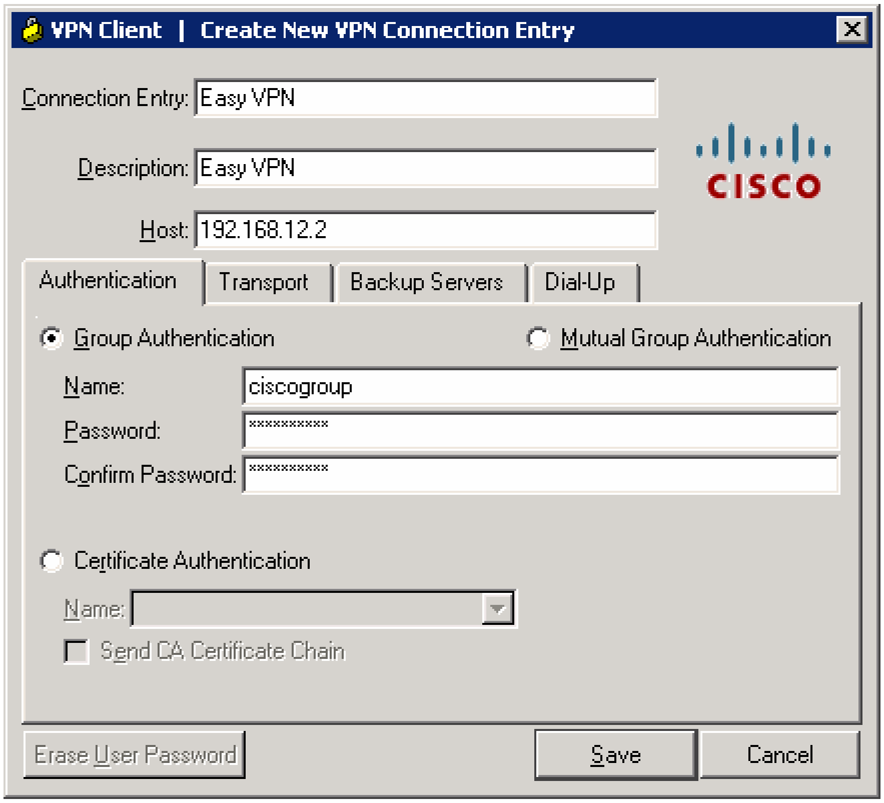


Figure 8-2: VPN Client Application

Once the VPN Client is open, you will need to create a new connection profile to connect to HQ with. Click the **New** button. Create the new connection with any name and description you want. For host, enter the IP of HQ's Serial0/0/0 interface, 192.168.12.2. The host IP address represents the IP address of the VPN server or concentrator to which you wish to connect. In this case, HQ is running the Easy VPN Server and will function as such. Use the group name and password previously configured in the Easy VPN wizard. Click **Save** when you are done configuring.



Connection Entry: Easy VPN

Description: Easy VPN

Host: 192.168.12.2

● Group Authentication ○ Mutual Group Authentication

Name: ciscogroup

Password: ***********

Confirm Password: ***********

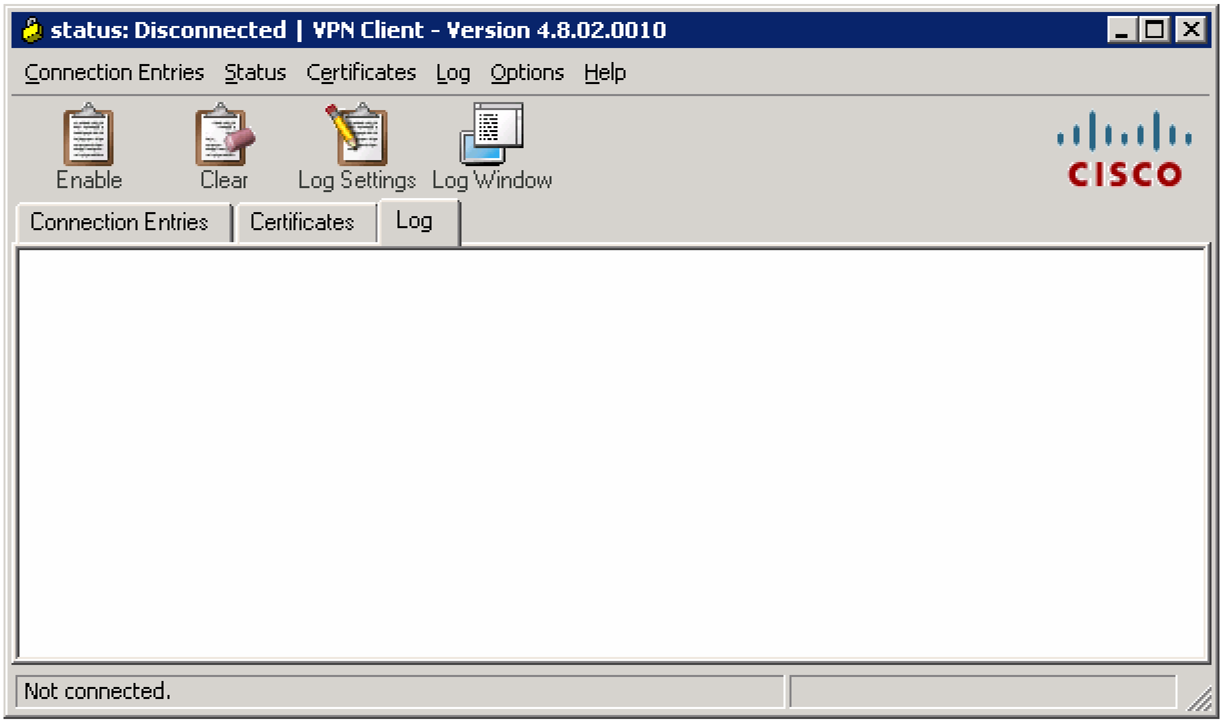
○ Certificate Authentication

Name:

□ Send CA Certificate Chain

Figure 8-3: Create New VPN Connection Dialog

You should see your new profile appear in the profiles list. Before connecting, click the **Log** tab so you can enable logging before attempting to connect. Logging is not normally required but it is helpful in this lab to watch the VPN client connect.



Not connected.

Figure 8-4: VPN Client Log Tab

Click **Log Window** to open up logging in a separate window.

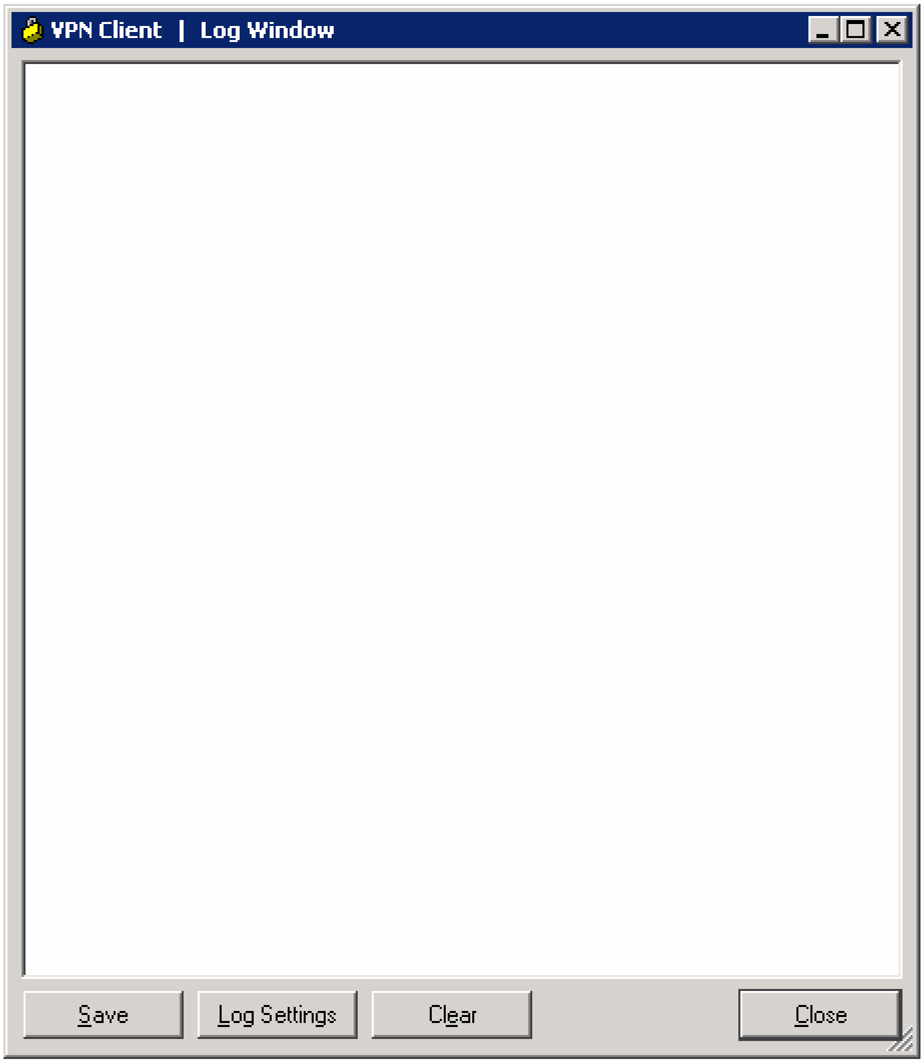
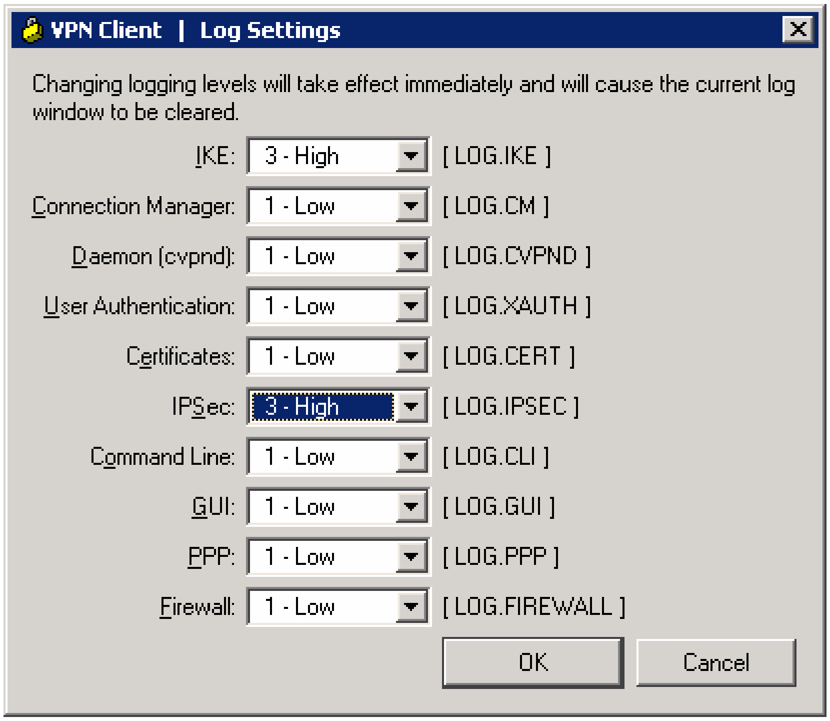


Figure 8-5: Log Window

While you have the log window open, go back to the main VPN client window and click **Log Settings**. Change the logging settings for IKE and IPsec to **3 – High**. Click **OK** to apply these settings.

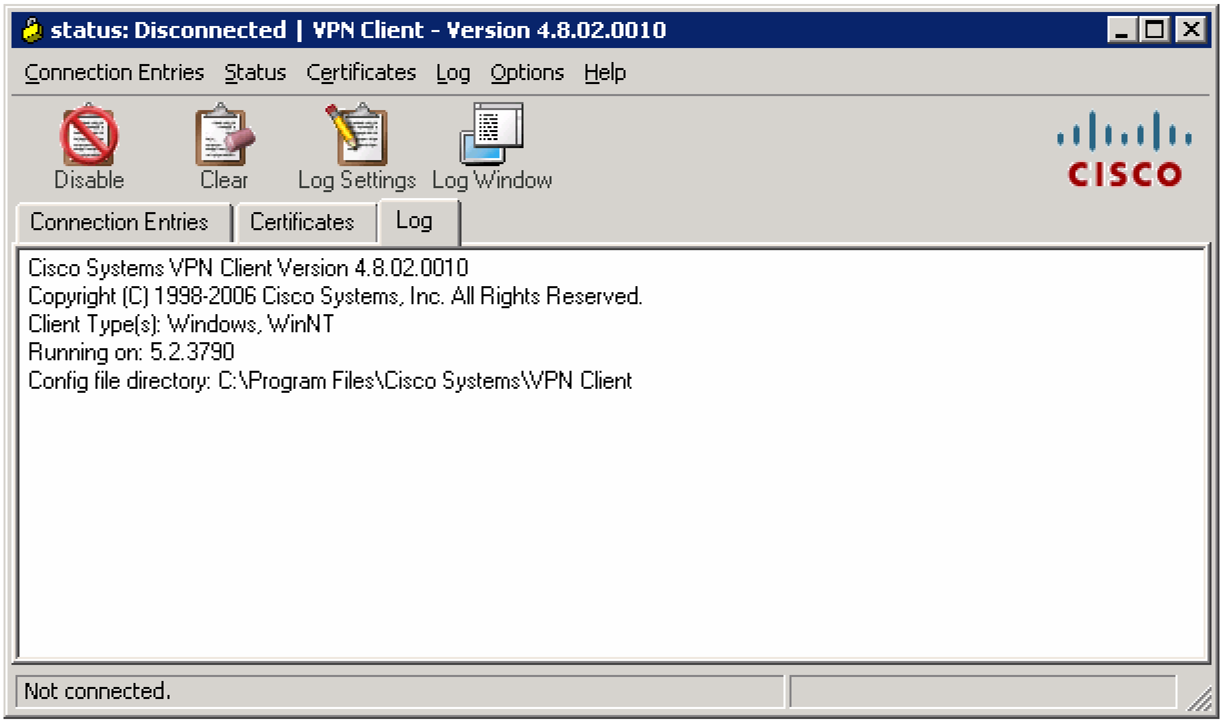


Changing logging levels will take effect immediately and will cause the current log window to be cleared.

IKE: 3 - High [LOG.IKE]
Connection Manager: 1 - Low [LOG.CM]
Daemon (cvpnd): 1 - Low [LOG.CVPND]
User Authentication: 1 - Low [LOG.XAUTH]
Certificates: 1 - Low [LOG.CERT]
IPSec: 3 - High [LOG.IPSEC]
Command Line: 1 - Low [LOG.CLI]
GUI: 1 - Low [LOG.GUI]
PPP: 1 - Low [LOG.PPP]
Firewall: 1 - Low [LOG.FIREWALL]

Figure 8-6: Logging Settings

Click **Enable** to enable logging. The **Enable** button should change to a **Disable** button.

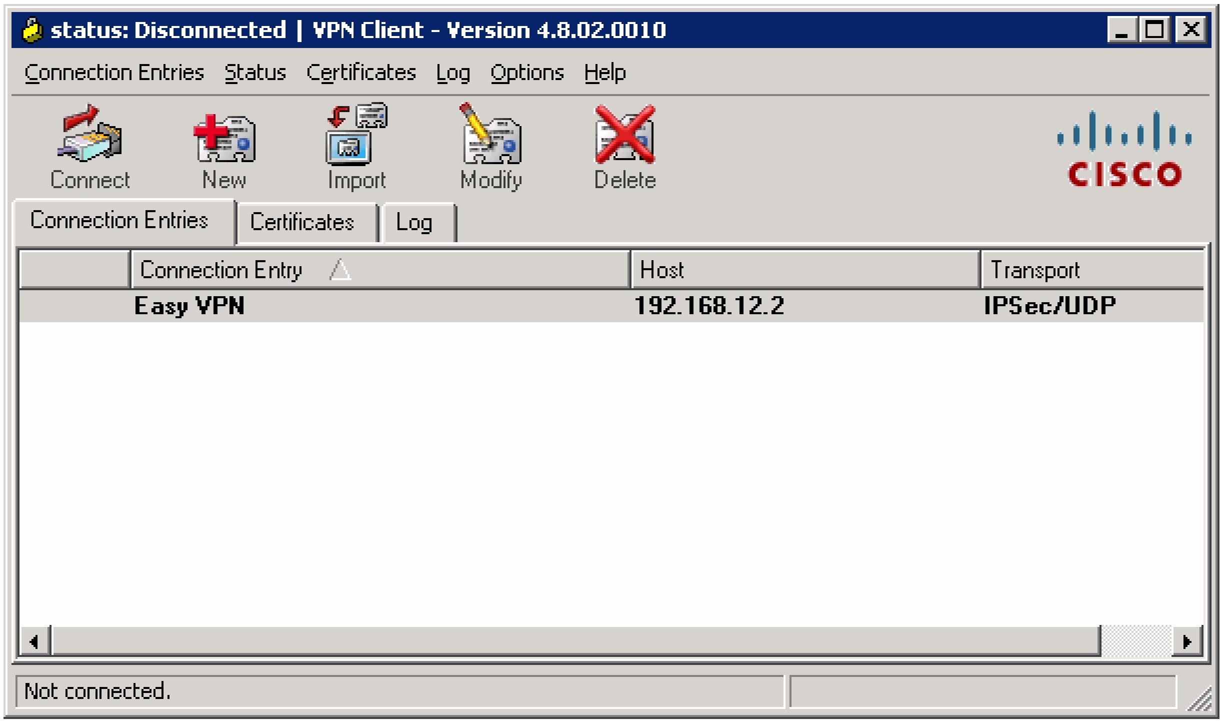


```
Cisco Systems VPN Client Version 4.8.02.0010
Copyright (C) 1998-2006 Cisco Systems, Inc. All Rights Reserved.
Client Type(s): Windows, WinNT
Running on: 5.2.3790
Config file directory: C:\Program Files\Cisco Systems\VPN Client
```

Not connected.

Figure 8-7: Log Tab with Logging Enabled

Click the **Connection Entries** tab, and double-click the entry or click **Connect** to connect to this profile.



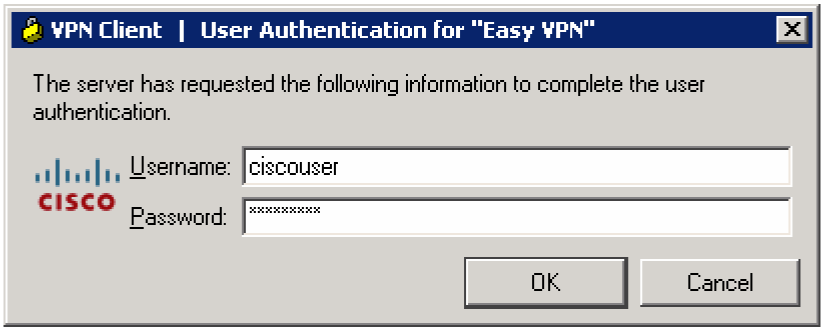
status: Disconnected | VPN Client - Version 4.8.02.0010
Connection Entries Status Certificates Log Options Help
Connect
New
Import
Modify
Delete
CISCO
Connection Entries
Certificates
Log

	Connection Entry	Host	Transport
	Easy VPN	192.168.12.2	IPSec/UDP

Not connected.

Figure 8-8: VPN Client Connections Tab

While the VPN client tries to connect to the VPN, it will prompt you for a username and password. Enter the user credentials you specified earlier during the VPN client wizard.



The server has requested the following information to complete the user authentication.

Username: ciscouser

Password: **********

Figure 8-9: User Authentication Prompt

When the VPN has successfully connected, you should see a locked padlock icon in the system tray.

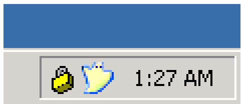
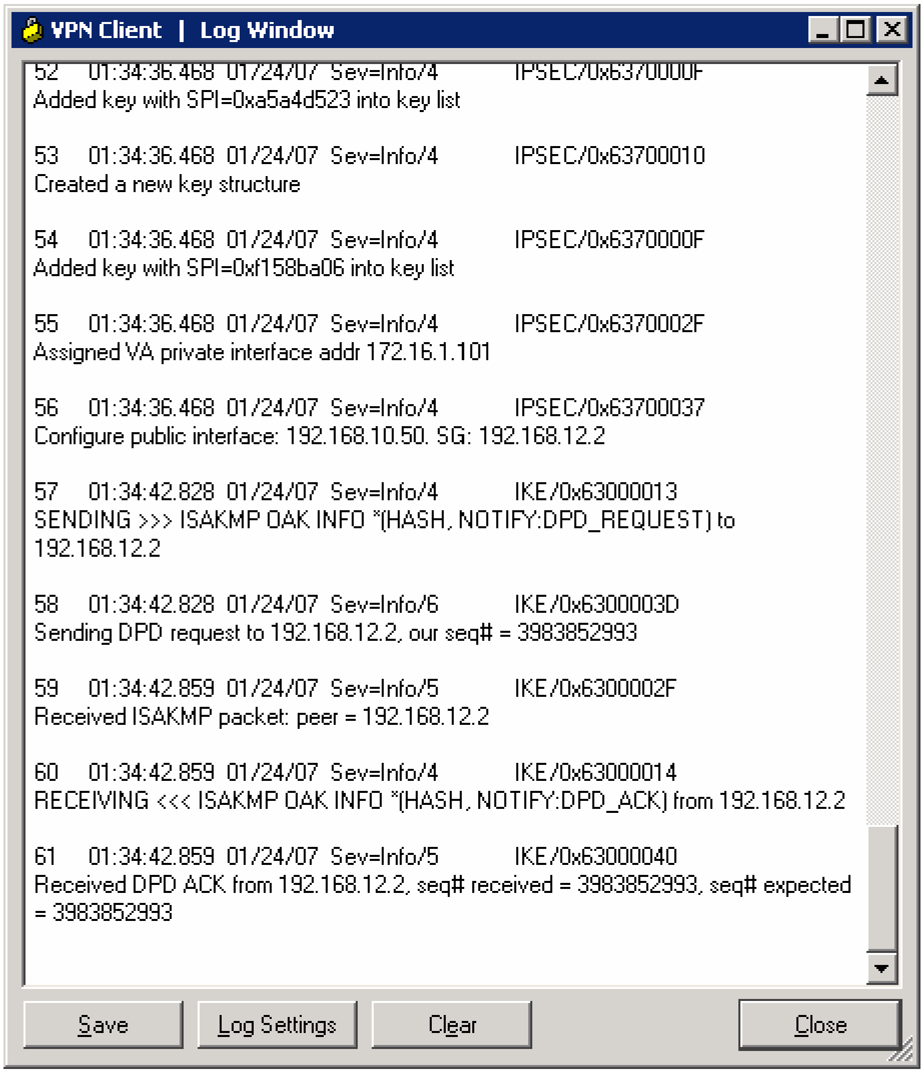


Figure 8-10: VPN Client System Tray Icon, Status: Connected

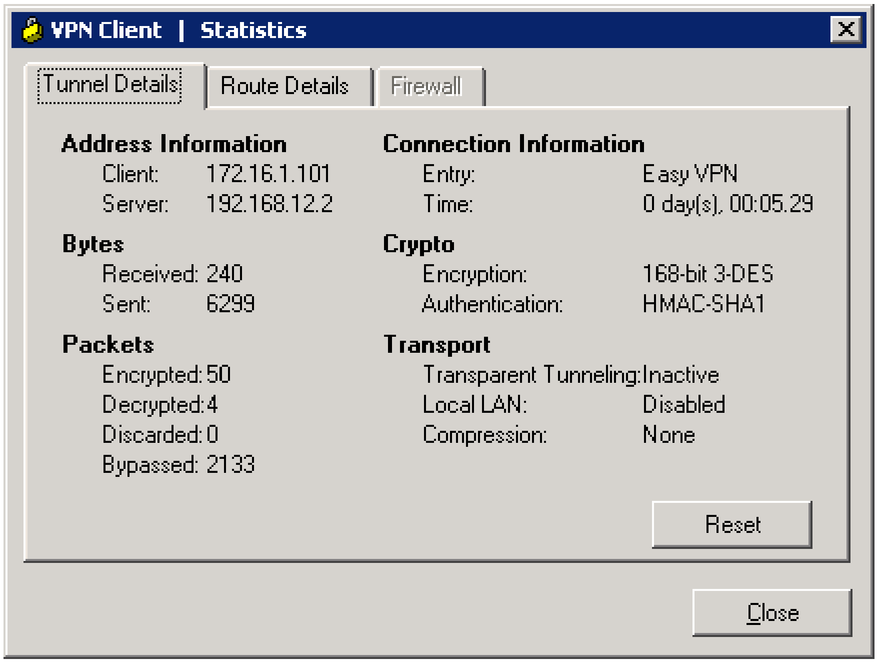
You can also see that your connection has populated the log window with information. After reviewing the information here, click **Close** to close this window. This logging functionality can be very useful when troubleshooting VPN client problems.



```
52    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x6370000F
Added key with SPI=0xa5a4d523 into key list

53    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x63700010
Created a new key structure

54    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x6370000F
Added key with SPI=0xf158ba06 into key list

55    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x6370002F
Assigned VA private interface addr 172.16.1.101

56    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x63700037
Configure public interface: 192.168.10.50. SG: 192.168.12.2

57    01:34:42.828  01/24/07  Sev=Info/4          IKE/0x63000013
SENDING >>> ISAKMP OAK INFO *(HASH, NOTIFY:DPD_REQUEST) to
192.168.12.2

58    01:34:42.828  01/24/07  Sev=Info/6          IKE/0x6300003D
Sending DPD request to 192.168.12.2, our seq# = 3983852993

59    01:34:42.859  01/24/07  Sev=Info/5          IKE/0x6300002F
Received ISAKMP packet: peer = 192.168.12.2

60    01:34:42.859  01/24/07  Sev=Info/4          IKE/0x63000014
RECEIVING <<< ISAKMP OAK INFO *(HASH, NOTIFY:DPD_ACK) from 192.168.12.2

61    01:34:42.859  01/24/07  Sev=Info/5          IKE/0x63000040
Received DPD ACK from 192.168.12.2, seq# received = 3983852993, seq# expected
= 3983852993
```

Figure 8-11: Log Window, Populated with Connection Messages

To view VPN connection statistics, right-click the padlock icon in the system tray and click **Statistics...**.



Address Information
Client: 172.16.1.101
Server: 192.168.12.2

Bytes
Received: 240
Sent: 6299

Packets
Encrypted: 50
Decrypted: 4
Discarded: 0
Bypassed: 2133

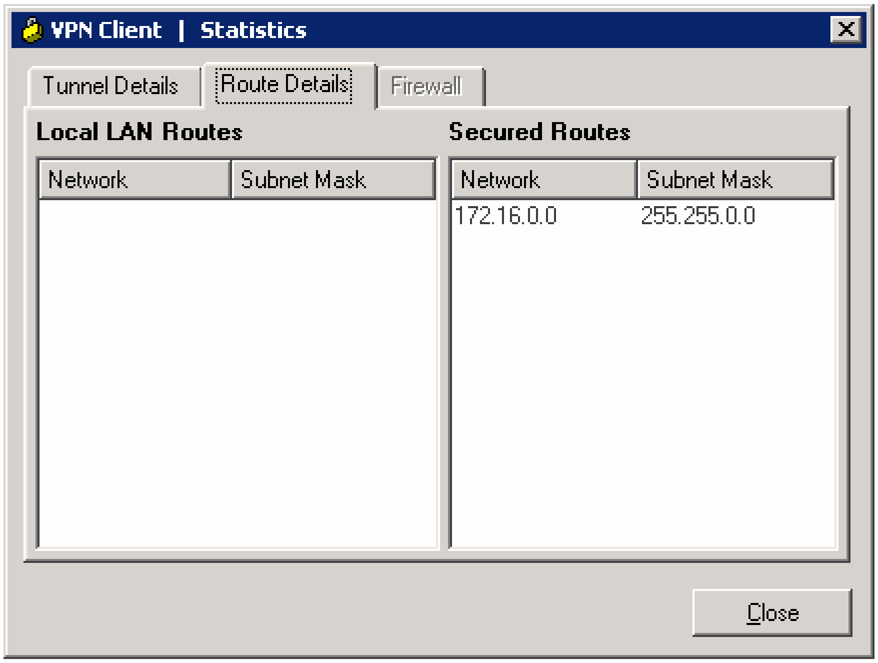
Connection Information
Entry: Easy VPN
Time: 0 day(s), 00:05.29

Crypto
Encryption: 168-bit 3-DES
Authentication: HMAC-SHA1

Transport
Transparent Tunneling: Inactive
Local LAN: Disabled
Compression: None

Figure 8-12: VPN Client Statistics

Click the **Route Details** tab to view routes sent out through split tunneling.



Local LAN Routes

Network	Subnet Mask

Secured Routes

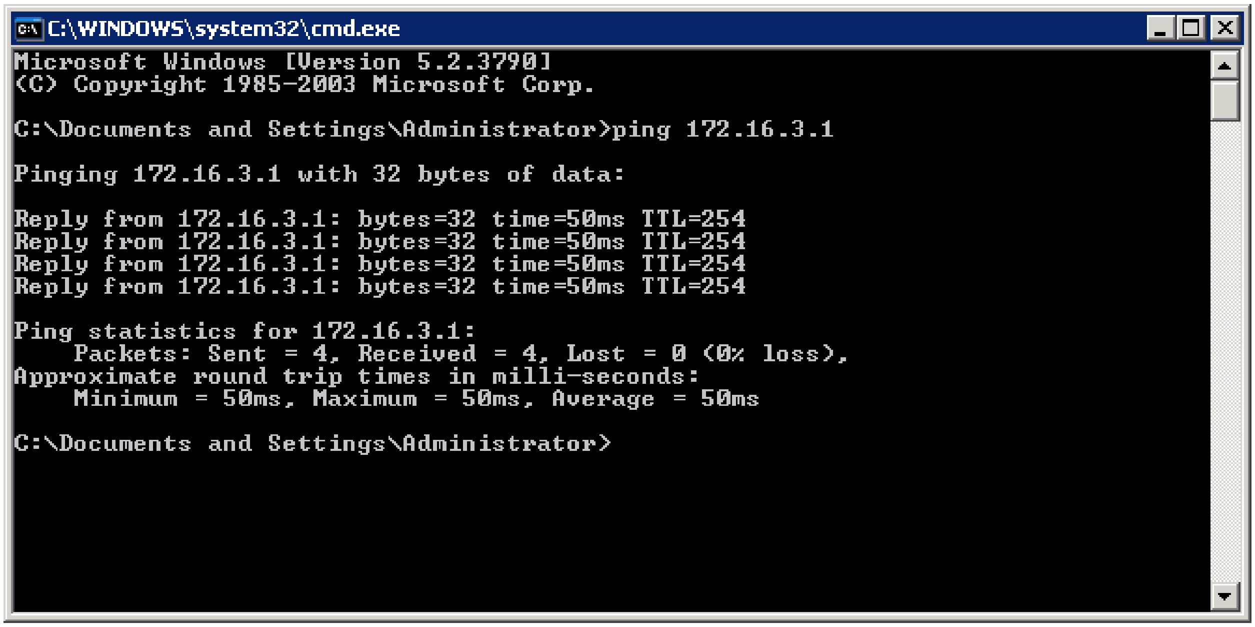
Network	Subnet Mask
172.16.0.0	255.255.0.0

Figure 8-13: Route Details Tab

Close the **Statistics** window when done.

Step 9: Test Network Access with VPN Connectivity

Now that the host has connected to the VPN, open up the command prompt again (see earlier steps if you don’t remember) and ping HQ2’s loopback. This time, it should be successful.

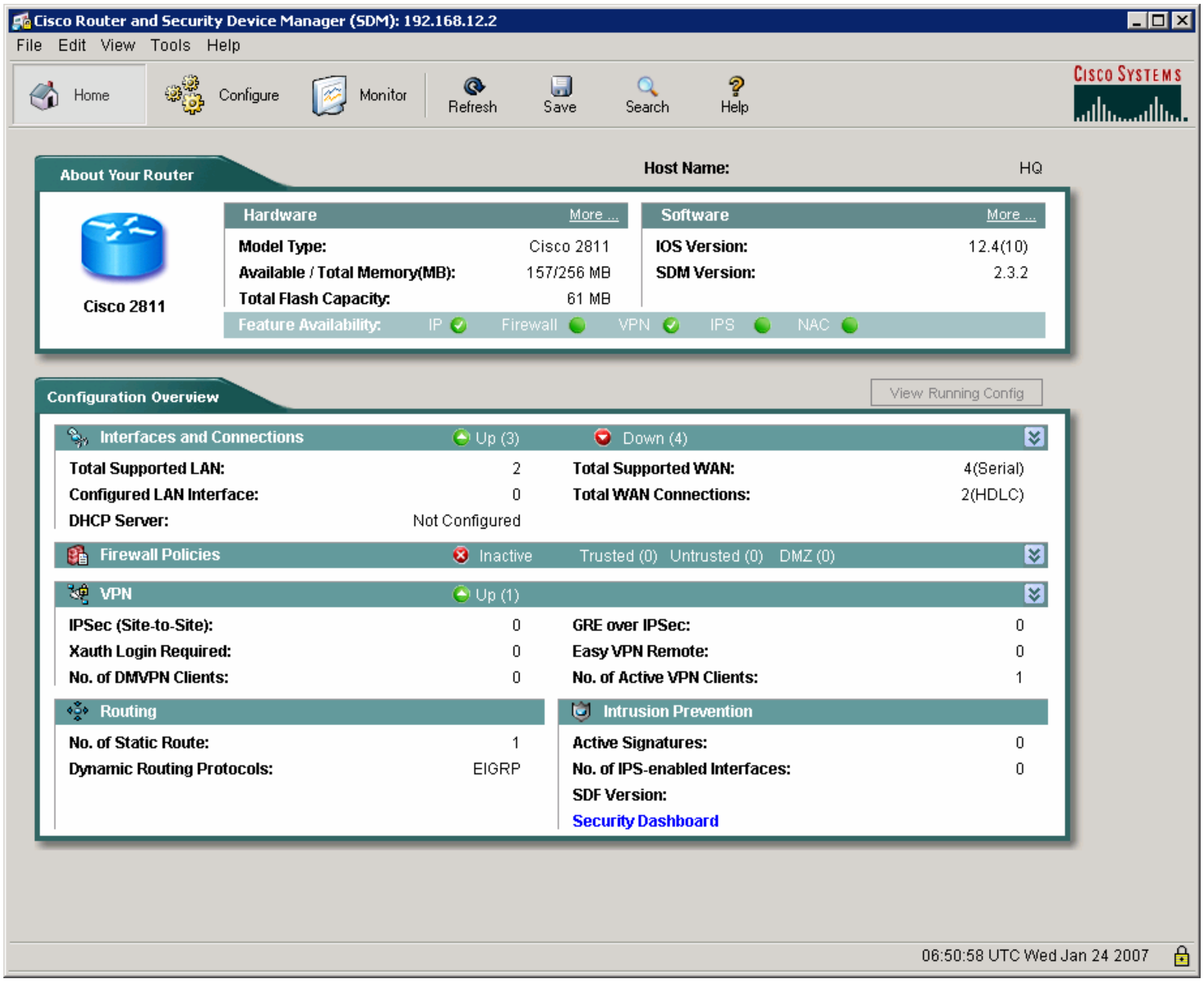


```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 172.16.3.1

Pinging 172.16.3.1 with 32 bytes of data:

Reply from 172.16.3.1: bytes=32 time=50ms TTL=254
Reply from 172.16.3.1: bytes=32 time=50ms TTL=254
Reply from 172.16.3.1: bytes=32 time=50ms TTL=254
Reply from 172.16.3.1: bytes=32 time=50ms TTL=254

Ping statistics for 172.16.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 50ms, Maximum = 50ms, Average = 50ms

C:\Documents and Settings\Administrator>
```

Figure 9-1: Successful Pings With VPN

Step 10: Verify Easy VPN Functionality with SDM

While connected through the VPN, open up SDM again on the host and connect to HQ. This time you can connect to any interface on HQ, not just the external one, because you are inside the VPN. Note on the home screen the number of active VPN clients under the VPN section.



Host Name: HQ

Cisco 2811

Hardware

Model Type: Cisco 2811
Available / Total Memory(MB): 157/256 MB
Total Flash Capacity: 61 MB

Software

IOS Version: 12.4(10)
SDM Version: 2.3.2

Feature Availability: IP ✓ Firewall ● VPN ✓ IPS ● NAC ●

Interfaces and Connections Up (3) Down (4)

Total Supported LAN: 2
Configured LAN Interface: 0
DHCP Server: Not Configured
Total Supported WAN: 4(Serial)
Total WAN Connections: 2(HDLC)

Firewall Policies ☒ Inactive Trusted (0) Untrusted (0) DMZ (0)

VPN Up (1)

IPSec (Site-to-Site): 0
Xauth Login Required: 0
No. of DMVPN Clients: 0
GRE over IPSec: 0
Easy VPN Remote: 0
No. of Active VPN Clients: 1

Routing

No. of Static Route: 1
Dynamic Routing Protocols: EIGRP

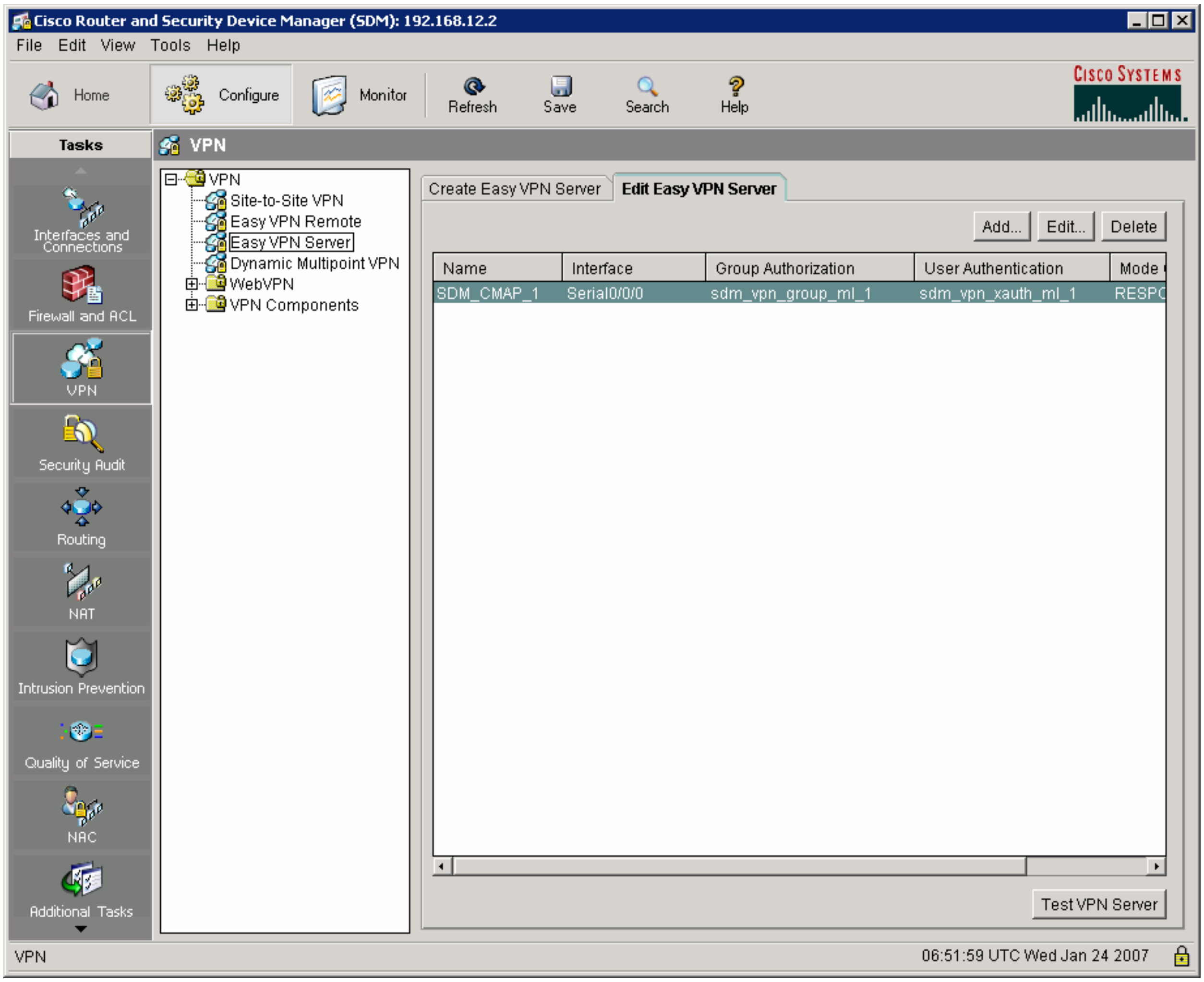
Intrusion Prevention

Active Signatures: 0
No. of IPS-enabled Interfaces: 0
SDF Version:
Security Dashboard

06:50:58 UTC Wed Jan 24 2007

Figure 10-1: SDM Home Screen

Click the **Configure** icon, and then click **VPN** on the left side bar. Choose **Easy VPN Server** from the VPN types. Click the **Edit Easy VPN Server** tab, and then click the **Test VPN Server** button.



Name	Interface	Group Authorization	User Authentication	Mode (
SDM_CMAP_1	Serial0/0/0	sdm_vpn_group_ml_1	sdm_vpn_xauth_ml_1	RESPC

VPN

06:51:59 UTC Wed Jan 24 2007

Figure 10-2: Edit Easy VPN Server Tab

Click **Start**.

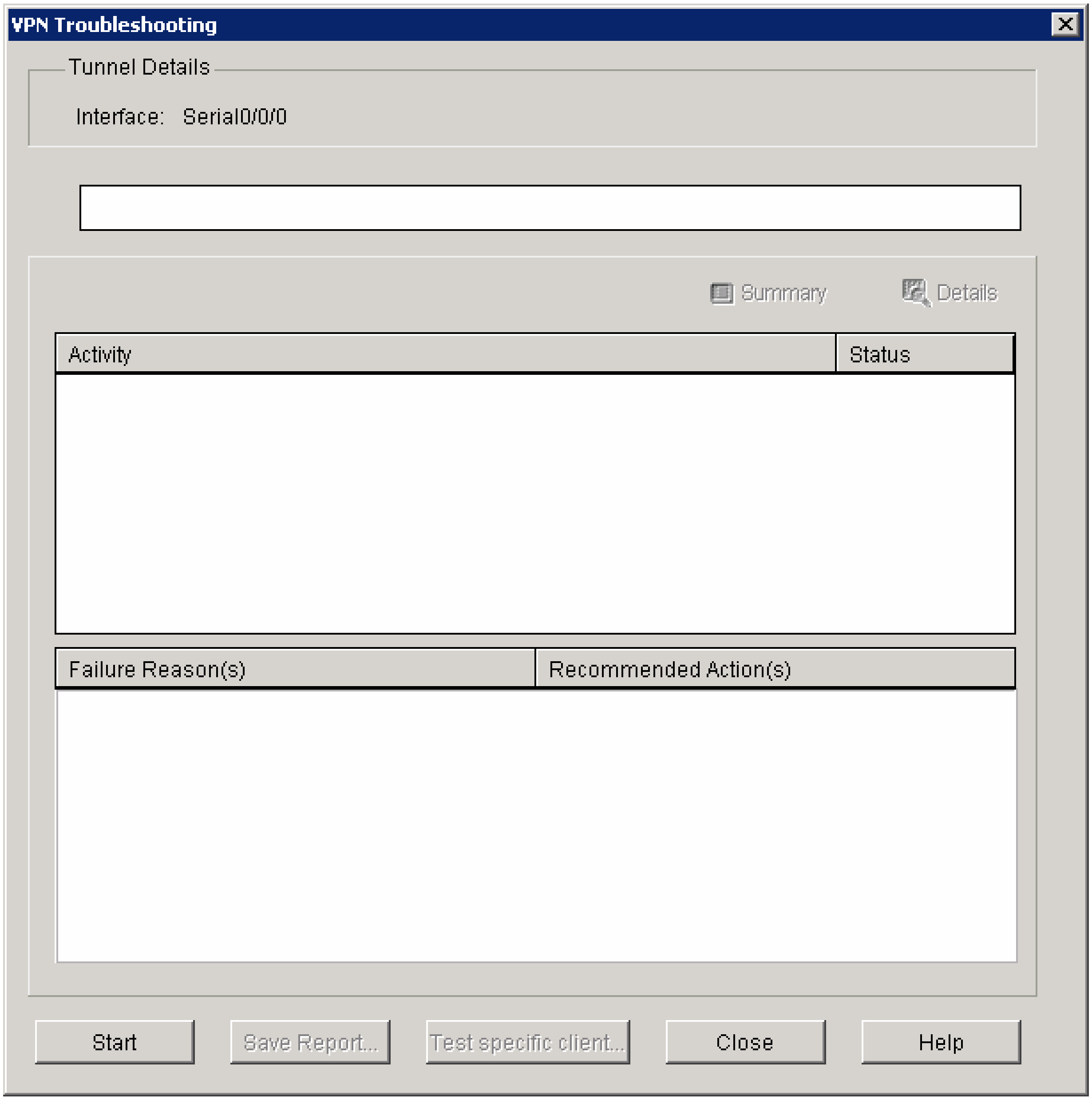
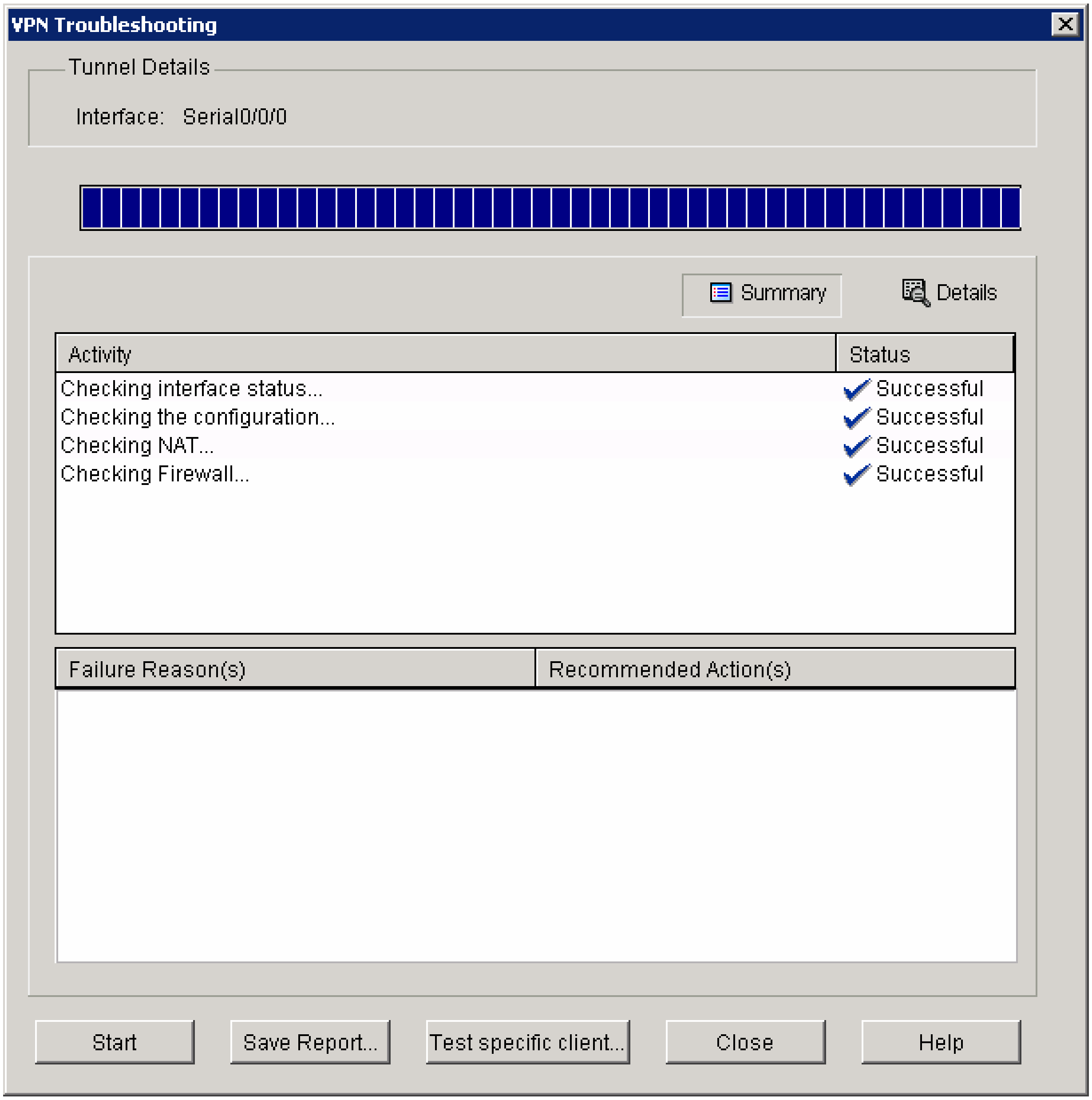


Figure 10-3: VPN Testing Window

The tests should be successful.



Tunnel Details

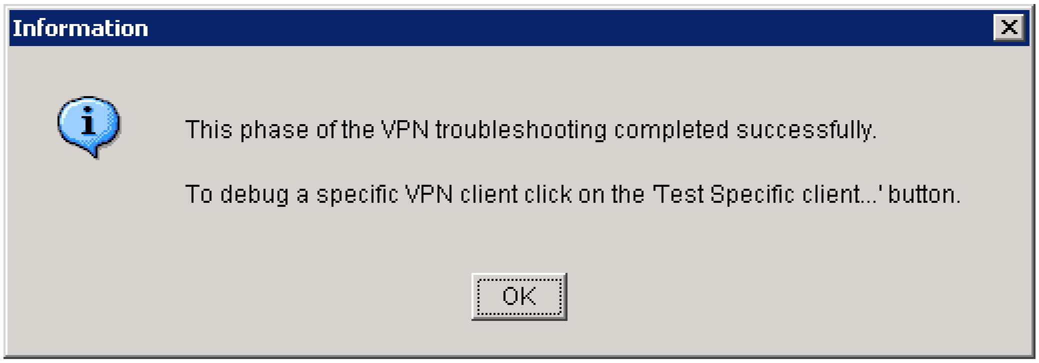
Interface: Serial0/0/0

Activity	Status
Checking interface status...	✓ Successful
Checking the configuration...	✓ Successful
Checking NAT...	✓ Successful
Checking Firewall...	✓ Successful

Failure Reason(s)	Recommended Action(s)

Figure 10-4: VPN Test In Progress

Click **OK** once the success message appears.



This phase of the VPN troubleshooting completed successfully.

To debug a specific VPN client click on the 'Test Specific client...' button.

Figure 10-5: Successful VPN Test Status Window

Click **Close** when you are finished, and then close SDM.

Step 11: Disconnecting the VPN Client

Right-click the padlock icon in the system tray, and click **Disconnect**. The VPN client will disconnnect.

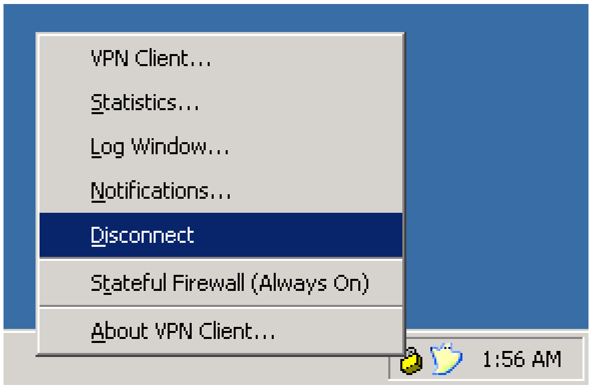


Figure 11-1: Disconnecting from the VPN via the System Tray Icon

The padlock should first change to a padlock with an 'X' through it, indicating that it is disconnecting. It will change to an unlocked icon, indicating no VPN connection. Finally, right-click on the padlock and click **Exit** to quit the VPN client.

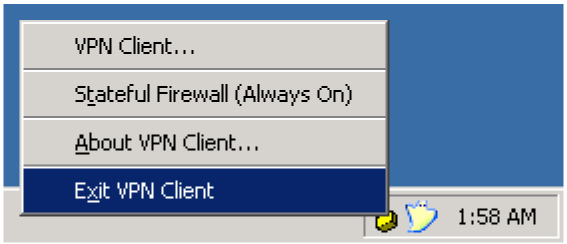


Figure 11-2: Exiting the VPN Client via the System Tray Icon

Final Configurations

```
ISP# show run
hostname ISP
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 ip address 192.168.12.1 255.255.255.0
 clock rate 64000
 no shutdown
end

HQ# show run
hostname HQ
!
aaa new-model
!
aaa authentication login default local
aaa authentication login sdm_vpn_xauth_ml_1 local
aaa authorization exec default local
aaa authorization network sdm_vpn_group_ml_1 local
!
aaa session-id common
!
crypto pki trustpoint TP-self-signed-3043721146
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-3043721146
 revocation-check none
 rsakeypair TP-self-signed-3043721146
!
crypto pki certificate chain TP-self-signed-3043721146
 certificate self-signed 01
  3082023A 308201A3 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303433 37323131 3436301E 170D3037 30313234 30343437
  32365A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D33 30343337
  32313134 3630819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100ADBE 1C08ACA4 0AF3D3FF 11F49933 1AC172FE 3D3D40A6 3AB342FF B952D3E2
  0F203935 83E9C1C0 E0B14B0B C44EF57E A9D7252E F8052060 8D194C9F 84BA3BE4
  F004217A 09B4A9E7 EFBD0D8C BA420B55 6055B135 ED9A33E5 D4294415 BC453756
  AB458059 4E6E23A4 159A87C1 E92F8AB3 E4C7BA5F 434C1BE0 9BF59A78 08961B55
  F0DD0203 010001A3 62306030 0F060355 1D130101 FF040530 030101FF 300D0603
  551D1104 06300482 02485130 1F060355 1D230418 30168014 5BCB0C4C C995CEA2
  F7E9667E DC80525B BB481946 301D0603 551D0E04 1604145B CB0C4CC9 95CEA2F7
  E9667EDC 80525BBB 48194630 0D06092A 864886F7 0D010104 05000381 81008FFA
  728302E8 CA86686E 5394BA3A C8260F99 75CA12D4 3B86EAF2 EE3F9AB5 E5D18FEA
  FC495B41 C716BEF5 82A0F21C 7D085C01 EEFE4302 BA666344 D0D51346 9BDB4AD0
  94B91A93 FEB44001 E50D3BFF 9479456F D2658D25 8BE61405 2AA5229A 3AFF2096
  ECDD7C61 3EB564C8 9608CA67 2A3CC3D6 B7A5B918 863E901E E2ABBD0D 279A
  quit
username ciscosdm privilege 15 password 0 ciscosdm
username ciscouser password 0 ciscouser
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp client configuration group ciscogroup
```

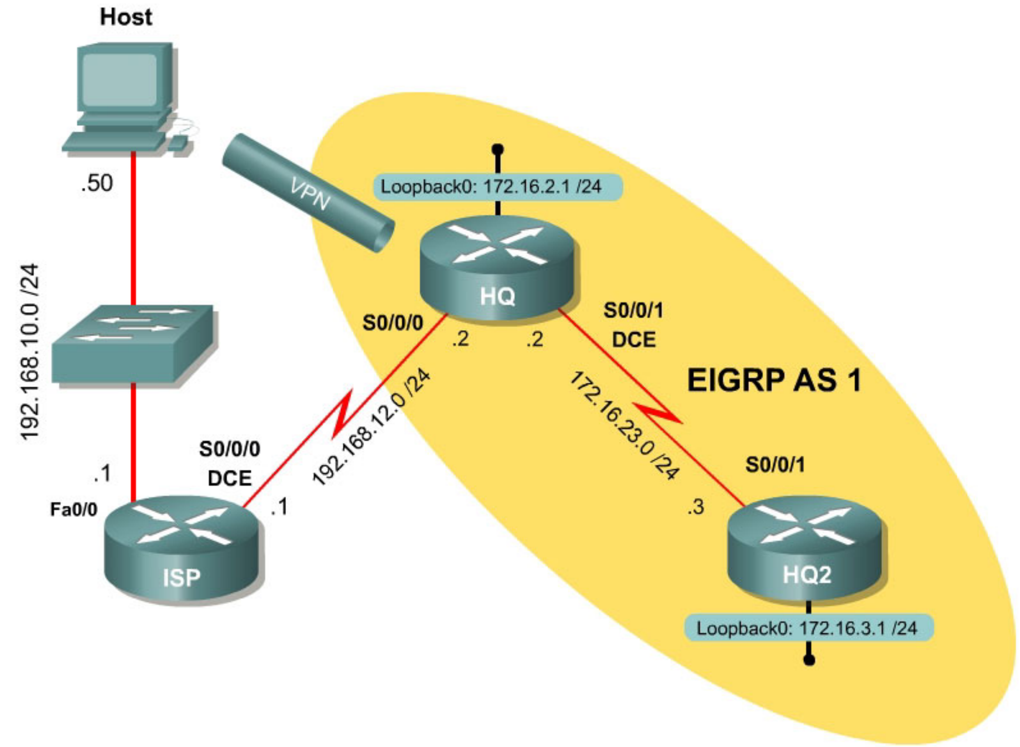
```
 key ciscogroup
 pool SDM_POOL_1
 acl 100
 netmask 255.255.255.0
!
crypto ipsec transform-set ESP-3DES-SHA esp-3des esp-sha-hmac
!
crypto dynamic-map SDM_DYNMAP_1 1
 set security-association idle-time 28800
 set transform-set ESP-3DES-SHA
 reverse-route
!
crypto map SDM_CMAP_1 client authentication list sdm_vpn_xauth_ml_1
crypto map SDM_CMAP_1 isakmp authorization list sdm_vpn_group_ml_1
crypto map SDM_CMAP_1 client configuration address respond
crypto map SDM_CMAP_1 65535 ipsec-isakmp dynamic SDM_DYNMAP_1
!
interface Loopback0
 ip address 172.16.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.12.2 255.255.255.0
 crypto map SDM_CMAP_1
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.23.2 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 redistribute static
 network 172.16.0.0
 no auto-summary
!
ip local pool SDM_POOL_1 172.16.1.100 172.16.1.200
ip route 0.0.0.0 0.0.0.0 192.168.12.1
!
ip http server
ip http authentication local
ip http secure-server
!
access-list 100 remark SDM_ACL Category=4
access-list 100 permit ip 172.16.0.0 0.0.255.255 any
!
line vty 0 4
 transport input telnet ssh
end

HQ2# show run
hostname HQ2
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.16.23.3 255.255.255.0
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
end
```

Lab 3.10 Configuring Easy VPN with the IOS CLI

Learning Objectives

- Configure EIGRP on a router
- Configure Easy VPN Server
- Install the Cisco VPN Client to a host
- Connect to the VPN using Cisco VPN client
- Verify VPN operation

Topology Diagram



Scenario

In this lab, you will set up Easy VPN for the International Travel Agency and connect to Headquarters (HQ) using the IOS Command Line Interface (CLI). The host will simulate an employee connecting from home over the Internet. ISP will simulate an internet router representing the Internet connection for both the home user and the company headquarters.

Step 1: Configure Addressing

Configure the loopback interfaces with the addresses shown in the diagram. Also configure the serial interfaces shown in the diagram. Set the clockrate on the appropriate interfaces and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command. Do not set up the tunnel interface.

```
ISP# configure terminal
ISP(config)# interface fastethernet 0/0
ISP(config-if)# ip address 192.168.10.1 255.255.255.0
ISP(config-if)# no shutdown
ISP(config-if)# interface serial 0/0/0
ISP(config-if)# ip address 192.168.12.1 255.255.255.0
ISP(config-if)# clockrate 64000
ISP(config-if)# no shutdown

HQ# configure terminal
HQ(config)# interface loopback 0
HQ(config-if)# ip address 172.16.2.1 255.255.255.0
HQ(config-if)# interface serial0/0/0
HQ(config-if)# ip address 192.168.12.2 255.255.255.0
HQ(config-if)# no shutdown
HQ(config-if)# interface serial 0/0/1
HQ(config-if)# ip address 172.16.23.2 255.255.255.0
HQ(config-if)# clockrate 64000
HQ(config-if)# no shutdown

HQ2# configure terminal
HQ2(config)# interface loopback 0
HQ2(config-if)# ip address 172.16.3.1 255.255.255.0
HQ2(config-if)# interface serial 0/0/1
HQ2(config-if)# ip address 172.16.23.3 255.255.255.0
HQ2(config-if)# no shutdown
```

Step 2: Configure EIGRP AS 1

Configure EIGRP for AS1 on HQ and HQ2. Add the entire 172.16.0.0/16 major network and disable automatic summarization. ISP will not participate in this routing process.

```
HQ(config)# router eigrp 1
HQ(config-router)# no auto-summary
HQ(config-router)# network 172.16.0.0

HQ2(config)# router eigrp 1
HQ2(config-router)# no auto-summary
HQ2(config-router)# network 172.16.0.0
```

An EIGRP neighbor adjacency should form between HQ and HQ2. If not, troubleshoot by checking your interface configuration, EIGRP configuration, and physical connectivity.

Step 3: Configure a Static Default Route

Since ISP represents a connection to the Internet, send all traffic whose destination network does not exist in the routing tables at company headquarters out this connection via a default route. This route can be statically created on HQ, but will need to be redistributed into EIGRP so HQ2 will learn the route too.

```
HQ(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.1
HQ(config)# router eigrp 1
HQ(config-router)# redistribute static
```

For which types of routes is it unnecessary to assign a default/seed metric when redistributing into EIGRP?

How else could you configure HQ to advertise the default route?

Step 4: Enable AAA on HQ

To run Easy VPN server, AAA must be enabled on the router. To prevent getting locked out of the router, create a local username and make sure that authentication is performed through the local database. HQ will be the Easy VPN Server, so this is where it must be configured.

```
HQ(config)# username cisco password cisco
HQ(config)# aaa new-model
HQ(config)# aaa authentication login default local none
```

Step 5: Create the IP Pool

Create a pool that VPN clients will draw their IP addresses from using the command **ip local pool** *name low-address high-address*. Use addresses from 172.16.2.100 to 172.16.2.200.

```
HQ(config)# ip local pool VPNCLIENTS 172.16.2.100 172.16.2.200
```

Step 6: Configure the Group Authorization

Use the AAA authorization command **aaa authorization network** *name types* to configure the VPN group authentication list. This list will authenticate remote users connecting to the VPN using the group set up in their client. Use the local group list, which will be configured in the next step.

```
HQ(config)# aaa authorization network VPNAUTH local
```

Step 7: Create an IKE Policy and Group

Just like previous crypto configurations, you must set up ISAKMP policies to be used during IKE Phase I negotiation. Use the following settings. If your version of the IOS does not support the same settings that appear here, try to make them as similar as possible.

```
HQ(config)# crypto isakmp policy 10
HQ(config-isakmp)# authentication pre-share
HQ(config-isakmp)# encryption aes 256
HQ(config-isakmp)# group 2
```

Since you don’t know specific peers, you cannot statically associate ISAKMP keys with IP addresses or hosts. Rather, VPN clients could connect from anywhere on the internet. So, we configure an ISAKMP client group, instead. The group will exist locally on the router, as specified by the AAA network authorization command in the last step.

To enter the ISAKMP group configuration mode, use the global configuration command **crypto isakmp client configuration group** *name*. Use the name “ciscogroup”. Use **?** once in this mode to find out what options you have available.

```
HQ(config)# crypto isakmp client configuration group ciscogroup
HQ(config-isakmp-group)# ?
ISAKMP group policy config commands:
  access-restrict    Restrict clients in this group to an interface
  acl                Specify split tunneling inclusion access-list number
  backup-gateway     Specify backup gateway
  dns                Specify DNS Addresses
  domain             Set default domain name to send to client
  exit               Exit from ISAKMP client group policy configuration mode
  firewall           Enforce group firewall feature
  group-lock         Enforce group lock feature
  include-local-lan  Enable Local LAN Access with no split tunnel
  key                pre-shared key/IKE password
  max-logins         Set maximum simultaneous logins for users in this group
  max-users          Set maximum number of users for this group
  netmask            netmask used by the client for local connectivity
  no                 Negate a command or set its defaults
  pfs                The client should propose PFS
  pool               Set name of address pool
  save-password      Allows remote client to save XAUTH password
  split-dns          DNS name to append for resolution
  wins               Specify WINS Addresses
```

Configure a pre-shared key to be the same as the group name. Also, associate the address pool created earlier with this group. In addition, specify an access list to be used as the split tunneling list, to enable split tunneling in this configuration. This access list doesn't exist yet, but you will create it shortly. Finally, set the network mask, since the IP pool does not specify one.

```
HQ(config-isakmp-group)# key ciscogroup
HQ(config-isakmp-group)# pool VPNCLIENTS
HQ(config-isakmp-group)# acl 100
HQ(config-isakmp-group)# netmask 255.255.255.0
```

Now that you have referenced the split tunneling access-list, create it. Source networks permitted by the access-list will be sent to the VPN clients and injected into their IP table. Create an access list allowing traffic sourced from the entire 172.16.0.0/16 network.

```
HQ(config)# access-list 100 permit ip 172.16.0.0 0.0.255.255 any
```

Step 8: Configure the IPsec Transform Set

Configure an IPsec transform set for use with the VPN. Use the 3DES algorithm for encryption and the SHA-HMAC hash function for data integrity.

```
HQ(config)# crypto ipsec transform-set mytrans esp-3des esp-sha-hmac
HQ(cfg-crypto-trans)# exit
```

Step 9: Create a Dynamic Crypto Map

As in previous IPsec configurations, you must set up a crypto map. However, this type of crypto map will be different than earlier configurations. Previously you configured static crypto maps, which configure certain traffic to establish VPNs with certain peers. However, since you don't know what the peers will be, as well as what the triggering traffic will be, create and apply a dynamic crypto map.

Use the global configuration command **crypto dynamic-map** *name sequence*, similar to a regular crypto map. Use the name "mymap" and the sequence number 10. Once you have entered crypto map configuration mode, set the transform set you configured in the previous step. Use the command **reverse-route,** which ensures that a route is installed on the local router for the remote VPN peer.

```
HQ(config)# crypto dynamic-map mymap 10
HQ(config-crypto-map)# set transform-set mytrans
HQ(config-crypto-map)# reverse-route
```

After creating the map, there are more commands that modify it. The first of these will make the map respond to VPN requests, which is the **crypto map** *name* **client configuration address respond** command. The next command is **crypto map** *name* **isakmp authorization list** *name*, which associates an AAA group authorization list with the map. The final command creates a regular crypto map using the dynamic one created earlier.

```
HQ(config)# crypto map mymap client configuration address respond
HQ(config)# crypto map mymap isakmp authorization list VPNAUTH
HQ(config)# crypto map mymap 10 ipsec-isakmp dynamic mymap
```

Finally, apply the crypto map to the interface that is facing ISP.

```
HQ(config)#int serial0/0/0
HQ(config-if)#crypto map mymap
```

Step 10: Enable IKE DPD and User Authentication

IKE Dead Peer Detection (DPD) is a keepalive mechanism for checking VPN connections. This is beneficial when a VPN server has to manage many connections that are potentially on unstable connections. To configure IKE DPD, use the global configuration command **crypto isakmp keepalive** *seconds retry-time*, where seconds is how often to send a keepalive packet and retry-time is how soon to retry if one is missed. Use a keepalive timer of 30 seconds and a retry-time of 5 seconds.

```
HQ(config)# crypto isakmp keepalive 30 5
```

Xauth, or extended authentication, is the method used to authenticate VPN clients on a per-user basis, in addition to the group authentication. To configure this, use the AAA login authentication command **aaa authentication login** *group types*. We will reuse the name VPNAUTH (the last time we used it, it was for network authentication, not login authentication), and keep the authentication type as local. Also, add a user for VPN access with the username/password of ciscouser/ciscouser.

```
HQ(config)# aaa authentication login VPNAUTH local
HQ(config)# username ciscouser password ciscouser
```

Globally configure the Xauth timeout to be 60 seconds using the **crypto isakmp xauth timeout** *seconds* command. This controls the amount of time that the VPN server will wait before terminating the IKE session with a client if user authentication is not performed.

```
HQ(config)# crypto isakmp xauth timeout 60
```

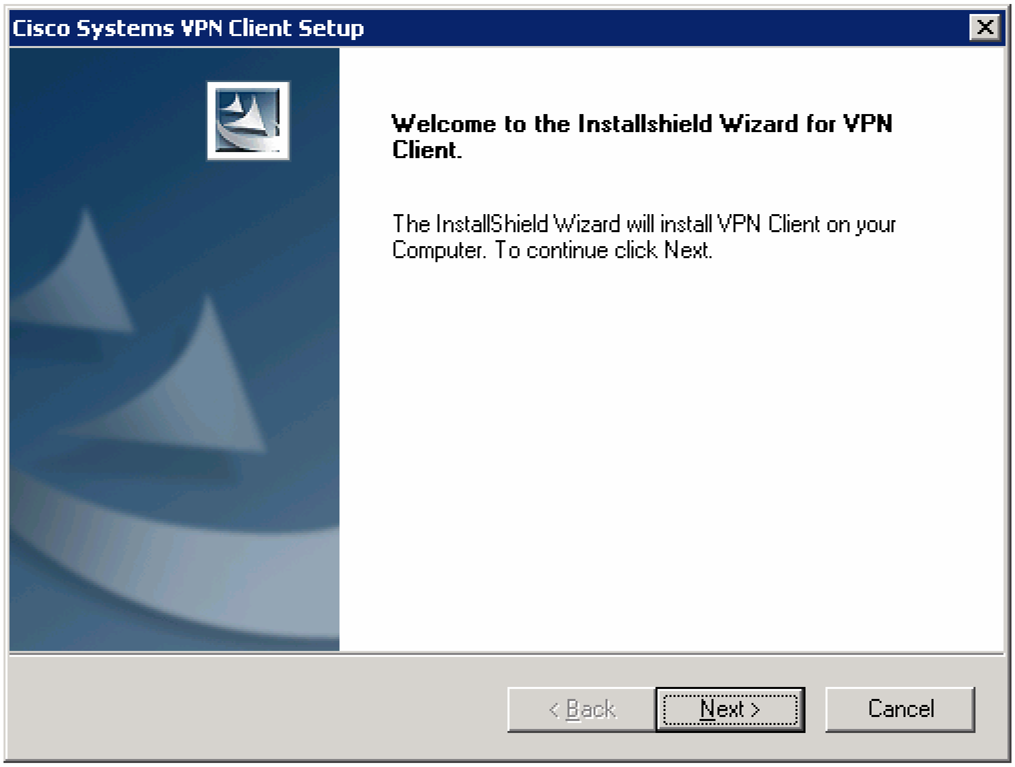
Finally, associate the AAA login list with the crypto map configured earlier.

```
HQ(config)# crypto map mymap client authentication list VPNAUTH
```

Step 11: Install the Cisco VPN Client

Now that HQ has been set up as an Easy VPN Server, the host will change its role from management host to a VPN client connecting across the Internet to HQ. Before you can connect, you must install the Cisco VPN Client if you haven't already. If you have already installed the VPN client, skip this step and move on to Step 12.

To begin the installation, download the VPN Client from Cisco, and extract it to a temporary directory. Run the setup.exe file in the temporary directory to start installation. Click **Next** when the installer welcomes you.

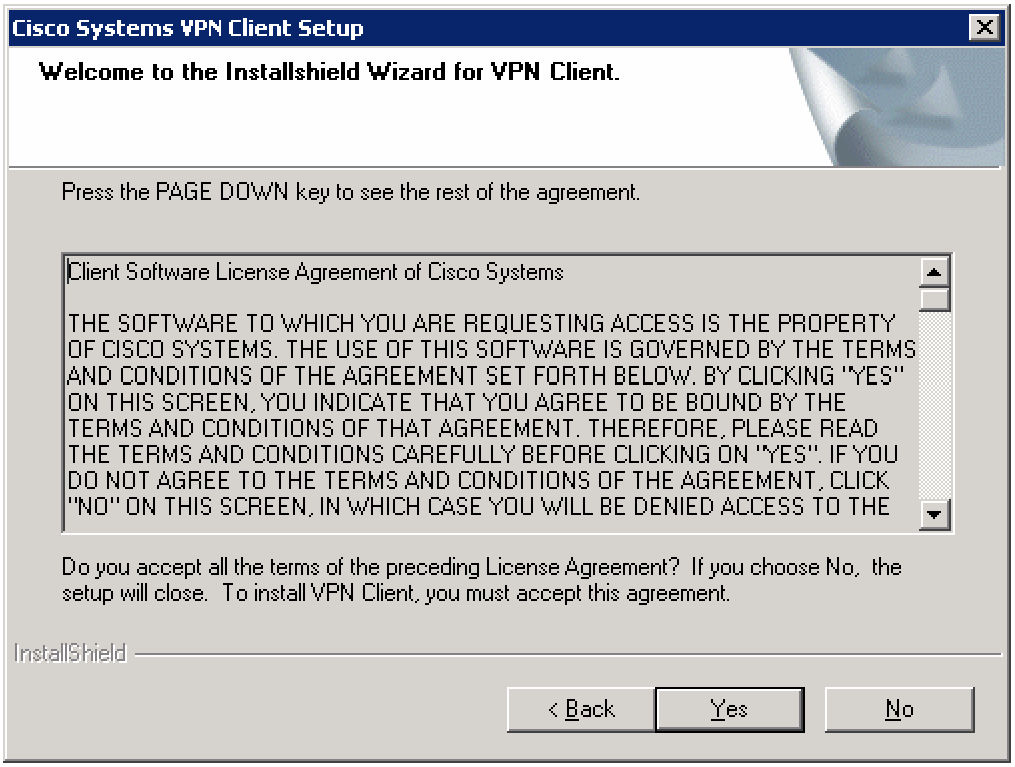


Welcome to the Installshield Wizard for VPN Client.

The InstallShield Wizard will install VPN Client on your Computer. To continue click Next.

Figure 11-1: VPN Client Installation Wizard

Click **Yes** after reading the software license agreement.


Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

Press the PAGE DOWN key to see the rest of the agreement.

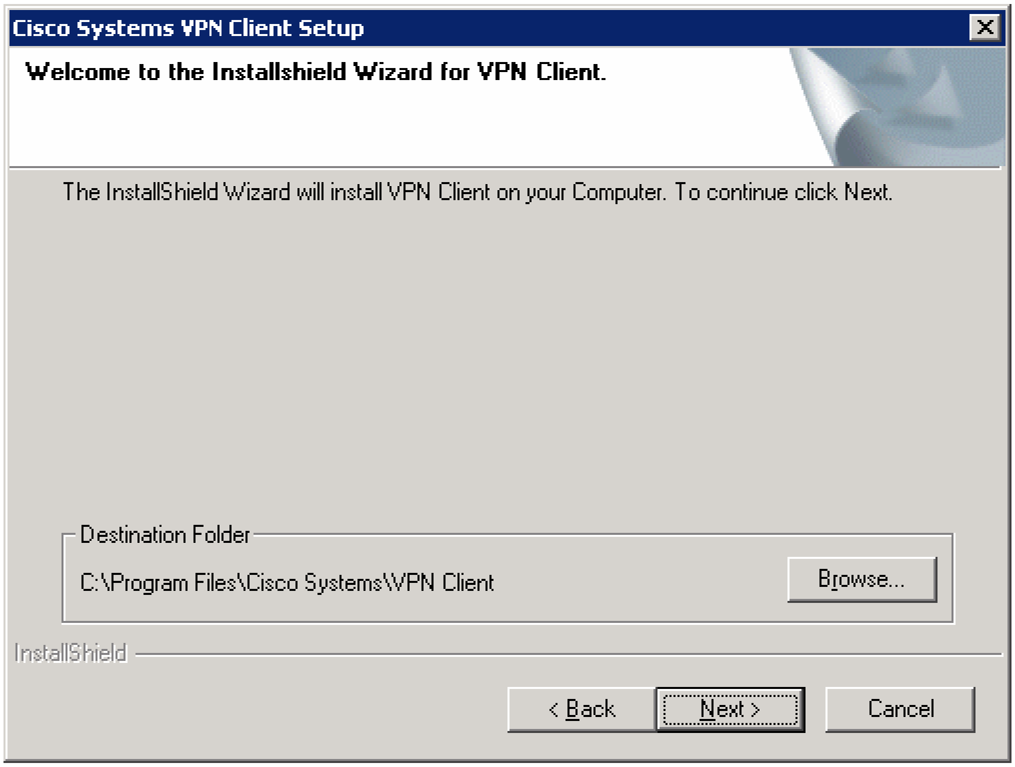
Client Software License Agreement of Cisco Systems

THE SOFTWARE TO WHICH YOU ARE REQUESTING ACCESS IS THE PROPERTY OF CISCO SYSTEMS. THE USE OF THIS SOFTWARE IS GOVERNED BY THE TERMS AND CONDITIONS OF THE AGREEMENT SET FORTH BELOW. BY CLICKING "YES" ON THIS SCREEN, YOU INDICATE THAT YOU AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THAT AGREEMENT. THEREFORE, PLEASE READ THE TERMS AND CONDITIONS CAREFULLY BEFORE CLICKING ON "YES". IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THE AGREEMENT, CLICK "NO" ON THIS SCREEN, IN WHICH CASE YOU WILL BE DENIED ACCESS TO THE

Do you accept all the terms of the preceding License Agreement? If you choose No, the setup will close. To install VPN Client, you must accept this agreement.

Figure 11-2: Cisco VPN Client License Agreement

Click **Next** to use the default installation.



Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

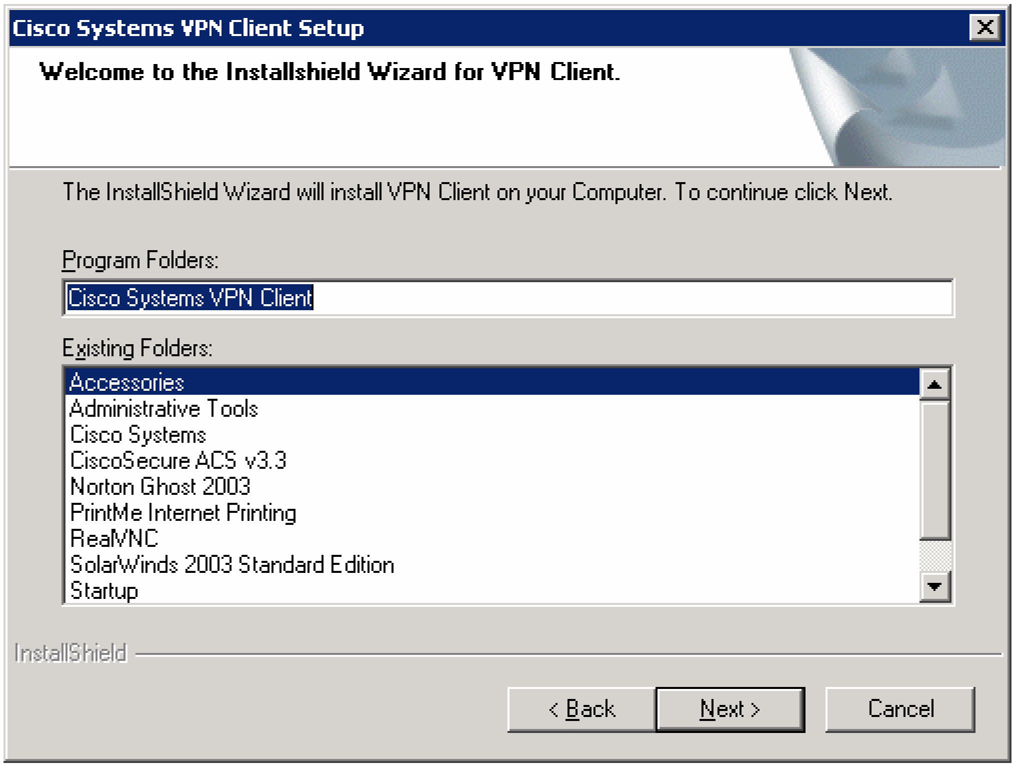
The InstallShield Wizard will install VPN Client on your Computer. To continue click Next.

Destination Folder

C:\Program Files\Cisco Systems\VPN Client

Figure 11-3: VPN Client Installation Location

Choose the default program group and click **Next**.



Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

The InstallShield Wizard will install VPN Client on your Computer. To continue click Next.

Program Folders:

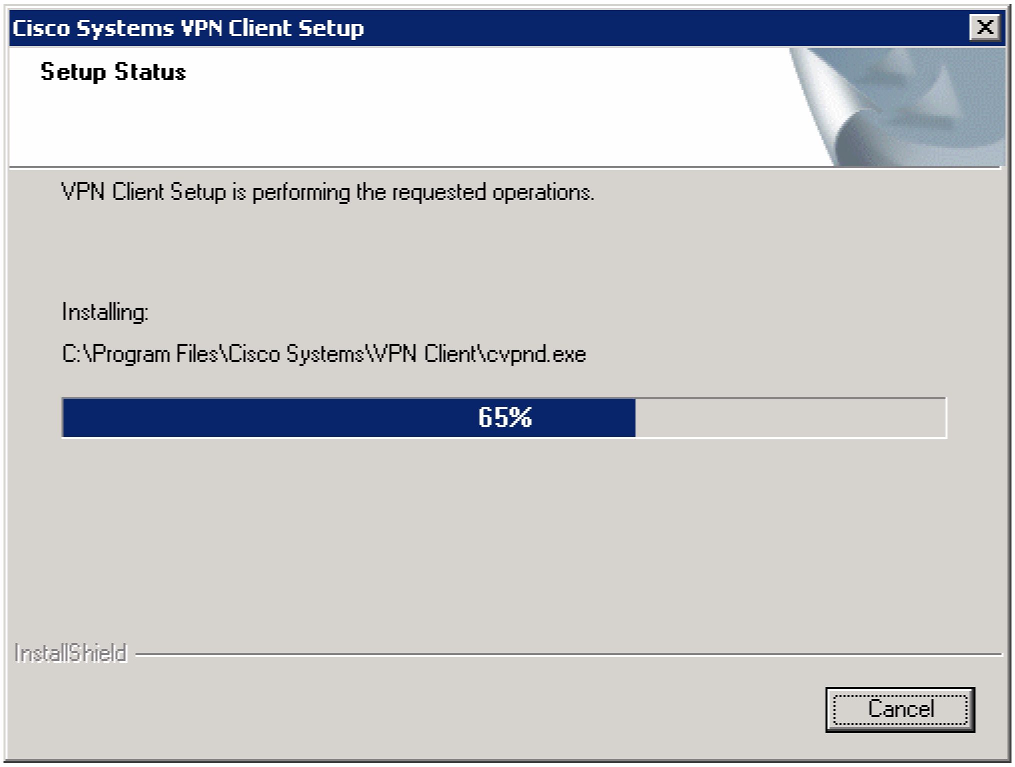
Cisco Systems VPN Client

Existing Folders:

Accessories
Administrative Tools
Cisco Systems
CiscoSecure ACS v3.3
Norton Ghost 2003
PrintMe Internet Printing
RealVNC
SolarWinds 2003 Standard Edition
Startup

Figure 11-4: Start Menu Program Folder Selection

Allow the wizard to install all the necessary files. Toward the end of the process, the wizard will try to add the virtual network interfaces required for VPN use. This may take some time.



Cisco Systems VPN Client Setup

Setup Status

VPN Client Setup is performing the requested operations.

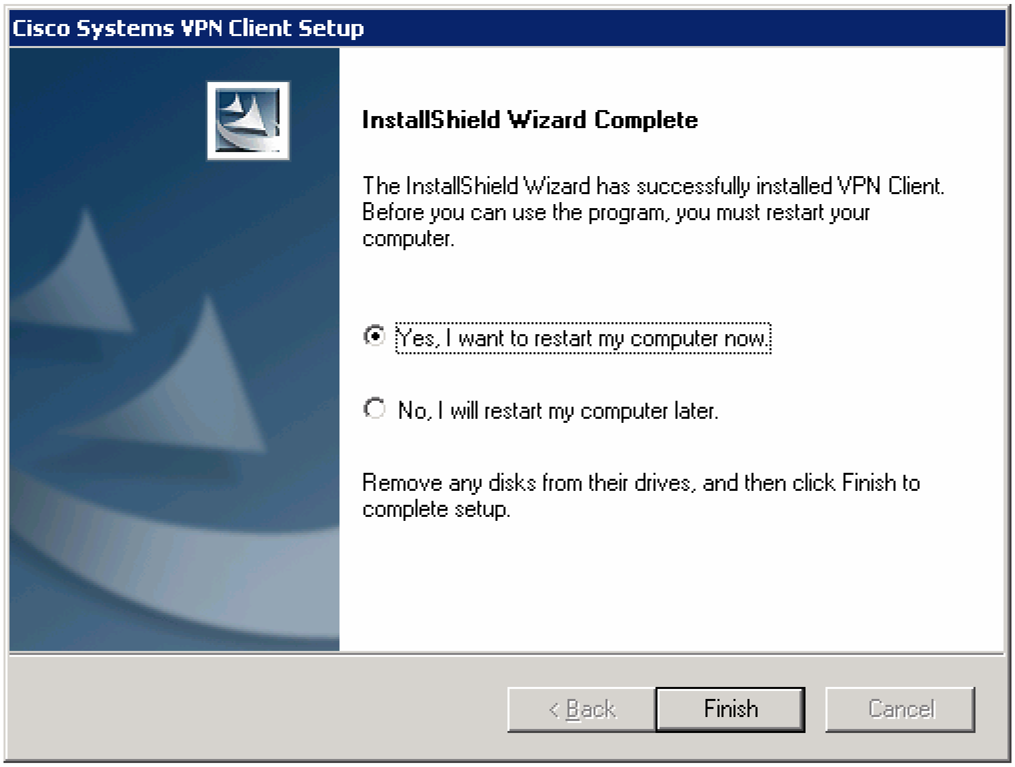
Installing:

C:\Program Files\Cisco Systems\VPN Client\cvpnd.exe

65%

Figure 11-5: VPN Client Installation Progress Indicator

At the end of the installer, you will be required to restart. Click **Finish** to let your computer restart.



Cisco Systems VPN Client Setup

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed VPN Client. Before you can use the program, you must restart your computer.

● Yes, I want to restart my computer now.

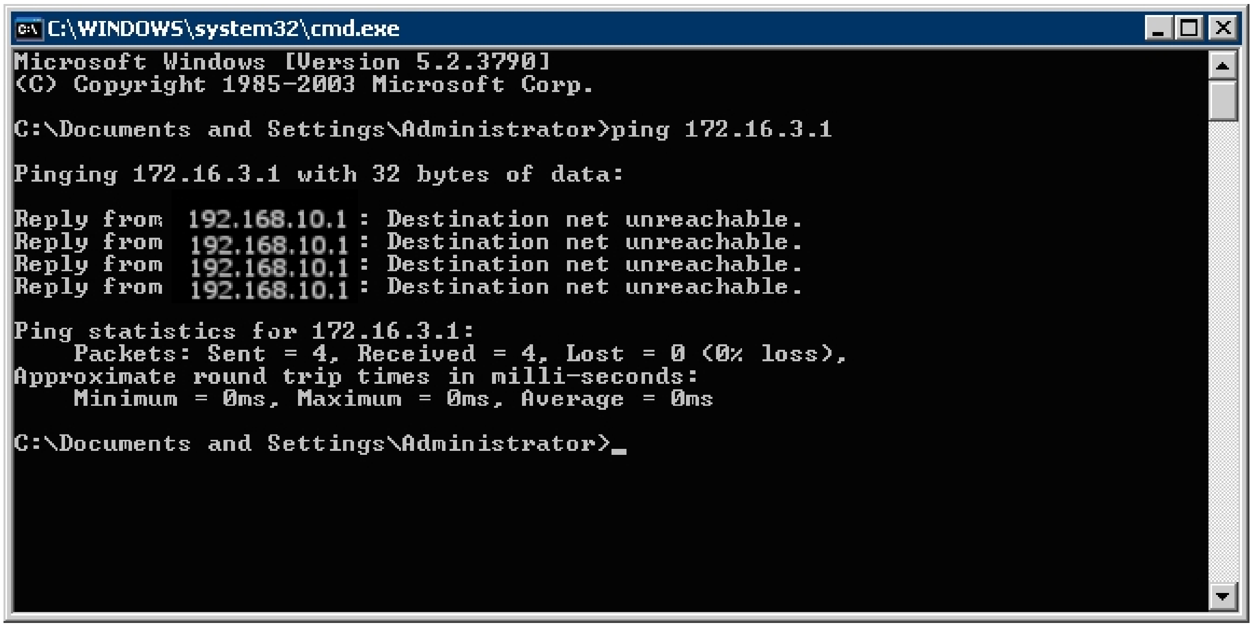
○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

Figure 11-6: Final Installation Wizard Window

Step 12: Test Access from Client without VPN Connection

After restarting the host with the VPN client installed, open up a command prompt. Click on the **Start** button, choose **Run...** and type **cmd,** and click **OK**. Try pinging HQ2’s loopback address. The pings should fail.



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 172.16.3.1

Pinging 172.16.3.1 with 32 bytes of data:

Reply from 192.168.10.1: Destination net unreachable.
Reply from 192.168.10.1: Destination net unreachable.
Reply from 192.168.10.1: Destination net unreachable.
Reply from 192.168.10.1: Destination net unreachable.

Ping statistics for 172.16.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator>_
```

Figure 12-1: Unsuccessful Pings Without VPN

Step 13: Connect to the VPN

To start the Cisco VPN Client, click the **Start** button and choose **Programs > Cisco Systems VPN Client > VPN Client**.

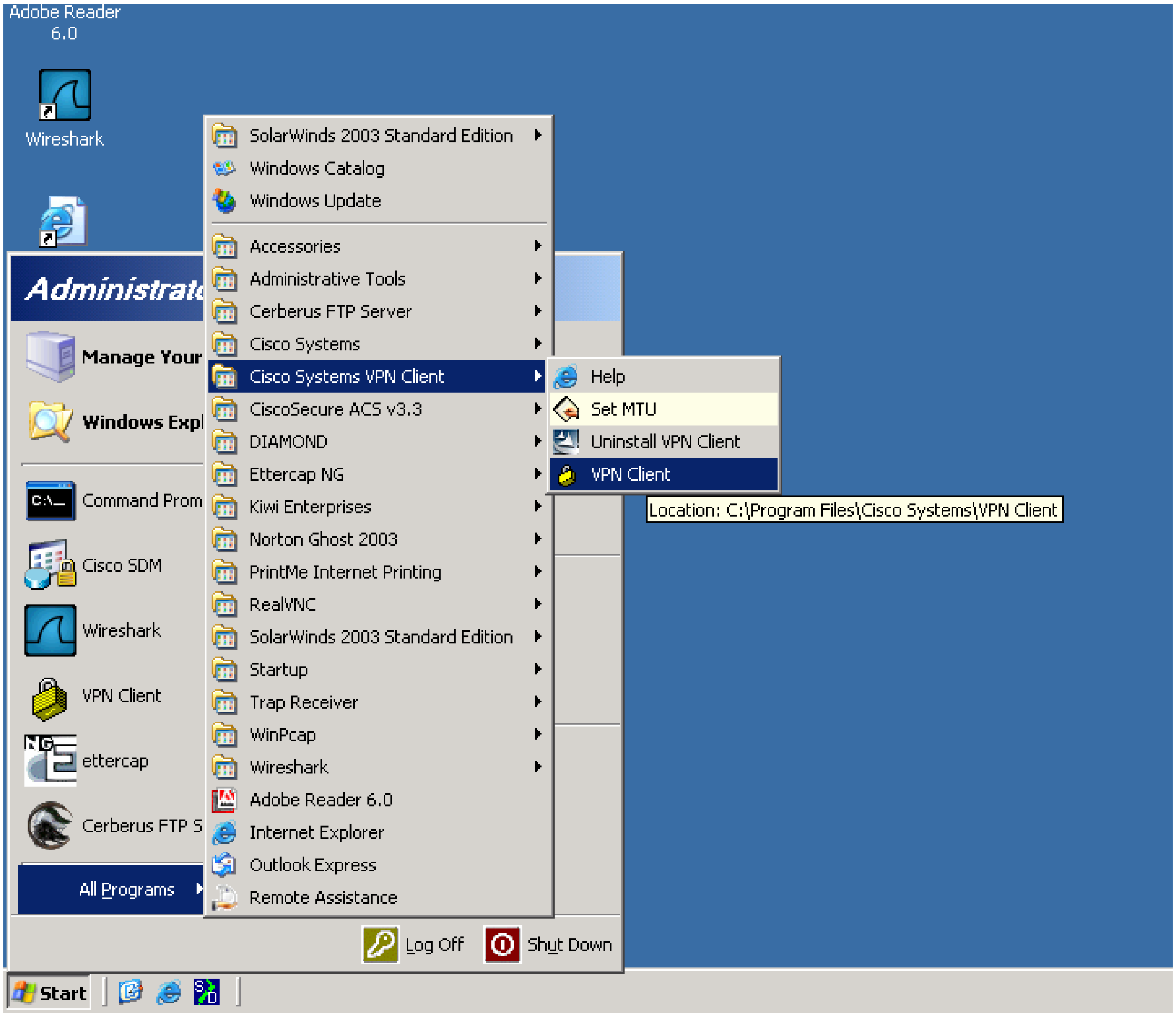


Figure 13-1: Launching the VPN Client

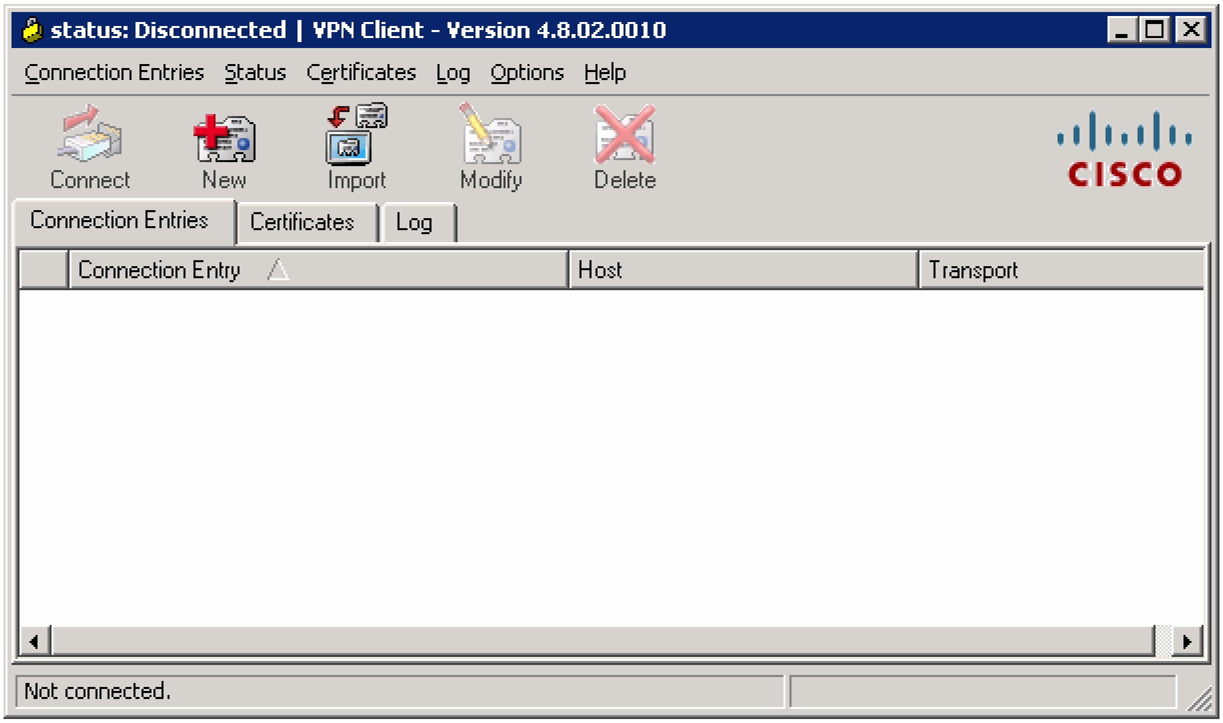
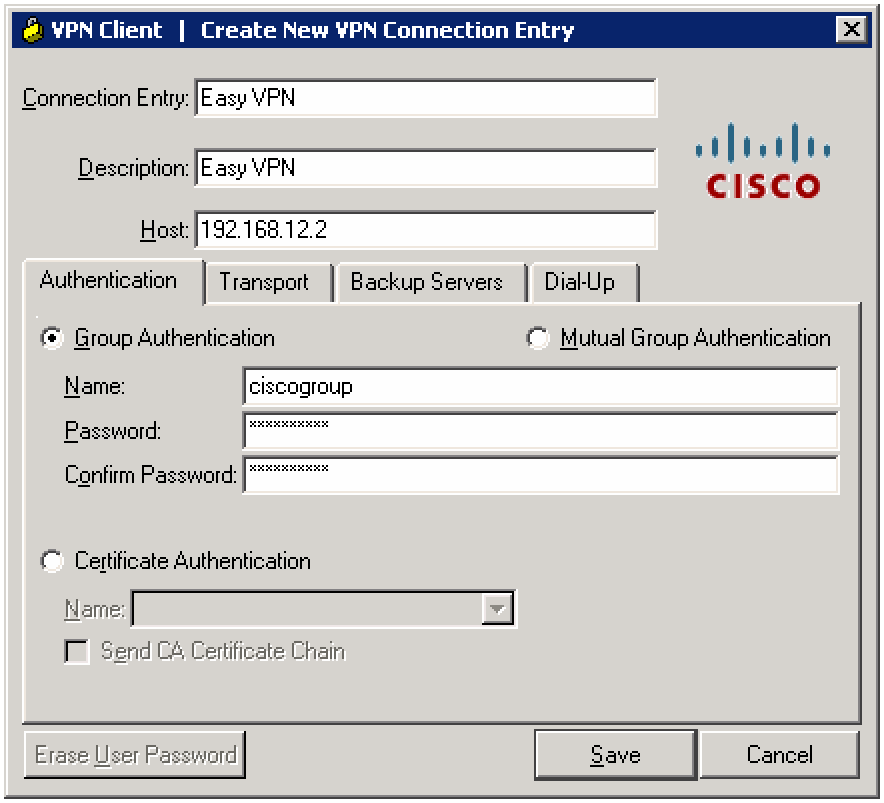


Figure 13-2: VPN Client Application

Once the VPN Client is open, you will need to create a new connection profile to connect to HQ. Click the **New** button. Create the new connection with any name and description you want. For host, enter the IP of HQ’s Serial0/0/0 interface, 192.168.12.2. The host IP address represents the IP address of the VPN server or concentrator to which you wish to connect. In this case, HQ is running the Easy VPN Server and will function as such. Use the group name and password previously configured in the Easy VPN wizard. Click **Save** when you are done configuring.



VPN Client | Create New VPN Connection Entry

Connection Entry: Easy VPN

Description: Easy VPN

Host: 192.168.12.2

● Group Authentication ○ Mutual Group Authentication

Name: ciscogroup

Password: ***********

Confirm Password: ***********

○ Certificate Authentication

Name:

□ Send CA Certificate Chain

Figure 13-3: Create New VPN Connection Dialog

You should see your new profile appear in the profiles list. Before connecting, click the **Log** tab so you can enable logging before attempting to connect. Logging is not normally required but it is helpful in this lab to watch the VPN client connect.

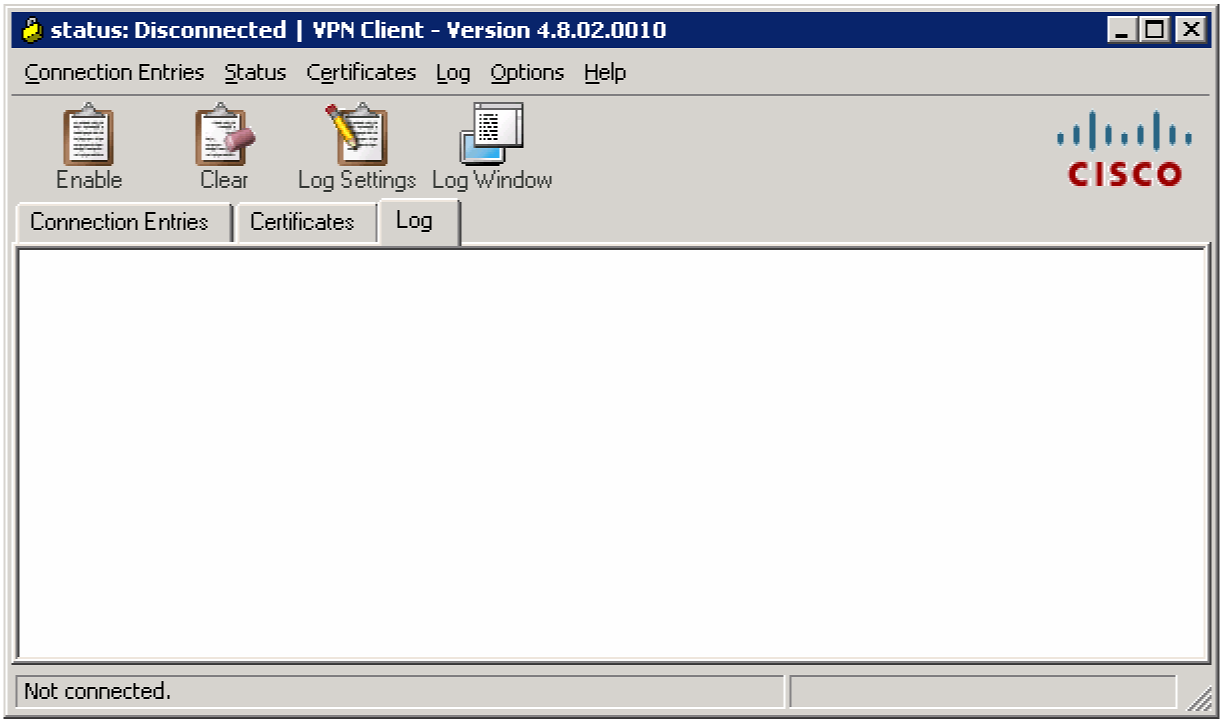


Figure 13-4: VPN Client Log Tab

Click **Log Window** to open up logging in a separate window.

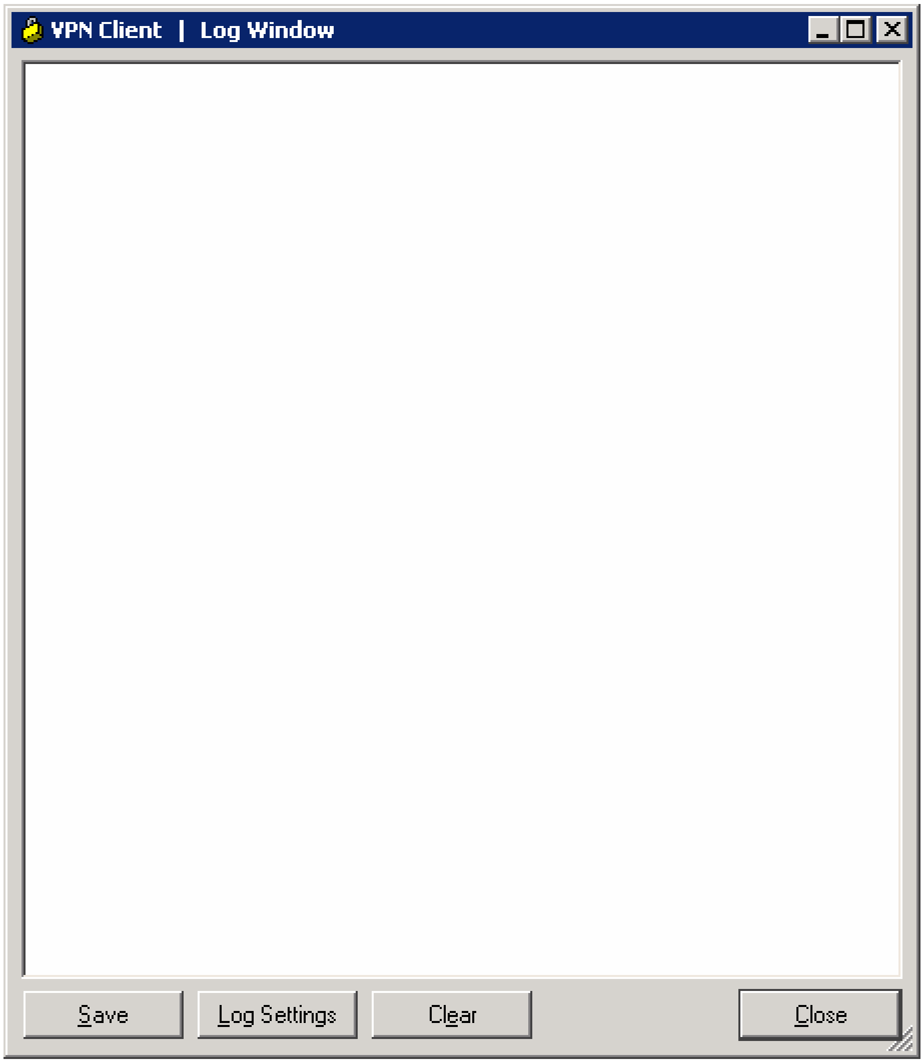
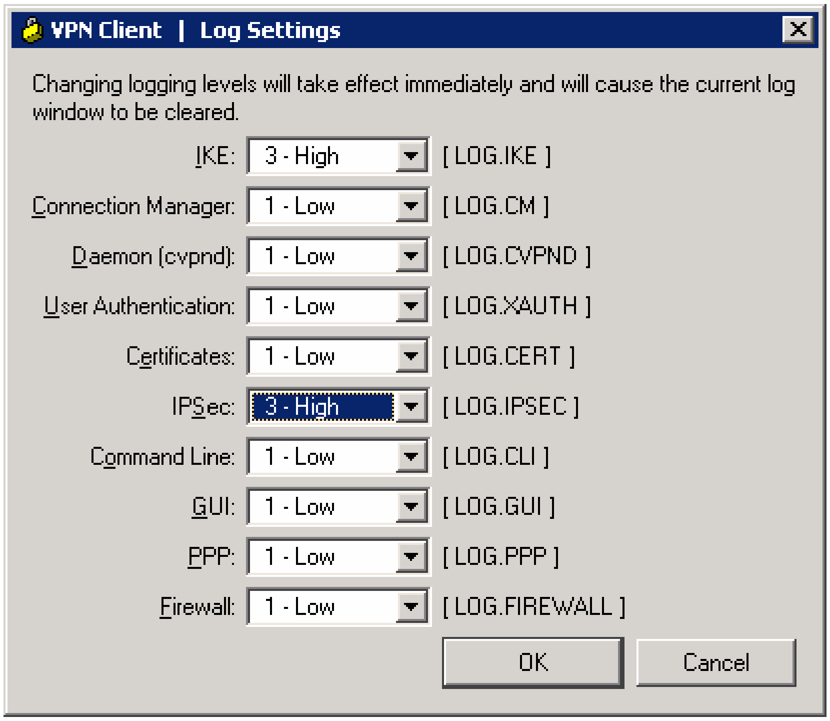


Figure 13-5: Log Window

While you have the log window open, go back to the main VPN client window and click **Log Settings**. Change the logging settings for IKE and IPsec to **3 – High**. Click **OK** to apply these settings.

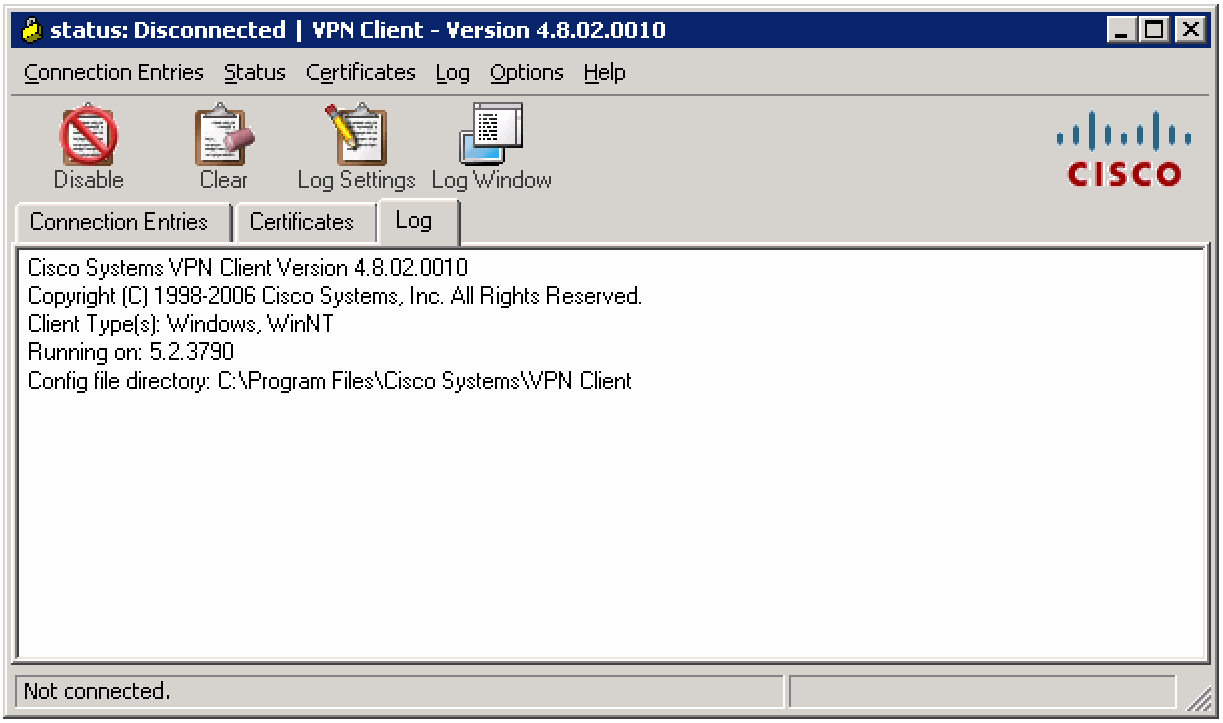


Changing logging levels will take effect immediately and will cause the current log window to be cleared.

IKE: 3 - High [LOG.IKE]
Connection Manager: 1 - Low [LOG.CM]
Daemon (cvpnd): 1 - Low [LOG.CVPND]
User Authentication: 1 - Low [LOG.XAUTH]
Certificates: 1 - Low [LOG.CERT]
IPSec: 3 - High [LOG.IPSEC]
Command Line: 1 - Low [LOG.CLI]
GUI: 1 - Low [LOG.GUI]
PPP: 1 - Low [LOG.PPP]
Firewall: 1 - Low [LOG.FIREWALL]

Figure 13-6: Logging Settings

Click **Enable** to enable logging. The **Enable** button should change to a **Disable** button.

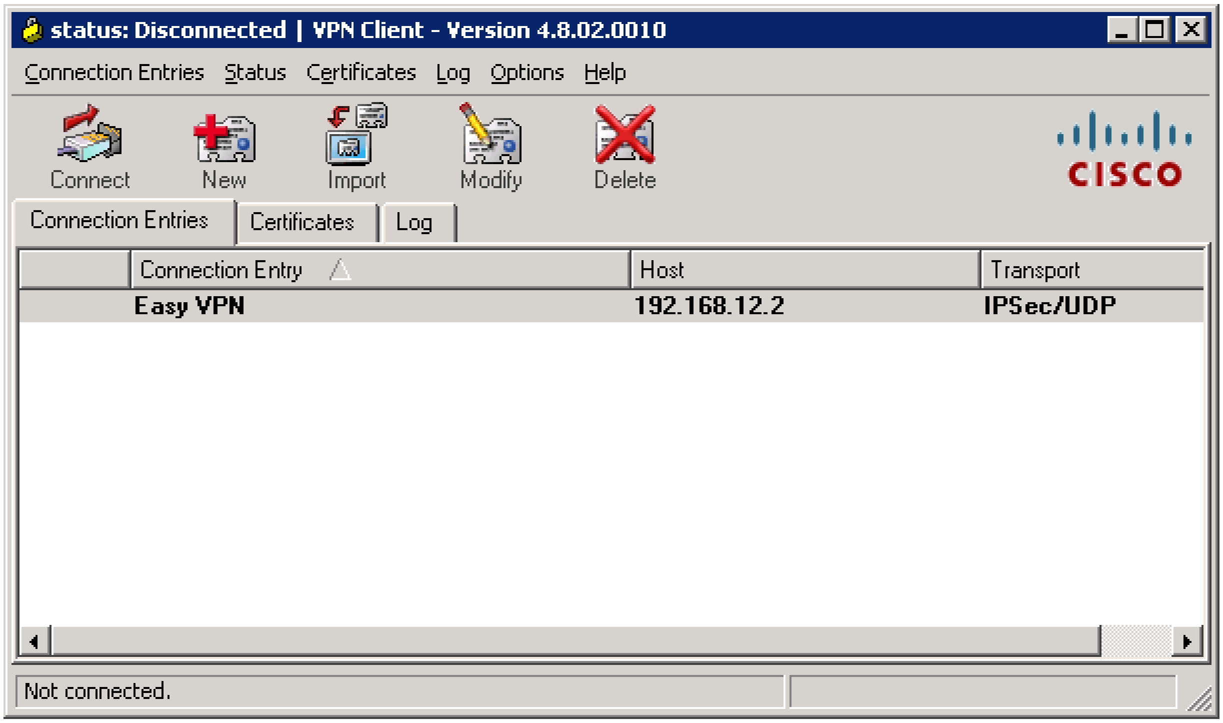


```
Cisco Systems VPN Client Version 4.8.02.0010
Copyright (C) 1998-2006 Cisco Systems, Inc. All Rights Reserved.
Client Type(s): Windows, WinNT
Running on: 5.2.3790
Config file directory: C:\Program Files\Cisco Systems\VPN Client
```

Not connected.

Figure 13-7: Log Tab with Logging Enabled

Click the **Connection Entries** tab, and double-click the entry or click **Connect** to connect to this profile.

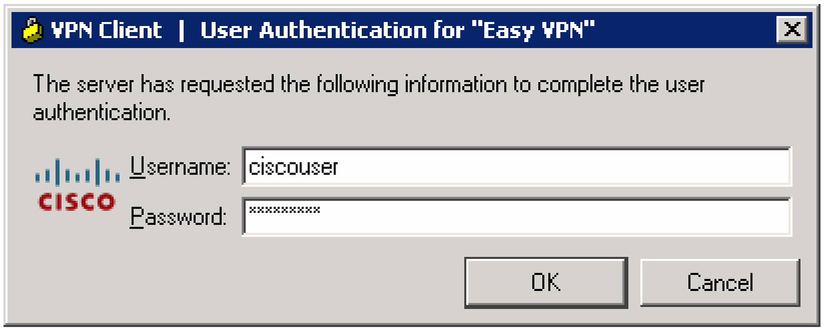

status: Disconnected | VPN Client - Version 4.8.02.0010
Connection Entries Status Certificates Log Options Help
Connect
New
Import
Modify
Delete
CISCO
Connection Entries
Certificates
Log

	Connection Entry	Host	Transport
	Easy VPN	192.168.12.2	IPSec/UDP

Not connected.

Figure 13-8: VPN Client Connections Tab

While the VPN client tries to connect to the VPN, it will prompt you for a username and password. Enter the user credentials you specified earlier during the VPN client wizard.



The server has requested the following information to complete the user authentication.

Username: ciscouser

Password: *********

Figure 13-9: User Authentication Prompt

When the VPN has successfully connected, you should see a locked padlock icon in the system tray.

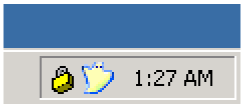
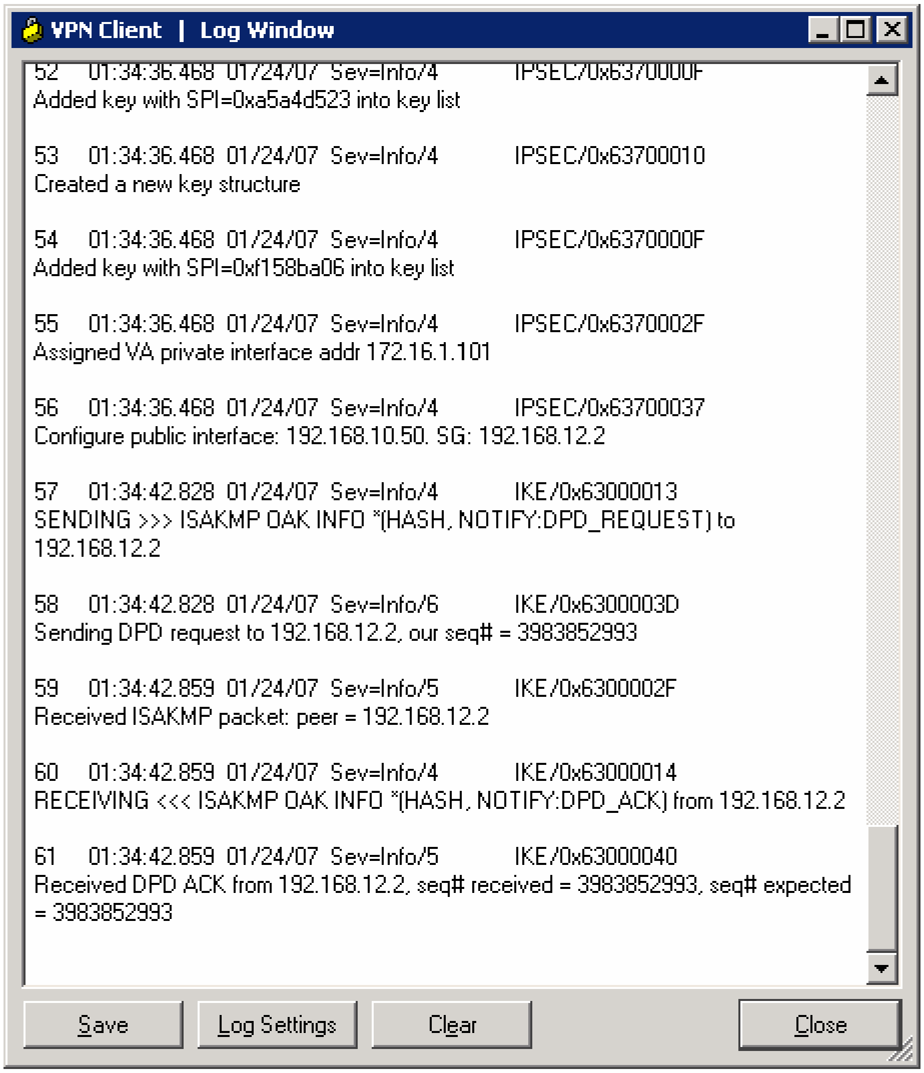


Figure 13-10: VPN Client System Tray Icon, Status: Connected

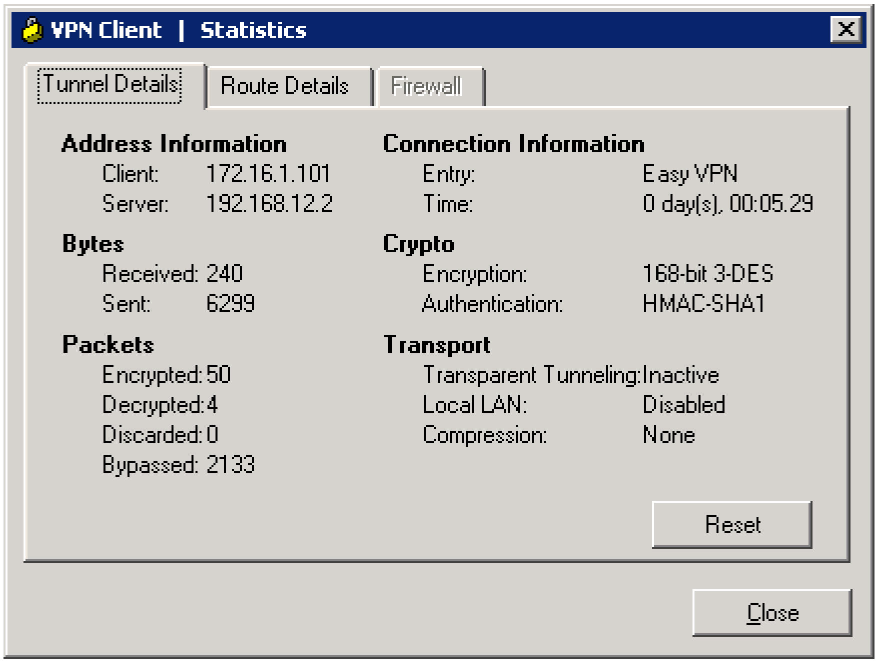
You can also see that your connection has populated the log window with information. After reviewing the information here, click **Close** to close this window. This logging functionality can be very useful when troubleshooting VPN client problems.



```
52    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x6370000F
Added key with SPI=0xa5a4d523 into key list

53    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x63700010
Created a new key structure

54    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x6370000F
Added key with SPI=0xf158ba06 into key list

55    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x6370002F
Assigned VA private interface addr 172.16.1.101

56    01:34:36.468  01/24/07  Sev=Info/4          IPSEC/0x63700037
Configure public interface: 192.168.10.50. SG: 192.168.12.2

57    01:34:42.828  01/24/07  Sev=Info/4          IKE/0x63000013
SENDING >>> ISAKMP OAK INFO *(HASH, NOTIFY:DPD_REQUEST) to
192.168.12.2

58    01:34:42.828  01/24/07  Sev=Info/6          IKE/0x6300003D
Sending DPD request to 192.168.12.2, our seq# = 3983852993

59    01:34:42.859  01/24/07  Sev=Info/5          IKE/0x6300002F
Received ISAKMP packet: peer = 192.168.12.2

60    01:34:42.859  01/24/07  Sev=Info/4          IKE/0x63000014
RECEIVING <<< ISAKMP OAK INFO *(HASH, NOTIFY:DPD_ACK) from 192.168.12.2

61    01:34:42.859  01/24/07  Sev=Info/5          IKE/0x63000040
Received DPD ACK from 192.168.12.2, seq# received = 3983852993, seq# expected
= 3983852993
```

Figure 13-11: Log Window, Populated with Connection Messages

To view VPN connection statistics, right-click the padlock icon in the system tray and click **Statistics...**.



Address Information
Client: 172.16.1.101
Server: 192.168.12.2

Bytes
Received: 240
Sent: 6299

Packets
Encrypted: 50
Decrypted: 4
Discarded: 0
Bypassed: 2133

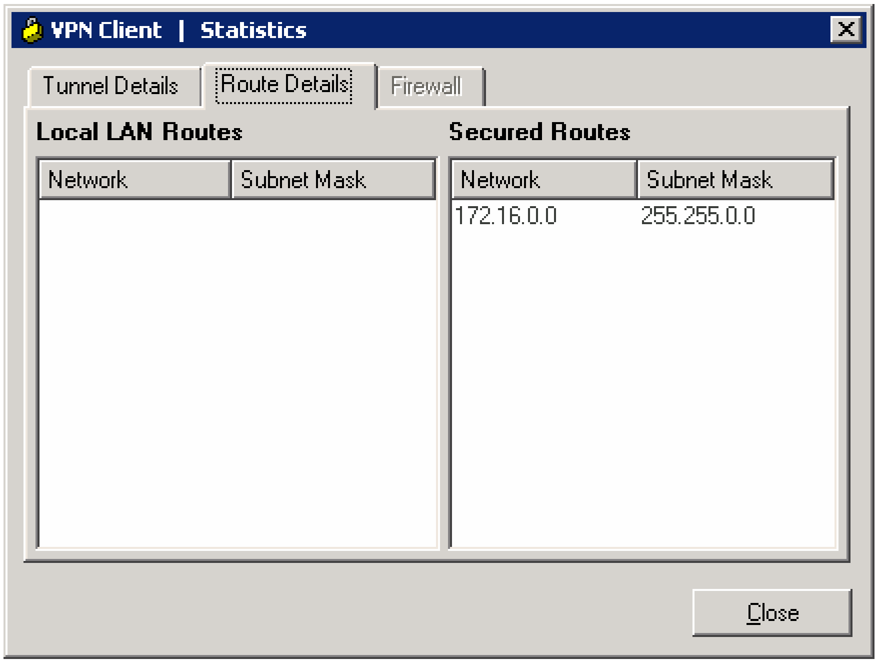
Connection Information
Entry: Easy VPN
Time: 0 day(s), 00:05.29

Crypto
Encryption: 168-bit 3-DES
Authentication: HMAC-SHA1

Transport
Transparent Tunneling: Inactive
Local LAN: Disabled
Compression: None

Figure 13-12: VPN Client Statistics

Click the **Route Details** tab to view routes sent out through split tunneling.



Local LAN Routes

Network	Subnet Mask

Secured Routes

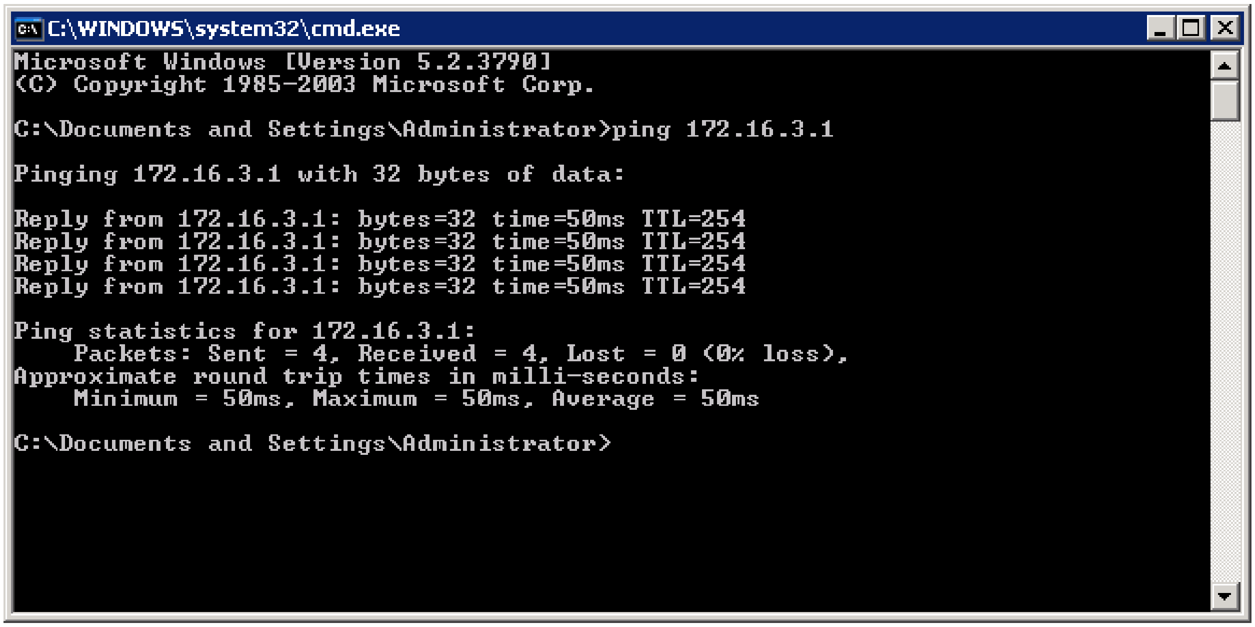
Network	Subnet Mask
172.16.0.0	255.255.0.0

Figure 13-13: Route Details Tab

Close the Statistics window when done.

Step 14: Test Inside VPN Connectivity

Now that the host has connected to the VPN, open up the command prompt again (see earlier steps if you don't remember how) and ping HQ2's loopback. This time, it should be successful.



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 172.16.3.1

Pinging 172.16.3.1 with 32 bytes of data:

Reply from 172.16.3.1: bytes=32 time=50ms TTL=254
Reply from 172.16.3.1: bytes=32 time=50ms TTL=254
Reply from 172.16.3.1: bytes=32 time=50ms TTL=254
Reply from 172.16.3.1: bytes=32 time=50ms TTL=254

Ping statistics for 172.16.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 50ms, Maximum = 50ms, Average = 50ms

C:\Documents and Settings\Administrator>
```

Figure 14-1: Successful Pings With VPN

Step 15: Verify VPN Operation using the CLI

There are many command line **show** commands that you can use to verify VPN configuration. You can use the **show crypto isakmp sa** and **show crypto ipsec sa** commands to verify crypto security associations.

```
HQ# show crypto isakmp sa
dst             src             state          conn-id slot status
192.168.12.2    192.168.10.50   QM_IDLE              1    0 ACTIVE

HQ# show crypto ipsec sa

interface: Serial0/0/0
    Crypto map tag: mymap, local addr 192.168.12.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.2.100/255.255.255.255/0/0)
   current_peer 192.168.10.50 port 1471
     PERMIT, flags={}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 44, #pkts decrypt: 44, #pkts verify: 44
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
```

```
     local crypto endpt.: 192.168.12.2, remote crypto endpt.: 192.168.10.50
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
     current outbound spi: 0xECC953E1(3972617185)

     inbound esp sas:
      spi: 0xB18FB7F1(2978985969)
        transform: esp-3des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3001, flow_id: NETGX:1, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4600939/3552)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xECC953E1(3972617185)
        transform: esp-3des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3002, flow_id: NETGX:2, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4600946/3551)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:
```

Use the command **show ip local pool** to view IP pool information.

```
HQ# show ip local pool

 Pool                     Begin           End             Free  In use
 VPNCLIENTS               172.16.2.100    172.16.2.200     100       1
```

Step 16: Disconnecting the VPN Client

Right-click the padlock icon in the system tray and click **Disconnect**. The VPN client will disconnect.

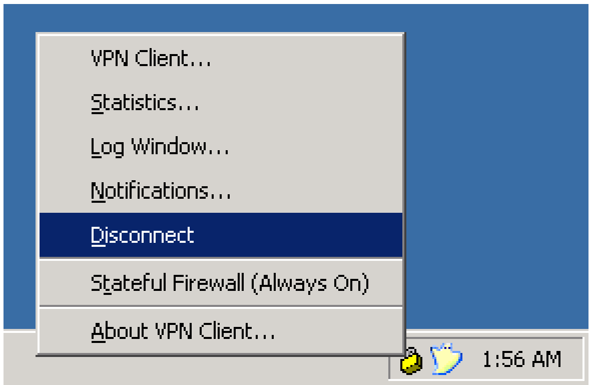


Figure 16-1: Disconnecting from the VPN via the System Tray Icon

The padlock should first change to a padlock with an ‘X’ through it, indicating that it is disconnecting. It will change to an unlocked icon, indicating no VPN connection. Finally, right-click the padlock and click **Exit** to quit the VPN client.

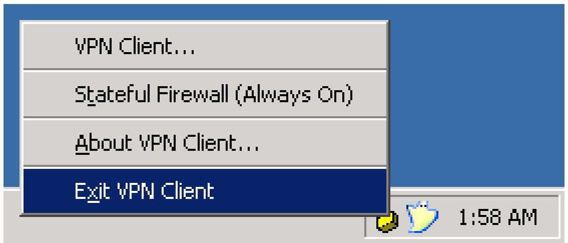


Figure 16-2: Exiting the VPN Client via the System Tray Icon

Final Configurations

```
ISP# show run
hostname ISP
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 ip address 192.168.12.1 255.255.255.0
 clock rate 64000
 no shutdown
end

HQ# show run
hostname HQ
!
aaa new-model
!
aaa authentication login default local none
aaa authentication login VPNAUTH local
aaa authorization network VPNAUTH local
!
username cisco password 0 cisco
username ciscouser password 0 ciscouser
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 2
crypto isakmp keepalive 30 5
crypto isakmp xauth timeout 60
!
crypto isakmp client configuration group ciscogroup
 key ciscogroup
 pool VPNCLIENTS
 acl 100
 netmask 255.255.255.0
!
crypto ipsec transform-set mytrans esp-3des esp-sha-hmac
!
crypto dynamic-map mymap 10
 set transform-set mytrans
 reverse-route
!
```

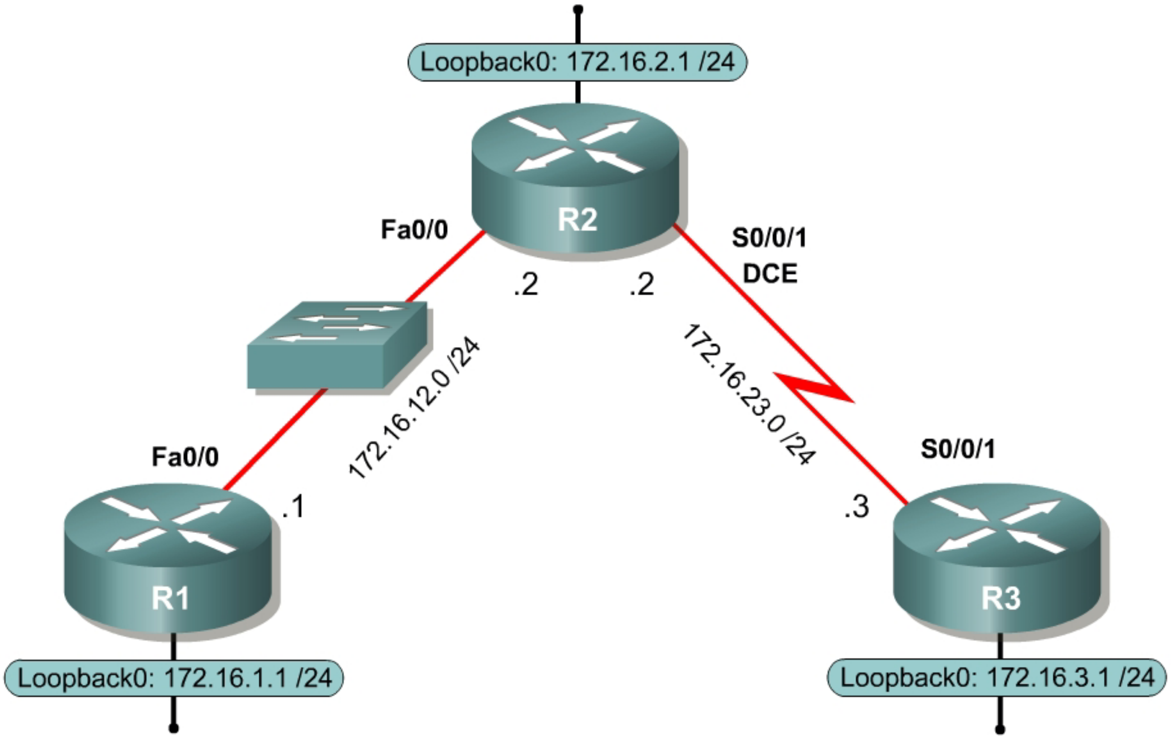
```
crypto map mymap client authentication list VPNAUTH
crypto map mymap isakmp authorization list VPNAUTH
crypto map mymap client configuration address respond
crypto map mymap 10 ipsec-isakmp dynamic mymap
!
interface Loopback0
 ip address 172.16.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.12.2 255.255.255.0
 crypto map mymap
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.23.2 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 redistribute static
 network 172.16.0.0
 no auto-summary
!
ip local pool VPNCLIENTS 172.16.2.100 172.16.2.200
ip route 0.0.0.0 0.0.0.0 192.168.12.1
!
access-list 100 permit ip 172.16.0.0 0.0.255.255 any
end

HQ2# show run
hostname HQ2
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.16.23.3 255.255.255.0
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
end
```

Lab 4.1 Configuring Frame Mode MPLS

Learning Objectives

- Configure EIGRP on a router
- Configure Label Distribution Protocol on a router
- Change the size of the Maximum Transmission Unit (MTU)
- Verify MPLS behavior

Topology Diagram


Loopback0: 172.16.2.1 /24
R2
Fa0/0
.2
.2
S0/0/1
DCE
172.16.12.0 /24
172.16.23.0 /24
Fa0/0
.1
S0/0/1
.3
R1
R3
Loopback0: 172.16.1.1 /24
Loopback0: 172.16.3.1 /24

Scenario

In this lab, you will configure a simple Enhanced Interior Gateway Routing Protocol (EIGRP) network to route IP packets. You will run Multiprotocol Label Switching (MPLS) over the IP internetwork to fast-switch Layer 2 frames.

Step 1: Configure Addressing

Configure the loopback interfaces with the addresses shown in the topology diagram. Also configure the serial interfaces shown in the diagram. Set the clock rate on the appropriate interface and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command.

```
R1(config)# interface loopback 0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface fastethernet 0/0
R1(config-if)# ip address 172.16.12.1 255.255.255.0
R1(config-if)# no shutdown

R2(config)# interface loopback 0
R2(config-if)# ip address 172.16.2.1 255.255.255.0
R2(config-if)# interface fastethernet 0/0
R2(config-if)# ip address 172.16.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip address 172.16.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3(config)# interface loopback 0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip address 172.16.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Configure EIGRP AS 1

Configure EIGRP for AS1 on all three routers. Add the whole major network 172.16.0.0 and disable automatic summarization.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.0.0

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 172.16.0.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
R3(config-router)# network 172.16.0.0
```

EIGRP neighbor adjacencies should form between R1 and R2 and between R2 and R3. If the adjacencies do not form, troubleshoot by checking your interface configuration, EIGRP configuration, and physical connectivity.

What impact does IP connectivity have on MPLS?

Step 3: Observe CEF Operation

Since all the routers have EIGRP adjacencies and are advertising the entire major 172.16.0.0 network, all routers should have full routing tables.

```
R1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
D       172.16.23.0 [90/2172416] via 172.16.12.2, 00:01:56, FastEthernet0/0
C       172.16.12.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, Loopback0
D       172.16.2.0 [90/156160] via 172.16.12.2, 00:01:56, FastEthernet0/0
D       172.16.3.0 [90/2300416] via 172.16.12.2, 00:01:51, FastEthernet0/0
```

On R1, if you perform a **traceroute** to the R3s loopback, you see the path the packet follows. This output changes slightly once we configure MPLS.

```
R1# traceroute 172.16.3.1

Type escape sequence to abort.
Tracing the route to 172.16.3.1

  1 172.16.12.2 0 msec 0 msec 0 msec
  2 172.16.23.3 16 msec 12 msec *
```

Cisco Express Forwarding (CEF) is Cisco’s proprietary Layer 3 switching algorithm for Cisco IOS routers. CEF allows forwarding to be distributed throughout the line cards on Cisco models like the Catalyst 6500. CEF also provides quicker switching than switching based on the routing table (process switching) or switching based on a standards-compliant forwarding information base (fast switching).

What is the function of CEF?

Which information does CEF view as significant in making a forwarding determination for an IP packet?

You can also see that CEF is enabled by default by using the **show ip cef command**.

```
R1# show ip cef
Prefix              Next Hop             Interface
0.0.0.0/0           drop                 Null0 (default route handler entry)
0.0.0.0/32          receive
172.16.1.0/24       attached             Loopback0
172.16.1.0/32       receive
172.16.1.1/32       receive
172.16.1.255/32     receive
172.16.2.0/24       172.16.12.2          FastEthernet0/0
172.16.3.0/24       172.16.12.2          FastEthernet0/0
172.16.12.0/24      attached             FastEthernet0/0
172.16.12.0/32      receive
172.16.12.1/32      receive
172.16.12.2/32      172.16.12.2          FastEthernet0/0
172.16.12.255/32    receive
172.16.23.0/24      172.16.12.2          FastEthernet0/0
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive
```

Another important CEF command is the **show ip cef non-recursive** command which allows the user to display CEF forwarding information for prefixes installed in the routing table.

```
R1# show ip cef non-recursive
Prefix              Next Hop             Interface
172.16.1.0/24       attached             Loopback0
172.16.2.0/24       172.16.12.2          FastEthernet0/0
172.16.3.0/24       172.16.12.2          FastEthernet0/0
172.16.12.0/24      attached             FastEthernet0/0
172.16.12.2/32      172.16.12.2          FastEthernet0/0
172.16.23.0/24      172.16.12.2          FastEthernet0/0
```

CEF records both the Layer 3 next-hop information and the Layer 2 frame next-hop information. CEF currently supports the following Layer 2 protocols: ATM, Frame Relay, Ethernet, Fiber Distributed Data Interface (FDDI), PPP, High-Level Datalink Control (HDLC), and tunnels.

CEF is critical to the operation of MPLS on Cisco routers because MPLS packets must be forwarded based on label. Since the CEF architecture can support multiple protocols such as IPv4, IPv6, CEF switching could naturally be extended to support MPLS labels as well.

CEF should be enabled by default. If CEF is not enabled, issue the **ip cef** command in global configuration mode on each router.

Step 4: Enable MPLS on All Physical Interfaces

MPLS is a standardized protocol that allows routers to switch packets based on labels, rather than route switch packets based on standards in the protocol's routing formula. Under normal IP routing, every intermediate system looks up the destination prefix of an IP packet in the Routing Information Base (RIB) of a router or in the Forwarding Information Base (FIB) of a fast switch at every Layer 3 node. Instead of switching that is based on prefix, the first router running MPLS can encapsulate the IP packet in an MPLS frame and then further encapsulate the packet in the Layer 2 frame before sending it across one of many supported Layer 2 media. At the next MPLS-enabled Label Switch Router (LSR), the MPLS frame is read and the IP packet is switched as an MPLS frame from router to router with little rewrite at each node.

This allows routers to switch multiple protocols (hence the name) using the same switching mechanism, as well as perform some other functionality not available in traditional destination-based forwarding, including Layer 2 VPNs (AToM), Layer 3 VPNs, and traffic engineering. MPLS runs between Layers 2 and 3 of the OSI model and, because of this, is sometimes said to run at Layer 2½. The MPLS header is 4 bytes long and includes a 20-bit label.

Configuring the interface-level command **mpls ip** on an interface tells the router to switch MPLS packets inbound and outbound on that interface as well as attempt to bring up MPLS adjacencies with the Label Distribution Protocol (LDP) out that egress interface. LDP facilitates communication between MPLS peers by allowing them to inform each other of labels to assign packets to particular destinations based on Layer 2, Layer 3, or other significant information.

Configure MPLS on all physical interfaces in the topology.

NOTE: If you are running the 12.4 version of the IOS on your routers, then the **mpls ip** command is what you will use in this lab. However, when Cisco first developed packet-labeling technology, it was called tag switching. Therefore, if you are running an older version of the IOS, then you may see one of two different variations. The first variation is that your router will accept the **mpls ip** command. However, the commands will be stored in IOS as **tag-switching** commands. The second variation is that your router will not accept the **mpls ip** command. In this event, the **mpls ip** command may be entered as the **tag-switching ip** command. Try the newer commands first, beginning with the **mpls** keyword.

```
R1(config)# interface fastethernet0/0
R1(config-if)# mpls ip

R2(config)# interface fastethernet0/0
R2(config-if)# mpls ip
*Jan 31 08:28:54.315: %LDP-5-NBRCHG: LDP Neighbor 172.16.1.1:0 (1) is UP
R2(config-if)# interface serial0/0/1
R2(config-if)# mpls ip

R3(config)# interface serial0/0/1
R3(config-if)# mpls ip
*Jan 31 08:32:11.571: %LDP-5-NBRCHG: LDP Neighbor 172.16.2.1:0 (1) is UP
```

Notice that as you configure MPLS on both ends of a connection, IOS logs a messages to the console on both routers indicating that an LDP neighbor adjacency has formed.

Although you are going to use LDP in this lab, there is another Cisco-proprietary label exchanging protocol called Tag Distribution Protocol (TDP) which was part of the Cisco Tag Switching architecture. To change the protocol being used, use the **mpls label protocol** *protocol* command either on a global level at the global configuration prompt or on a per-interface basis, using the interface-level version of this command. Cisco TDP and MPLS LDP are nearly identical in function, but use incompatible message formats and some different procedures. Cisco is changing from TDP to a fully compliant LDP.

Step 5: Verify MPLS Configuration

MPLS has many **show** commands that you can use to verify proper MPLS operation. Issue the **show mpls interfaces** command to see a quick summary of interfaces configured with MPLS. Keep in mind that you will see this output because you applied the **mpls ip** command to these interfaces.

```
R1# show mpls interfaces
Interface              IP            Tunnel   Operational
FastEthernet0/0        Yes (ldp)     No       Yes


R2# show mpls interfaces
Interface              IP            Tunnel   Operational
FastEthernet0/0        Yes (ldp)     No       Yes
Serial0/0/1            Yes (ldp)     No       Yes

R3# show mpls interfaces
Interface              IP            Tunnel   Operational
Serial0/0/1            Yes (ldp)     No       Yes
```

Issue the **show mpls ldp discovery** command to find out local sources for LDP exchanges and the **show mpls ldp neighbor** command to show LDP adjacencies. Notice that MPLS chooses its IDs based on loopback interfaces, similar to other protocols such asOpen Shortest Path First (OSPF), Border Gateway Protocol (BGP).

```
R1# show mpls ldp discovery
 Local LDP Identifier:
    172.16.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/0 (ldp): xmit/recv
            LDP Id: 172.16.2.1:0; no host route

R1# show mpls ldp neighbor
    Peer LDP Ident: 172.16.2.1:0; Local LDP Ident 172.16.1.1:0
        TCP connection: 172.16.2.1.49525 - 172.16.1.1.646
        State: Oper; Msgs sent/rcvd: 29/26; Downstream
        Up time: 00:16:40
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 172.16.12.2
        Addresses bound to peer LDP Ident:
          172.16.12.2     172.16.23.2     172.16.2.1


R2# show mpls ldp discovery
```

```
 Local LDP Identifier:
    172.16.2.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/0 (ldp): xmit/recv
            LDP Id: 172.16.1.1:0; no host route
        Serial0/0/1 (ldp): xmit/recv
            LDP Id: 172.16.3.1:0; no host route

R2# show mpls ldp neighbor
    Peer LDP Ident: 172.16.1.1:0; Local LDP Ident 172.16.2.1:0
        TCP connection: 172.16.1.1.646 - 172.16.2.1.49525
        State: Oper; Msgs sent/rcvd: 27/30; Downstream
        Up time: 00:17:06
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 172.16.12.1
        Addresses bound to peer LDP Ident:
          172.16.12.1     172.16.1.1
    Peer LDP Ident: 172.16.3.1:0; Local LDP Ident 172.16.2.1:0
        TCP connection: 172.16.3.1.34352 - 172.16.2.1.646
        State: Oper; Msgs sent/rcvd: 27/26; Downstream
        Up time: 00:16:23
        LDP discovery sources:
          Serial0/0/1, Src IP addr: 172.16.23.3
        Addresses bound to peer LDP Ident:
          172.16.23.3     172.16.3.1


R3# show mpls ldp discovery
 Local LDP Identifier:
    172.16.3.1:0
    Discovery Sources:
    Interfaces:
        Serial0/0/1 (ldp): xmit/recv
            LDP Id: 172.16.2.1:0; no host route
R3# show mpls ldp neighbor
    Peer LDP Ident: 172.16.2.1:0; Local LDP Ident 172.16.3.1:0
        TCP connection: 172.16.2.1.646 - 172.16.3.1.34352
        State: Oper; Msgs sent/rcvd: 27/28; Downstream
        Up time: 00:17:19
        LDP discovery sources:
          Serial0/0/1, Src IP addr: 172.16.23.2
        Addresses bound to peer LDP Ident:
          172.16.12.2     172.16.23.2     172.16.2.1
```

What interface does LDP use on R1 to identify itself to other LDP peers?

What transport protocol does LDP use to communicate with other LDP peers?

In the configuration you set up in Step 4, all routers are acting as Label Switch Routers (LSRs) and running LDP. On LSRs, each forwarding equivalence class (in this case, each routable IP prefix) is assigned an MPLS label. LDP automatically distributes labels to peers to be used when sending traffic to specific destinations through the LSR. Once labels have been distributed, switching for MPLS packets is done through the Label Information Base (LIB).

Display the contents of the LIB using the **show mpls ldp bindings** command. There is a binding for every routed prefix; however, the bindings may vary from router to router since they can get swapped at each hop. In a larger network, the way labels are swapped is easier to see. The LIB is also referred to on Cisco routers as the TIB, a legacy name from Tag Switching. Do not be alarmed to see the LIB entries listed instead as TIB entries: this does not signal that TDP is the protocol being used for distribution.

```
R1# show mpls ldp bindings
  tib entry: 172.16.1.0/24, rev 6
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.2.1:0, tag: 16
  tib entry: 172.16.2.0/24, rev 8
        local binding:  tag: 17
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.3.0/24, rev 10
        local binding:  tag: 18
        remote binding: tsr: 172.16.2.1:0, tag: 17
  tib entry: 172.16.12.0/24, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.23.0/24, rev 2
        local binding:  tag: 16
        remote binding: tsr: 172.16.2.1:0, tag: imp-null

R2# show mpls ldp bindings
  tib entry: 172.16.1.0/24, rev 6
        local binding:  tag: 16
        remote binding: tsr: 172.16.1.1:0, tag: imp-null
        remote binding: tsr: 172.16.3.1:0, tag: 17
  tib entry: 172.16.2.0/24, rev 8
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.1.1:0, tag: 17
        remote binding: tsr: 172.16.3.1:0, tag: 18
  tib entry: 172.16.3.0/24, rev 10
        local binding:  tag: 17
        remote binding: tsr: 172.16.1.1:0, tag: 18
        remote binding: tsr: 172.16.3.1:0, tag: imp-null
  tib entry: 172.16.12.0/24, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.1.1:0, tag: imp-null
        remote binding: tsr: 172.16.3.1:0, tag: 16
  tib entry: 172.16.23.0/24, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.1.1:0, tag: 16
        remote binding: tsr: 172.16.3.1:0, tag: imp-null

R3# show mpls ldp bindings
  tib entry: 172.16.1.0/24, rev 6
        local binding:  tag: 17
        remote binding: tsr: 172.16.2.1:0, tag: 16
  tib entry: 172.16.2.0/24, rev 8
        local binding:  tag: 18
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.3.0/24, rev 10
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.2.1:0, tag: 17
  tib entry: 172.16.12.0/24, rev 4
        local binding:  tag: 16
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.23.0/24, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
```

The local bindings are generated by LDP on a Label Switch Router when LDP is enabled. A label is generated for every prefix in the routing table. These labels are then sent to all of the router's LDP peers. A tag of implicit-NULL ("imp-null" in the output of the command **show mpls ldp bindings**) is advertised when the packet with not be forwarded locally based on label, but based on prefix. This situation regularly occurs with connected networks.

For instance, assume R2 and R3 have already peered with each other using LDP. Now R1 begins running MPLS and attempts to peer to R2:

1. R1 generates the locally bound label, namely 18, for the prefix 172.16.3.0/24 in its routing table.

2. R1 advertises the local binding to its LDP peer, R2.

3. R2 enters R1's binding for the 172.16.3.0/24 prefix, now classified as a remote binding, into its LIB, regardless of whether it uses it to reach the destination network. The remote binding for this IP prefix through R1 is label 18.

4. Based on the routing table, R2 will use R3 as the next hop for the 172.16.3.0/24. R2 will not forward IP packets inside an MPLS encapsulation, but rather simply as IP packets because R3 has advertised the label of implicit-NULL to R2.

What is the significance of the "local binding" entry?

What is the significance of a "remote binding" entry?

On R2, why is there more than one remote binding for each of the networks in the diagram?

Note that LDP assigns local labels to *all* Interior Gateway Protocol (IGP) prefixes and advertises the bindings to *all* LDP peers. The concept of split horizon does not exist; an LDP peer assigns its own local label to a prefix and advertises that back to the other LDP peer, even though that other LDP peer owns the prefix (it is a connected prefix) or that other LDP peer is the downstream LSR.

What is the meaning of the implicit NULL label?

As mentioned earlier, **traceroute** would differ slightly once MPLS was set up. The output now includes labels for each hop. Unfortunately, because of the size of this network, you only see one label. In a larger network, you would see more hops, and therefore more labels.

```
R1# traceroute 172.16.3.1

Type escape sequence to abort.
Tracing the route to 172.16.3.1

  1 172.16.12.2 [MPLS: Label 17 Exp 0] 44 msec 44 msec 48 msec
  2 172.16.23.3 12 msec 12 msec *
```

Step 6: Change MPLS MTU

Because you are adding in extra header information to packets, the MTU of packets can change. Remember that each MPLS header is 4 bytes. The default MTU size of MPLS packets is taken from the interface it is running on, which in the case of Ethernet is 1500 bytes. To verify this, use the **show mpls interfaces** *interface-type interface-number* **detail** command to the Ethernet connections of R1 and R2.

```
R1# show mpls interfaces fastethernet 0/0 detail
Interface FastEthernet0/0:
        IP labeling enabled (ldp):
          Interface config
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1500

R2# show mpls interfaces fastethernet 0/0 detail
Interface FastEthernet0/0:
        IP labeling enabled (ldp):
          Interface config
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1500
```

For this lab, we will change the Ethernet connection between R1 and R2 to support 2 MPLS headers, so we will change the MPLS MTU to 1508 on their Fast Ethernet interfaces. To change the MPLS MTU, use the **mpls mtu** *size* command in interface configuration mode. Verify the change using the **show mpls interfaces** *interface* **detail** command used earlier.

```
R1(config)# interface fastethernet 0/0
R1(config-if)# mpls mtu 1508

R2(config)# interface fastethernet0/0
R2(config-if)# mpls mtu 1508

R1# show mpls interface fastethernet 0/0 detail
Interface FastEthernet0/0:
        IP labeling enabled (ldp):
          Interface config
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1508

R2# show mpls interface fastethernet 0/0 detail
Interface FastEthernet0/0:
        IP labeling enabled (ldp):
          Interface config
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1508
```

Final Configurations

```
R1# show run
!
hostname R1
!
```

```
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 172.16.12.1 255.255.255.0
 mpls ip
 mpls mtu 1508
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
!
end
```

```
R2# show run
!
hostname R2
!
interface Loopback0
 ip address 172.16.2.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 172.16.12.2 255.255.255.0
 mpls ip
 mpls mtu 1508
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.23.2 255.255.255.0
 mpls ip
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
!
end
```

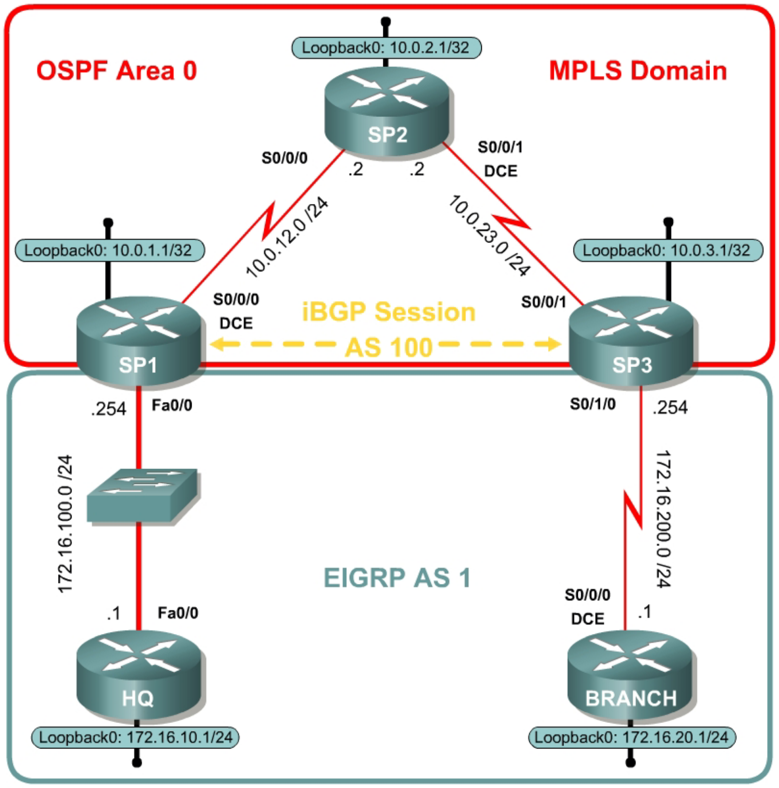
```
R3# show run
!
hostname R3
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.16.23.3 255.255.255.0
 mpls ip
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
!
end
```

Lab 4.2 Challenge Lab: Implementing MPLS VPNs

Learning Objectives

- Configure Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP) on a router
- Enable MPLS on a router
- Verify MPLS implementation
- Configure a Virtual Routing/Forwarding (VRF) instance
- Use MBGP to exchange VPN routing updates
- Verify VPN activity

Topology Diagram



Scenario

As a network engineer at a service provider corporation, you suggest rolling out MPLS as a new transport technology to facilitate VPNs between customer sites that connect through your network. Your CIO has asked you to implement proof-of-concept in a lab environment, starting with a small implementation of MPLS VPNs before moving up to more moderately sized test cases.

MPLS VPN technology is a powerful technology that leverages the multiprotocol aspect of MPLS to switch MPLS frames between VPN endpoints while hiding the customer networks from the MPLS transport network that connects them. In other words, the intermediate transport network has no knowledge of the customer's IP networks, but is still able to label-switch frames based on information it receives from MPLS Label Distribution Protocol (LDP) relationships.

You decide to model one of your current customer's connections and then show how MPLS VPNs can be used to carry customer traffic through the provider network. The International Travel Agency currently uses your network to connect from its corporate headquarters to a remote branch office, so you choose this customer network to model in your demonstration.

First, set up the model of both the service provider's network and the agency's network. Then use appropriate routing and forwarding techniques to set up an MPLS VPN between the provider edge routers to which the customer connects.

SP1, SP2, and SP3 will represent a service provider network, and HQ and BRANCH will represent the International Travel Agency routers at their headquarters and at a branch site.

Step 1: Configure Addressing

Configure the loopback interfaces with the addresses shown in the diagram. Also configure the serial interfaces shown in the diagram. Set the clock rate on the appropriate interfaces and issue the **no shutdown** command on all physical interfaces. Verify that you have connectivity across the local subnet using the **ping** command inside the service provider domain. Wait to configure the interface on SP1 facing HQ and the interface on SP3 facing BRANCH. These will be configured later.

```
SP1(config)# interface loopback 0
SP1(config-if)# ip address 10.0.1.1 255.255.255.255
SP1(config-if)# interface serial 0/0/0
SP1(config-if)# ip address 10.0.12.1 255.255.255.0
SP1(config-if)# no shutdown
SP1(config-if)# interface fastethernet 0/0
SP1(config-if)# ip address 172.16.100.254 255.255.255.0
SP1(config-if)# no shutdown

SP2(config)# interface loopback 0
SP2(config-if)# ip address 10.0.2.1 255.255.255.255
SP2(config-if)# interface serial 0/0/0
SP2(config-if)# ip address 10.0.12.2 255.255.255.0
SP2(config-if)# no shutdown
SP2(config-if)# interface serial 0/0/1
SP2(config-if)# ip address 10.0.23.2 255.255.255.0
SP2(config-if)# clockrate 64000
SP2(config-if)# no shutdown

SP3(config)# interface loopback 0
SP3(config-if)# ip address 10.0.3.1 255.255.255.255
```

```
SP3(config-if)# interface serial 0/0/1
SP3(config-if)# ip address 10.0.23.3 255.255.255.0
SP3(config-if)# no shutdown
SP3(config-if)# interface serial 0/1/0
SP3(config-if)# ip address 172.16.200.254 255.255.255.0
SP3(config-if)# no shutdown
```

Configure customer sites HQ and BRANCH.

```
HQ(config)# interface loopback 0
HQ(config-if)# ip address 172.16.10.1 255.255.255.0
HQ(config-if)# interface fastethernet 0/0
HQ(config-if)# ip address 172.16.100.1 255.255.255.0
HQ(config-if)# no shutdown

BRANCH(config)# interface loopback 0
BRANCH(config-if)# ip address 172.16.20.1 255.255.255.0
BRANCH(config-if)# interface serial 0/0/0
BRANCH(config-if)# ip address 172.16.200.1 255.255.255.0
BRANCH(config-if)# clockrate 64000
BRANCH(config-if)# no shutdown
```

Step 2: Configure Routing in the Service Provider Domain

Your service provider network uses OSPF as its routing protocol, advertising internal loopback interfaces and transit networks. Configure OSPF to model the service provider domain. Add all of the interfaces addressed within the 10.0.0.0 major network into Area 0 of the OSPF process. You will only need to configure OSPF in this manner on the service provider routers, namely SP1, SP2, and SP3.

```
SP1(config)# router ospf 1
SP1(config-router)# network 10.0.0.0 0.255.255.255 area 0

SP2(config)# router ospf 1
SP2(config-router)# network 10.0.0.0 0.255.255.255 area 0

SP3(config)# router ospf 1
SP3(config-router)# network 10.0.0.0 0.255.255.255 area 0
```

Verify that all of your OSPF adjacencies come up. OSPF adjacencies should form between SP1 and SP2 and between SP2 and SP3. If the adjacencies do not form, troubleshoot by checking your interface configuration, OSPF configuration, and physical connectivity.

What purpose does OSPF serve in the configuration above?

Consider that you will deploy Border Gateway Protocol (BGP) in the SP domain later in the lab using loopback addresses as the sources for BGP updates. Why do you need to deploy an Interior Gateway Protocol (IGP) in the SP domain?

Step 3: Configure MPLS in the SP Domain

On all the service provider routers, force MPLS to use the Loopback 0 interface as the router ID for Label Distribution Protocol (LDP) adjacencies. The loopback interface would be chosen by each router automatically, but it is advisable to force the ID so that the value is persistent through topology changes and reloads. To force LDP's selection of the loopback interface as the router ID, use the **mpls ldp router-id** *interface* **force** command in global configuration mode. Also, enable MPLS on all of the physical interfaces in the MPLS domain with the **mpls ip** command.

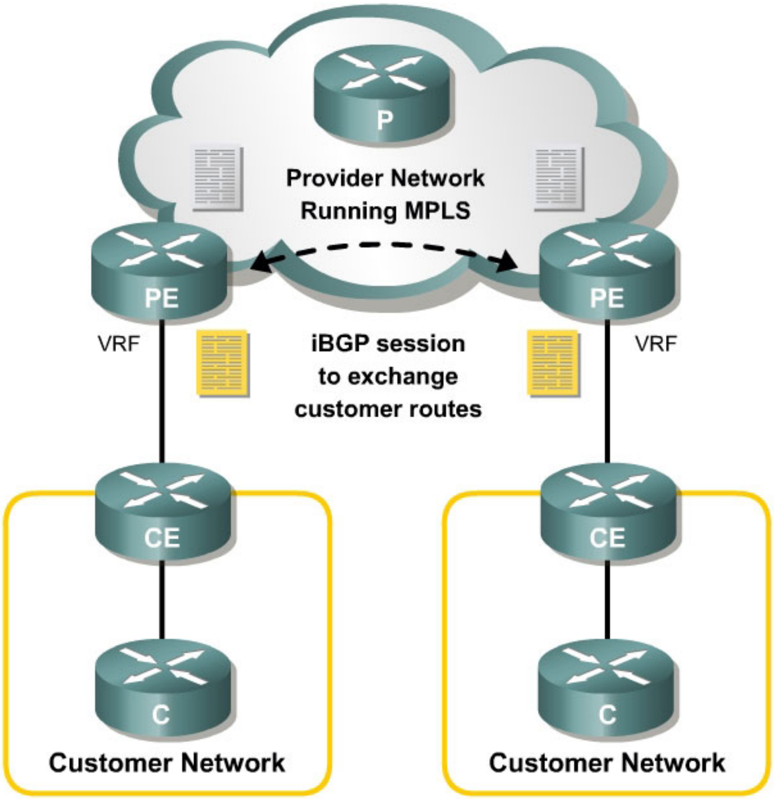
```
SP1(config)# mpls ldp router-id loopback0 force
SP1(config)# interface serial0/0/0
SP1(config-if)# mpls ip

SP2(config)# mpls ldp router-id loopback0 force
SP2(config)# interface serial0/0/0
SP2(config-if)# mpls ip
SP2(config-if)# interface serial0/0/1
SP2(config-if)# mpls ip

SP3(config)# mpls ldp router-id loopback0 force
SP3(config)# interface serial0/0/1
SP3(config-if)# mpls ip
```

You should see console messages notifying you that the MPLS-enabled routers have become adjacent with each other via LDP. Verify that these adjacencies have formed using the **show mpls ldp neighbor** command.

```
SP1# show mpls ldp neighbor
    Peer LDP Ident: 10.0.2.1:0; Local LDP Ident 10.0.1.1:0
        TCP connection: 10.0.2.1.62676 - 10.0.1.1.646
        State: Oper; Msgs sent/rcvd: 9/9; Downstream
        Up time: 00:01:43
        LDP discovery sources:
          Serial0/0/0, Src IP addr: 10.0.12.2
        Addresses bound to peer LDP Ident:
          10.0.12.2       10.0.23.2       10.0.2.1

SP2# show mpls ldp neighbor
    Peer LDP Ident: 10.0.1.1:0; Local LDP Ident 10.0.2.1:0
        TCP connection: 10.0.1.1.646 - 10.0.2.1.62676
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:02:03
        LDP discovery sources:
          Serial0/0/0, Src IP addr: 10.0.12.1
        Addresses bound to peer LDP Ident:
          10.0.12.1       10.0.1.1
    Peer LDP Ident: 10.0.3.1:0; Local LDP Ident 10.0.2.1:0
        TCP connection: 10.0.3.1.42919 - 10.0.2.1.646
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:01:58
        LDP discovery sources:
          Serial0/0/1, Src IP addr: 10.0.23.3
        Addresses bound to peer LDP Ident:
          10.0.23.3       10.0.3.1

SP3# show mpls ldp neighbor
    Peer LDP Ident: 10.0.2.1:0; Local LDP Ident 10.0.3.1:0
        TCP connection: 10.0.2.1.646 - 10.0.3.1.42919
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:02:08
        LDP discovery sources:
          Serial0/0/1, Src IP addr: 10.0.23.2
        Addresses bound to peer LDP Ident:
          10.0.12.2       10.0.23.2       10.0.2.1
```

Step 4: Configure a VRF

An MPLS VPN is a Layer 3 VPN that allows the routing of packets through a MPLS core. This type of VPN provides a customer with connections to multiple sites through a service provider’s network. The service provider not only provides the physical connection, but the ability to dynamically route between the VPN endpoints. This is especially impressive when one considers that the customers may not be using globally unique Layer 3 addresses. For instance, different customers can use private addresses, as defined by RFC 1918, but still use the same transit provider to route their specific endpoints without translation. The routers at the provider’s edge run the same routing protocol as the customer’s network and allow the customer offices to interface with the provider.



The standard model for MPLS VPNs uses the following designations:

- Provider (P) — Routers owned by the SP that act as label switch routers (LSRs) to provide transit across the provider backbone. P routers do not carry customer routes in their routing tables.

- Customer (C) — Routers owned by the customer that provide transit through the normal customer network.

- Customer Edge (CE) — The CE router is installed at the customer site. Depending upon the business model of the ISP, this router may be managed by the customer, the ISP, or both. The CE router connects to, and communicates with, the service provider routers, and allows the service provider to participate in customer routing.

- Provider Edge (PE) — Routers owned by the provider that actively participate in customer routing, guaranteeing optimum routing between customer sites. PE routers use a separate virtual routing table for each customer, resulting in perfect isolation between customers.

It is important to note that neither the C nor the CE routers need any special configuration. The P routers only require a simple MPLS LDP configuration.

In this lab, SP2 models the P router, SP1 and SP3 model the PE routers. HQ and BRANCH are both CE routers with loopback networks to simulate connections to other C routers.

The PE routers control the entire MPLS VPN from end to end. There are a number of relevant questions you may be asking: How can a single router determine which routes in its table belong to the service provider's internal network and which routes belong to each customer? How can the PE device allow customers to utilize existing networks including private addressing without creating routing problems?

The answer to all of these questions lies in the ability of routers to maintain virtual routing and forwarding (VRF) instances. Each VRF uses and maintains its own routing information base (RIB) and Cisco Express Forwarding (CEF) table. Interfaces are either assigned to specific VRF instances or they use the default RIB and CEF tables. The VRF instance's RIB fulfills the role of control plane while the VRF's CEF table fulfills the role of the data forwarding plane. Routing protocols between the PE and CE routers populate the VRF RIB and CEF makes forwarding decisions based on the routes in the VRF RIB. When an IP packet arrives on an interface that has been associated with a VRF, the packet is routed according to the CEF table for that VRF instance. CEF is the only IP switching protocol supported for VRF, so CEF should be enabled globally with the **ip cef** command and on the interfaces associated with the VRF instance.

However, PE routers must now be connected through the provider network to perform routing and forwarding between customer sites. The most efficient and only scalable method to achieve this is to use the multiprotocol extensions to BGP (MP-BGP) that enable it to carry routes for different routed protocols. PE routers will establish iBGP sessions with other PEs in your carrier network to exchange for each VPN routes. This will help populate the VRF routing tables on each of the PE routers with the VRF tables from other customer sites. CEF tables will be updated with the RIB information so that forwarding may occur between customer sites once the label-switched paths have been created through the provider network.

PE routers advertise routes that are part of their VPN using a new traffic class to distinguish these routes from internal routes in the provider's network. BGP uses a new address family called VPNv4 to carry MPLS-VPN routes to IPv4 networks. The VPNv4 address family is a 12-byte address consisting of an 8-byte route distinguisher (RD) and a 4-byte IPv4 address. The RD acts as a unique prefix when appended with the IPv4 address. Each VRF must have an RD for unique advertisement.

VRFs use the route target attribute to control the import and export of VPNv4 routes through iBGP. The route target is an extended BGP community that indicates which routes should be imported from MP-BGP into the VRF.

Exporting a route target (RT) means that the exported VPNv4 route receives an additional BGP extended community—this is the route target—when the route is redistributed from the VRF RIB into MP-BGP. Importing an RT means that the received VPNv4 route from MP-BGP is checked for a matching extended community—this is the route target—with the ones in the configuration.

To configure a VRF instance on the PE routers, use the **ip vrf** *name* command in global configuration mode on SP1 and SP3. At the VRF configuration prompt, create a VRF named “customer.” Each VRF instance will need a route distinguisher and a route target. The route distinguisher and route target are each eight bytes in length, with a colon separating four bytes on either side. There are various conventions for allocating route distinguishers for MPLS VPNs, the most useful of which is ASN:nn. Another popular notation is IP address:nn. In each of these cases, ‘nn’ represents an arbitrary value assigned by the network administrator. In this lab, use 100:1 as the route distinguisher. The route target is also an arbitrary eight-byte value used later in BGP.

Configure a route distinguisher (RD) of 100:1 and route target (RT) of 1:100 using the commands **rd** *ASN:nn* and **route-target** {**import | export | both**} *nn:nn*. In this case, you need to use the **both** keyword because you want PEs to import and export from that VRF.

```
SP1(config)# ip vrf customer
SP1(config-vrf)# rd 100:1
SP1(config-vrf)# route-target both 1:100

SP3(config)# ip vrf customer
SP3(config-vrf)# rd 100:1
SP3(config-vrf)# route-target both 1:100
```

Imagine that SP1 is running MP-BGP and it receives a VPNv4 route with a route target of 100:100. Given the above configuration, should BGP import the route into the *customer* VRF routing table?

After creating the VRFs, add interfaces to the VRF using the interface-level **ip vrf forwarding** *name* command, where name is the VRF instance name. Use this command on the interfaces of SP1 and SP3 (the PE routers) facing the CE routers. Add the IP address shown in the diagram to those interfaces, as well.

```
SP1(config)# interface fastethernet 0/0
SP1(config-if)# ip vrf forwarding customer
SP1(config-if)# ip address 172.16.100.254 255.255.255.0
SP1(config-if)# no shut

SP3(config)# interface serial 0/1/0
SP3(config-if)# ip vrf forwarding customer
```

```
SP3(config-if)# ip address 172.16.200.254 255.255.255.0
SP3(config-if)# no shutdown
```

You should now be able to ping across those the PE-CE links because you configured the other end of these links in Step 1. However, since these are not in the default routing table, you must use the **ping vrf** *name address* command. Since the VRF is transparent to the customer routers, you can use a traditional **ping** command when you are pinging from the C and CE routers.

```
SP1# ping vrf customer 172.16.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

HQ# ping 172.16.100.254

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.254, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/8 ms

SP3# ping vrf customer 172.16.200.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.200.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

BRANCH# ping 172.16.200.254

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.200.254, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

Step 5: Configure EIGRP AS 1

The service provider by whom you are employed uses the BGP AS 100. Your customer, the International Travel Agency, uses the BGP AS 1. To keep the configuration logically consistent, use the AS number 100 for EIGRP and BGP in the provider’s network and use the AS number 1 for EIGRP and BGP in the customer’s network. You will configure EIGRP AS 1 on the PE routers from within the configuration of the global EIGRP AS 100.

On the customer routers, configure EIGRP AS 1 for the major network 172.16.0.0. Disable automatic summarization.

```
HQ(config)# router eigrp 1
HQ(config-router)# no auto-summary
HQ(config-router)# network 172.16.0.0

BRANCH(config)# router eigrp 1
BRANCH(config-router)# no auto-summary
BRANCH(config-router)# network 172.16.0.0
```

Given only the information above, will EIGRP form any adjacencies immediately?

On the PE routers, the configuration is more complex. Every IGP has a different method of configuring a VRF for it. To implement EIGRP for VRFs, start the EIGRP process by configuring EIGRP AS 100. Remember, this AS belongs to the provider and is not significant to the customer. If you were using EIGRP as the service provider's IGP instead of OSPF, you would configure your network statements at this point .

```
SP1(config)# router eigrp 100
```

```
SP3(config)# router eigrp 100
```

Now, to configure EIGRP for an individual VRF instance, use the command **address-family ipv4 vrf** *name*, where name is the name of the VRF instance. Although each VPN must be logically separate from other IPv4 address spaces using VRF, this separation must extend not only to the routing table but to the routing protocols as well. The **address-family** command creates a logical segment of a routing protocol and its routes and adjacencies in order to separate it from other sets of routes and adjacencies. In this case, you will separate an EIGRP autonomous system from the EIGRP instance initiated with the **router eigrp 100** command. Networks learned via this new autonomous system will be injected into the VRF routing table associated with the isolated EIGRP AS. It is also important to note that these networks will not be advertised to any neighbors in EIGRP AS 100; it is completely separate from the rest of the EIGRP domain.

```
SP1(config-router)# address-family ipv4 vrf customer
SP1(config-router-af)# autonomous-system 1
SP1(config-router-af)# no auto-summary
SP1(config-router-af)# network 172.16.0.0

SP3(config-router)# address-family ipv4 vrf customer
SP3(config-router-af)# autonomous-system 1
SP3(config-router-af)# no auto-summary
SP3(config-router-af)# network 172.16.0.0
```

On the PE routers, display the default routing table with the **show ip route** command. Notice that the PE routers do not possess any routes from the 172.16.0.0/16 major network in the default routing table. Display the VRF routing table with the **show ip route vrf** *name* command, where name is the VRF instance name.

```
SP1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.0.12.0/24 is directly connected, Serial0/0/0
O       10.0.3.1/32 [110/129] via 10.0.12.2, 05:29:59, Serial0/0/0
O       10.0.2.1/32 [110/65] via 10.0.12.2, 05:29:59, Serial0/0/0
C       10.0.1.1/32 is directly connected, Loopback0
O       10.0.23.0/24 [110/128] via 10.0.12.2, 05:29:59, Serial0/0/0

SP1# show ip route vrf customer

Routing Table: customer
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
D       172.16.10.0 [90/156160] via 172.16.100.1, 00:03:29, FastEthernet0/0
C       172.16.100.0 is directly connected, FastEthernet0/0

SP3# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O       10.0.12.0/24 [110/128] via 10.0.23.2, 05:30:42, Serial0/0/1
C       10.0.3.1/32 is directly connected, Loopback0
O       10.0.2.1/32 [110/65] via 10.0.23.2, 05:30:42, Serial0/0/1
O       10.0.1.1/32 [110/129] via 10.0.23.2, 05:30:42, Serial0/0/1
C       10.0.23.0/24 is directly connected, Serial0/0/1

SP3# show ip route vrf customer

Routing Table: customer
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.200.0 is directly connected, Serial0/1/0
D       172.16.20.0 [90/2297856] via 172.16.200.1, 00:02:06, Serial0/1/0
```

The SP1 and HQ routers do not possess routes to the customer networks on SP3 and BRANCH and vice versa. Explain why this occurs though EIGRP adjacencies have formed:

Step 6: Configure BGP

Now that the PE routers are routing to the CE routers over VRF tables, you can set up the PE routers to exchange routes through BGP. First, configure BGP between SP1 and SP3 and have them peer between their loopback addresses. Synchronization should be disabled by default on newer IOS releases. If synchronization is not already disabled, then explicitly disable it using the **no synchronization** command. For more information on configuring BGP, refer to CCNP1.

```
SP1(config)# router bgp 100
SP1(config-router)# neighbor 10.0.3.1 remote-as 100
SP1(config-router)# neighbor 10.0.3.1 update-source loopback0

SP3(config)# router bgp 100
SP3(config-router)# neighbor 10.0.1.1 remote-as 100
SP3(config-router)# neighbor 10.0.1.1 update-source loopback0
```

To configure the exchange of VPNv4 routes over BGP, use the **address-family vpnv4** command. At the address family prompt, activate the BGP neighbor for this address family with **neighbor** *address* **activate** command. Activating a neighbor for an address family allows BGP to send routes to and receive routes from the designated neighbor using the specified address family. By default, neighbors are only activated for IPv4.

The RTs are translated as extended BGP communities, so you must allow SP1 and SP3 to send both standard and extended communities over MP-BGP using the **neighbor** *address* **send-community both** command. The adjacencies may flap (temporarily go down and then come back up) when you activate the address family.

```
SP1(config-router)# address-family vpnv4
SP1(config-router-af)# neighbor 10.0.3.1 activate
SP1(config-router-af)# neighbor 10.0.3.1 send-community both
SP1(config-router-af)# exit

SP3(config-router)# address-family vpnv4
SP3(config-router-af)# neighbor 10.0.1.1 activate
```

```
SP3(config-router-af)# neighbor 10.0.1.1 send-community both
SP3(config-router-af)# exit
```

Finally, you need to configure BGP to redistribute the EIGRP routes in the VRF RIB into the BGP protocol so that these are advertised to the remote PE. Under the main BGP configuration prompt, enter another address family associated only with the routing table for the VRF *customer*. Redistribute the EIGRP routes that are associated with this VRF into BGP.

```
SP1(config-router)# address-family ipv4 vrf customer
SP1(config-router-af)# redistribute eigrp 1
SP1(config-router-af)# exit
SP1(config-router)# exit
```

```
SP3(config-router)# address-family ipv4 vrf customer
SP3(config-router-af)# redistribute eigrp 1
SP3(config-router-af)# exit
SP3(config-router)# exit
```

Based on the above configuration, will SP1's VRF RIB contain the 172.16.20.0/24 route that was originated by EIGRP on BRANCH? Explain.

Will HQ learn the same routes via EIGRP? Explain.

Do you expect to see the redistributed routes as internal or external EIGRP routes on the CE routers? Explain.

Enter the EIGRP instance that contains the VRF configuration on SP1 and SP3 and configure it to redistribute BGP routes. Since you are redistributing into EIGRP from BGP, the metrics are not comparable. Add a seed metric with a bandwidth of 64 kbps, 100 microseconds, reliability of 255/255, load of 1/255, and MTU of 1500 bytes.

```
SP1(config)# router eigrp 100
```

```
SP1(config-router)# address-family ipv4 vrf customer
SP1(config-router-af)# redistribute bgp 100 metric 64 1000 255 1 1500

SP3(config)# router eigrp 100
SP3(config-router)# address-family ipv4 vrf customer
SP3(config-router-af)# redistribute bgp 100 metric 64 1000 255 1 1500
```

Step 7: Investigate Control Plane Operation

Remember that MPLS differentiates the control plane from the forwarding plane. The control plane, represented by the routing table (the RIB) and the routing protocols, must operate so that the VRF routes reach remote PEs and are installed as necessary in the VRF routing tables. Not only the prefixes, but also the accompanying metrics and tags are important to the reconstruction of the route at the remote PE. Fortunately, MP-BGP allows you to send these metrics in the Network Layer Reachability Information (NLRI).

Through this step and also Step 8, you will investigate the routing and forwarding information associated with the route to 172.16.20.0/24.

Verify that the routes have propagated to the remote PE routers. Issue the **show ip route vrf** *name* command to see the VRF RIB. Notice the source of the routes on the PE routers.

```
SP1# show ip route vrf customer

Routing Table: customer
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
B       172.16.200.0 [200/0] via 10.0.3.1, 00:06:44
B       172.16.20.0 [200/2297856] via 10.0.3.1, 00:06:44
D       172.16.10.0 [90/156160] via 172.16.100.1, 00:17:34, FastEthernet0/0
C       172.16.100.0 is directly connected, FastEthernet0/0

SP3# show ip route vrf customer

Routing Table: customer
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
C       172.16.200.0 is directly connected, Serial0/1/0
```

```
D       172.16.20.0 [90/2297856] via 172.16.200.1, 16:47:37, Serial0/1/0
B       172.16.10.0 [200/156160] via 10.0.1.1, 00:17:28
B       172.16.100.0 [200/0] via 10.0.1.1, 00:17:28
```

You may be asking “Why does the source of the route to 172.16.20.0/24 on SP1 point to 10.0.3.1 since that address would be routed based on the default routing table?” Consider that when an internally generated route is sent to an iBGP peer, BGP sets the NEXT-HOP attribute to be the advertising router. In this case, SP3 generates the route in BGP by redistribution. The BGP peers are communicating between loopback interfaces, so the NEXT-HOP is set to the IP address of the BGP peer’s source interface. The VRF RIB thus points to an interface that must be reached through the default, global RIB. We will investigate the forwarding for packets destined for these networks in the next step.

On the CE routers, issue the **show ip route** command to see a full routing table.

```
HQ# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
D       172.16.200.0
           [90/2172416] via 172.16.100.254, 00:05:17, FastEthernet0/0
D       172.16.20.0 [90/2300416] via 172.16.100.254, 00:05:17, FastEthernet0/0
C       172.16.10.0 is directly connected, Loopback0
C       172.16.100.0 is directly connected, FastEthernet0/0

BRANCH# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
C       172.16.200.0 is directly connected, Serial0/0/0
C       172.16.20.0 is directly connected, Loopback0
D       172.16.10.0 [90/2300416] via 172.16.200.254, 00:02:02, Serial0/0/0
D       172.16.100.0 [90/2172416] via 172.16.200.254, 00:02:02, Serial0/0/0
```

On both the CE and PE routers, notice that the routes you redistributed from BGP into EIGRP are *internal* EIGRP routes because BGP preserves features of the EIGRP route while advertising the route to the other PEs. The PE encodes as much EIGRP information as possible into new extended communities TLV tuples (type, length, value) to preserve route characteristics through the VPN. This enables the remote PE router to reconstruct the EIGRP route with all of its characteristics, including the metric components, AS, TAG and, for external routes, the remote AS number, the remote ID, the remote protocol, and the remote metric. These are the EIGRP characteristics of a prefix that you can find in the topology table. If the EIGRP-advertised route is internal, the route is advertised as an internal route into the remote site if the destination AS matches the source AS carried by the BGP extended community attributes.

Display information on the VPNv4 BGP routes on SP1 with the **show bgp vpnv4 unicast all** command.

```
SP1# show bgp vpnv4 unicast all
BGP table version is 9, local router ID is 10.0.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf customer)
*> 172.16.10.0/24   172.16.100.1        156160         32768 ?
*>i172.16.20.0/24   10.0.3.1           2297856    100      0 ?
*> 172.16.100.0/24  0.0.0.0                  0         32768 ?
*>i172.16.200.0/24  10.0.3.1                 0    100      0 ?

SP3# show bgp vpnv4 unicast all
BGP table version is 9, local router ID is 10.0.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf customer)
*>i172.16.10.0/24   10.0.1.1            156160    100      0 ?
*> 172.16.20.0/24   172.16.200.1       2297856         32768 ?
*>i172.16.100.0/24  10.0.1.1                 0    100      0 ?
*> 172.16.200.0/24  0.0.0.0                  0         32768 ?
```

Notice that the metric (MED value) in BGP is the metric advertised through EIGRP for that route as well.

What does the value of the NEXT-HOP attribute for the 172.16.200.0/24 network on SP3 indicate?

What is the value of the BGP NEXT-HOP attribute for the 172.16.20.0/24 route on SP1?

By which routing protocol and from which router was the route to 10.0.3.1/32 installed in the default routing table on SP1?

View more specific detail on a particular prefix using **show bgp vpnv4 unicast all** *ip-address* command. Notice that the MPLS label information is included. Execute this on both of the PEs. Remember that SP3 is advertising the 172.16.20.0/24 prefix through BGP, while SP1 is receiving the route through BGP NLRI.

```
SP1# show bgp vpnv4 unicast all 172.16.20.0/24
BGP routing table entry for 100:1:172.16.20.0/24, version 15
Paths: (1 available, best #1, table customer)
Flag: 0x820
  Not advertised to any peer
  Local
    10.0.3.1 (metric 129) from 10.0.3.1 (10.0.3.1)
      Origin incomplete, metric 2297856, localpref 100, valid, internal, best
      Extended Community: RT:1:100
        Cost:pre-bestpath:128:2297856 (default-2145185791) 0x8800:32768:0
        0x8801:1:640000 0x8802:65281:1657856 0x8803:65281:1500
      mpls labels in/out nolabel/20

SP3# show bgp vpnv4 unicast all 172.16.20.1
BGP routing table entry for 100:1:172.16.20.0/24, version 15
Paths: (1 available, best #1, table customer)
  Advertised to update-groups:
     1
  Local
    172.16.200.1 from 0.0.0.0 (10.0.3.1)
      Origin incomplete, metric 2297856, localpref 100, weight 32768, valid,
sourced, best
      Extended Community: RT:1:100
        Cost:pre-bestpath:128:2297856 (default-2145185791) 0x8800:32768:0
        0x8801:1:640000 0x8802:65281:1657856 0x8803:65281:1500
      mpls labels in/out 20/nolabel
```

Notice that there are multiple values in the BGP extended communities. Recall that BGP sends the route information in NLRI as extended communities. These values are TLVs indicating such EIGRP attributes as the TAG, AS number, bandwidth, delay, reliability, load, MTU, and hop count.

Why is the origin code “incomplete”?

What type of attribute carries the route target information in MP-BGP NLRI?

Notice the MPLS labels indicated for the BGP route above. The in-label of ‘nolabel’ on SP1 indicates that SP1 is not advertising a label for the prefix 172.16.20.0/24. The out-label of 21 is advertised by SP3 and received by SP1. This label is significant only on the path between SP1 and SP3. This label has been allocated by BGP on SP3.

View the list of MPLS labels that are being used with BGP using **show bgp vpnv4 unicast all labels**.

```
SP1# show bgp vpnv4 unicast all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (customer)
   172.16.10.0/24   172.16.100.1    19/nolabel
   172.16.20.0/24   10.0.3.1        nolabel/20
   172.16.100.0/24  0.0.0.0         20/aggregate(customer)
   172.16.200.0/24  10.0.3.1        nolabel/19

SP3# show bgp vpnv4 unicast all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (customer)
   172.16.10.0/24   10.0.1.1        nolabel/19
   172.16.20.0/24   172.16.200.1    20/nolabel
   172.16.100.0/24  10.0.1.1        nolabel/20
   172.16.200.0/24  0.0.0.0         19/aggregate(customer)
```

How has SP1 learned the VPN label, label 20?

Will SP1 or SP2 learn the label via LDP?

Has the P router SP3 learned about label 20 from SP3? Explain.

Finally, display the route attributes for the same prefix, 172.16.20.0/24, in the EIGRP topology table on SP1 with the **show ip eigrp vrf customer topology** *ip-prefix/mask* command. Verify this against the originator of the EIGRP route in BGP, SP3.

```
SP1# show ip eigrp vrf customer topology 172.16.20.0/24
IP-EIGRP (AS 1): Topology entry for 172.16.20.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2297856
  Routing Descriptor Blocks:
  10.0.3.1, from VPNv4 Sourced, Send flag is 0x0
      Composite metric is (2297856/0), Route is Internal (VPNv4 Sourced)
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1

SP3# show ip eigrp vrf customer topology 172.16.20.0/24
IP-EIGRP (AS 1): Topology entry for 172.16.20.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2297856
  Routing Descriptor Blocks:
  172.16.200.1 (Serial0/1/0), from 172.16.200.1, Send flag is 0x0
      Composite metric is (2297856/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

Notice that there is absolutely no difference in the EIGRP route parameters between SP1 and SP3. BGP encodes and decodes the information on the PE routers with no changes.

Remember that SP2, a P router, has no knowledge of individual routes in the VRF tables on the PE routers. You can verify this with the show commands performed previously.

```
SP2# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.0.12.0/24 is directly connected, Serial0/0/0
O       10.0.3.1/32 [110/65] via 10.0.23.3, 1d00h, Serial0/0/1
C       10.0.2.1/32 is directly connected, Loopback0
O       10.0.1.1/32 [110/65] via 10.0.12.1, 1d00h, Serial0/0/0
C       10.0.23.0/24 is directly connected, Serial0/0/1

SP2# show ip route vrf customer
% IP routing table customer does not exist
```

Ping between the CE routers to verify connectivity through the MPLS VPN.

```
HQ# ping 172.16.20.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/89/93 ms

BRANCH# ping 172.16.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/86/88 ms
```

Step 8: Investigate Forwarding Plane Operation

Recall that MPLS has two tables, the Label Information Base (LIB) and the Label Forwarding Information Base (LFIB). Normally, LDP-allocated labels are advertised to LDP peers. BGP-allocated labels are advertised to BGP peers. BGP-allocated labels will be used by BGP peers as an MPLS label on packets destined for that network through the VPN. The BGP-allocated labels are only significant to the ingress and egress routers. P routers that are not BGP peers with the PE routers will not see the VPN label for the networks known by BGP.

In order to traverse the MPLS cloud, the packets need to be label-switched at every hop based on advertised labels. In order to ensure that VPN packets that reach the egress PE have the MPLS label needed to switch the packets once they arrive, the labels are stacked at the ingress PE. However, the packet still needs to be sent along the label-switched path.

Recall that the VRF RIB's next hop for the networks known via the VPN point to the loopback on the egress PE. CEF uses the 'inuse' label for the BGP next hop as the outermost label for packets traveling through the MPLS VPN. First, however, CEF must push on the VPN label that will be used at the egress PE. Thus, CEF stacks the label in a sequential manner so that the VPN label is available at the egress PE, but the label to traverse the label-switched path through the P routers is pushed as the outermost label.

Take some time to study and understand the details of how this is possible. BGP, LDP, CEF, the LFIB, and the provider's IGP are all involved in the use of MPLS labels as a VPN technology.

Once BGP learns the MPLS label to use as the VPN label, this information is entered into the CEF forwarding table on the ingress PE. Display the CEF forwarding entry for 172.16.20.0/24 on SP1 with the **show ip cef vrf name** *ip-address* command.

```
SP1# show ip cef vrf customer 172.16.20.0
172.16.20.0/24, version 12, epoch 0, cached adjacency to Serial0/0/0
0 packets, 0 bytes
  tag information set
    local tag: VPN-route-head
    fast tag rewrite with Se0/0/0, point2point, tags imposed: {16 20}
  via 10.0.3.1, 0 dependencies, recursive
    next hop 10.0.12.2, Serial0/0/0 via 10.0.3.1/32
    valid cached adjacency
    tag rewrite with Se0/0/0, point2point, tags imposed: {16 20}
```

CEF resolves the recursive lookup to the BGP next hop. Based on the labels learned by LDP, CEF may or may not apply the forwarding label to reach 10.0.3.1/32. In this case, LDP on SP2 has advertised a forwarding label to SP1. View the labels advertised to SP1 via LDP using the **show mpls ip binding** command.

```
SP1# show mpls ip binding
  10.0.1.1/32
        in label:     imp-null
        out label:    17        lsr: 10.0.2.1:0
  10.0.2.1/32
        in label:     16
        out label:    imp-null  lsr: 10.0.2.1:0       inuse
  10.0.3.1/32
        in label:     17
        out label:    16        lsr: 10.0.2.1:0       inuse
  10.0.12.0/24
        in label:     imp-null
        out label:    imp-null  lsr: 10.0.2.1:0
  10.0.23.0/24
        in label:     18
        out label:    imp-null  lsr: 10.0.2.1:0       inuse
```

CEF pushes the label of 20 onto the packet first, then pushes the outer label of 16. The CEF forwarding table decides which path to use based of course on the default RIB. The route has been installed in the RIB by OSPF. Thus, the ingress PE imposes two labels in the sequence {16, 20} as shown in the CEF forwarding table above.

Since the incoming VPN packets from SP1 are encapsulated in MPLS frames, SP2 acts according to the directives in its LFIB. SP2 is also the penultimate hop in the label-switched path from SP1 to SP3’s loopback interface, and therefore pops the outermost label from the MPLS frame. Display the LFIB with the **show mpls forwarding-table** command.

```
SP2# show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     10.0.3.1/32       5175       Se0/0/1     point2point
17     Pop tag     10.0.1.1/32       8079       Se0/0/0     point2point
```

Notice that the LFIB does not care whether there is an inner label or not, it simply performs the operation specified in the column labeled “Outgoing tag or VC.”

If you enable MPLS packet debugging on SP2 using **debug mpls packets** then issue a ping from one CE to the other, you can see the MPLS packets being label-switched. The ICMP packets are forwarded inside MPLS frames through SP2. Notice in the debug output that each ICMP echo request receives a reply which is label-switched on its return path through the MPLS network. When you are done, disable debugging.

```
SP2# debug mpls packets
MPLS packet debugging is on

HQ# ping 172.16.20.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/87/92 ms

SP2#
*Feb  3 20:55:57.422: MPLS: Se0/0/0: recvd: CoS=0, TTL=254, Label(s)=16/20
*Feb  3 20:55:57.422: MPLS: Se0/0/1: xmit: CoS=0, TTL=253, Label(s)=20
!
! These 2 messages indicate the label-switching of the ICMP echo request
!
*Feb  3 20:55:57.478: MPLS: Se0/0/1: recvd: CoS=0, TTL=254, Label(s)=17/20
*Feb  3 20:55:57.478: MPLS: Se0/0/0: xmit: CoS=0, TTL=253, Label(s)=20
!
! These 2 messages indicate the label-switching of the ICMP echo reply
!
*Feb  3 20:55:57.510: MPLS: Se0/0/0: recvd: CoS=0, TTL=254, Label(s)=16/20
*Feb  3 20:55:57.510: MPLS: Se0/0/1: xmit: CoS=0, TTL=253, Label(s)=20
*Feb  3 20:55:57.566: MPLS: Se0/0/1: recvd: CoS=0, TTL=254, Label(s)=17/20
*Feb  3 20:55:57.566: MPLS: Se0/0/0: xmit: CoS=0, TTL=253, Label(s)=20
*Feb  3 20:55:57.598: MPLS: Se0/0/0: recvd: CoS=0, TTL=254, Label(s)=16/20
*Feb  3 20:55:57.598: MPLS: Se0/0/1: xmit: CoS=0, TTL=253, Label(s)=20
*Feb  3 20:55:57.654: MPLS: Se0/0/1: recvd: CoS=0, TTL=254, Label(s)=17/20
*Feb  3 20:55:57.654: MPLS: Se0/0/0: xmit: CoS=0, TTL=253, Label(s)=20
*Feb  3 20:55:57.686: MPLS: Se0/0/0: recvd: CoS=0, TTL=254, Label(s)=16/20
*Feb  3 20:55:57.686: MPLS: Se0/0/1: xmit: CoS=0, TTL=253, Label(s)=20
*Feb  3 20:55:57.742: MPLS: Se0/0/1: recvd: CoS=0, TTL=254, Label(s)=17/20
*Feb  3 20:55:57.742: MPLS: Se0/0/0: xmit: CoS=0, TTL=253, Label(s)=20
*Feb  3 20:55:57.774: MPLS: Se0/0/0: recvd: CoS=0, TTL=254, Label(s)=16/20
*Feb  3 20:55:57.774: MPLS: Se0/0/1: xmit: CoS=0, TTL=253, Label(s)=20
*Feb  3 20:55:57.830: MPLS: Se0/0/1: recvd: CoS=0, TTL=254, Label(s)=17/20
*Feb  3 20:55:57.830: MPLS: Se0/0/0: xmit: CoS=0, TTL=253, Label(s)=20

SP2# undebug all
All possible debugging has been turned off
```

Continue tracing the label-switched path through the provider network to the egress PE, SP3.

Based on which forwarding table will the VPN packet be switched at SP3? Explain.

Display the MPLS LFIB on SP3 using the **show mpls forwarding-table** command that you used on SP2 previously.

```
SP3# show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     10.0.12.0/24      0          Se0/0/1    point2point
17     Pop tag     10.0.2.1/32       0          Se0/0/1    point2point
18     17          10.0.1.1/32       0          Se0/0/1    point2point
19     Aggregate   172.16.200.0/24[V]   \
                                     2704
20     Untagged    172.16.20.0/24[V] 2704       Se0/1/0    point2point
```

Notice that SP3 forwards the decapsulated IP packet untagged to the Serial 0/1/0 egress interface because it was received with a label of 20. This is the label that BGP advertised to SP1. SP1's CEF forwarding table encapsulated the IP packets within two MPLS labels {16 20} and then forwarded the packet to SP2.

Conclusion

Issue the **traceroute** command from one CE to another to find that it is going through multiple Layer 3 hops. This is an important debugging tool because it can also be issued from a PE router with reference to a VRF.

```
HQ# traceroute 172.16.20.1

Type escape sequence to abort.
Tracing the route to 172.16.20.1

  1 172.16.100.254 0 msec 0 msec 0 msec
  2 10.0.12.2 126 msec 117 msec 126 msec
  3 172.16.200.254 59 msec 50 msec 50 msec
  4 172.16.200.1 50 msec 42 msec *
```

Fill in the following table tracing the path of packets from 172.16.100.1 to 172.16.20.1 in order to trace the packet's path:

Router	Incoming (MPLS/IP)	Outgoing (MPLS/IP)	Switched By (CEF/LFIB)	Incoming Label(s)	Outgoing Label(s)

HQ						
SP1						
SP2						
SP3						
BRANCH						

Given the output shown below on each of the routers, trace the return path from 172.16.20.1 to 172.16.100.1 by filling in the chart.

```
BRANCH# show ip cef 172.16.100.1
172.16.100.0/24, version 22, epoch 0, cached adjacency to Serial0/0/0
0 packets, 0 bytes
  via 172.16.200.254, Serial0/0/0, 0 dependencies
    next hop 172.16.200.254, Serial0/0/0
    valid cached adjacency

SP3# show ip cef vrf customer 172.16.100.1
172.16.100.0/24, version 6, epoch 0, cached adjacency to Serial0/0/1
0 packets, 0 bytes
  tag information set
    local tag: VPN-route-head
    fast tag rewrite with Se0/0/1, point2point, tags imposed: {17 20}
  via 10.0.1.1, 0 dependencies, recursive
    next hop 10.0.23.2, Serial0/0/1 via 10.0.1.1/32
    valid cached adjacency
    tag rewrite with Se0/0/1, point2point, tags imposed: {17 20}

SP2# show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     10.0.3.1/32       15601      Se0/0/1    point2point
17     Pop tag     10.0.1.1/32       25413      Se0/0/0    point2point

SP1# show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     10.0.2.1/32       0          Se0/0/0    point2point
17     16          10.0.3.1/32       0          Se0/0/0    point2point
18     Pop tag     10.0.23.0/24      0          Se0/0/0    point2point
19     Untagged    172.16.10.0/24[V] 0          Fa0/0      172.16.100.1
20     Aggregate   172.16.100.0/24[V]   \0
```

Router	Incoming (MPLS/IP)	Outgoing (MPLS/IP)	Switched By (CEF/LFIB)	Incoming Label(s)	Outgoing Label(s)
BRANCH					
SP3					

SP2

SP1

HQ

Final Configurations

```
HQ# show run
hostname HQ
!
interface Loopback0
 ip address 172.16.10.1 255.255.255.0
 no shutdown
!
interface FastEthernet0/0
 ip address 172.16.100.1 255.255.255.0
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
end

SP1# show run
hostname SP1
!
ip vrf customer
 rd 100:1
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.1.1 255.255.255.255
!
interface FastEthernet0/0
 ip vrf forwarding customer
 ip address 172.16.100.254 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 ip address 10.0.12.1 255.255.255.0
 mpls ip
 clock rate 64000
 no shutdown
!
router eigrp 100
 no auto-summary
 !
 address-family ipv4 vrf customer
 redistribute bgp 100 metric 64 1000 255 1 1500
 network 172.16.0.0
 no auto-summary
 autonomous-system 1
 exit-address-family
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
```

```
!
router bgp 100
 no synchronization
 neighbor 10.0.3.1 remote-as 100
 neighbor 10.0.3.1 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.0.3.1 activate
 neighbor 10.0.3.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf customer
 redistribute eigrp 1
 no synchronization
 exit-address-family
!
mpls ldp router-id Loopback0 force
end

SP2# show run
hostname SP2
!
interface Loopback0
 ip address 10.0.2.1 255.255.255.255
!
interface Serial0/0/0
 ip address 10.0.12.2 255.255.255.0
 mpls ip
 no shutdown
!
interface Serial0/0/1
 ip address 10.0.23.2 255.255.255.0
 mpls ip
 clock rate 64000
 no shutdown
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
!
mpls ldp router-id Loopback0 force
end

SP3# show run
hostname SP3
!
ip vrf customer
 rd 100:1
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.3.1 255.255.255.255
!
interface Serial0/0/1
 ip address 10.0.23.3 255.255.255.0
 mpls ip
 no shutdown
!
interface Serial0/1/0
 ip vrf forwarding customer
 ip address 172.16.200.254 255.255.255.0
 no shutdown
```

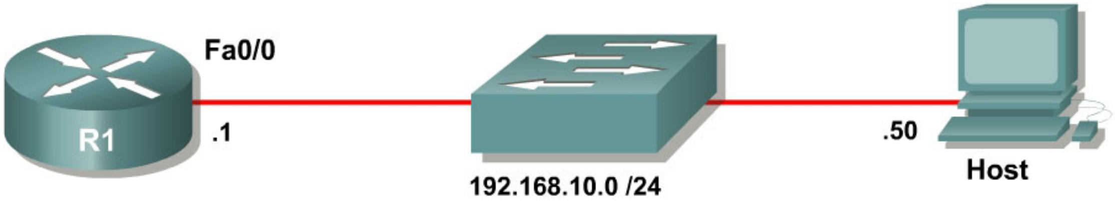
```
!
router eigrp 100
 no auto-summary
 !
 address-family ipv4 vrf customer
 redistribute bgp 100 metric 64 1000 255 1 1500
 network 172.16.0.0
 no auto-summary
 autonomous-system 1
 exit-address-family
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 100
 no synchronization
 neighbor 10.0.1.1 remote-as 100
 neighbor 10.0.1.1 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.0.1.1 activate
 neighbor 10.0.1.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf customer
 redistribute eigrp 1
 no synchronization
 exit-address-family
!
mpls ldp router-id Loopback0 force
end

BRANCH# show run
hostname BRANCH
!
interface Loopback0
 ip address 172.16.20.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.200.1 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
end
```

Lab 5.1 Using SDM One-Step Lockdown

Learning Objectives

- Install the Nmap application on a host
- Use Cisco Security Device Manager (SDM) One-Step Lockdown
- Use Nmap before and after to verify results

Topology Diagram



Scenario

Your CIO is rolling out a new security policy in the corporate network. Your manager notes that several of the requirements in the security policy can be accomplished automatically using one-step lockdown. One-step lockdown is a feature of SDM that allows automatic configuration of best practice security commands.

In order to rapidly implement the security policy, implement one-step lockdown using SDM, and verify the results using Nmap, a tool that allows you to identify remote system types with a port scan.

Step 1: Configure Addressing

Configure the router interface shown in the topology diagram.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
```

On the host, download Nmap for Windows from the following URL: http://insecure.org/Nmap/download.html.

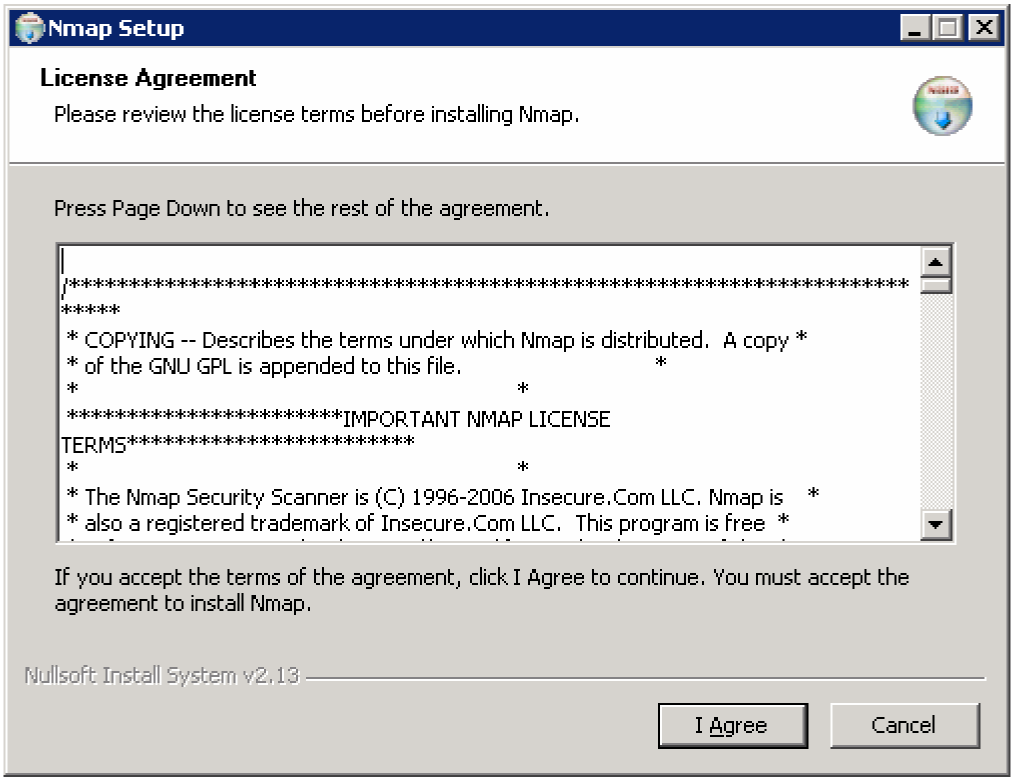
After you have downloaded the file, configure the host with the IP address 192.168.10.50 as shown in the diagram. If you do not know how to change an IP address in Windows, refer to Lab 3.1: Configuring a Router with SDM.

If you use a switch to connect routers to hosts, erase the switch’s configuration and leave the switch in its default configuration for the entirety of this lab exercise. By default, the switch assigns all switchports to VLAN 1. Thus, R1 and the PC host shown in the diagram will communicate across VLAN 1. If you are accessing your equipment remotely, follow the instructions given by your instructor concerning the configuration of the switch.

Step 2: Install Nmap on the Host

Run the installer program. Click **I Agree** to accept the licensing agreement for Nmap.

Note: At the time this lab was written, Nmap only partially works on computers running Microsoft Windows XP Service Pack 2. This may change in the future, so check the Nmap download page for updates.



License Agreement

Please review the license terms before installing Nmap.

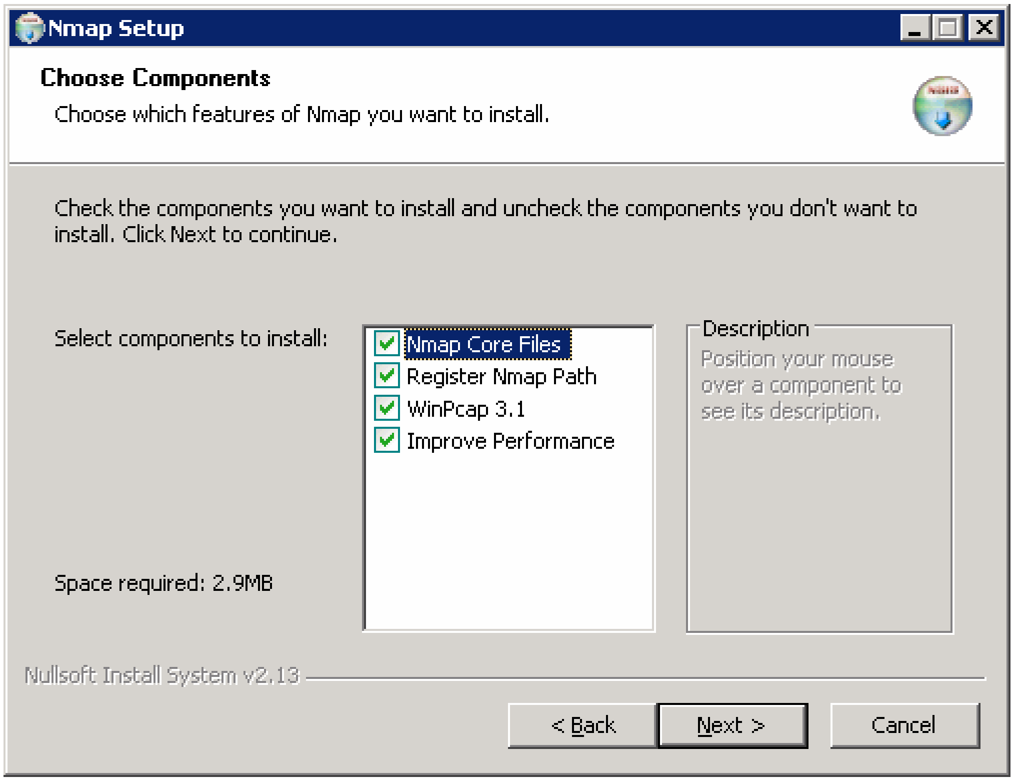
Press Page Down to see the rest of the agreement.

```
/************************************************************************
*****
 * COPYING -- Describes the terms under which Nmap is distributed.  A copy *
 * of the GNU GPL is appended to this file.                      *
 *                                                  *
 ***********************IMPORTANT NMAP LICENSE
TERMS*************************
 *                                                  *
 * The Nmap Security Scanner is (C) 1996-2006 Insecure.Com LLC. Nmap is    *
 * also a registered trademark of Insecure.Com LLC.  This program is free  *
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Nmap.

Figure 2-1: Nmap License Agreement

On the **Choose Components** page, do not change the default components that are checked. Click **Next** to install the defaults.



Nmap Setup

Choose Components

Choose which features of Nmap you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ Nmap Core Files
☑ Register Nmap Path
☑ WinPcap 3.1
☑ Improve Performance

Description

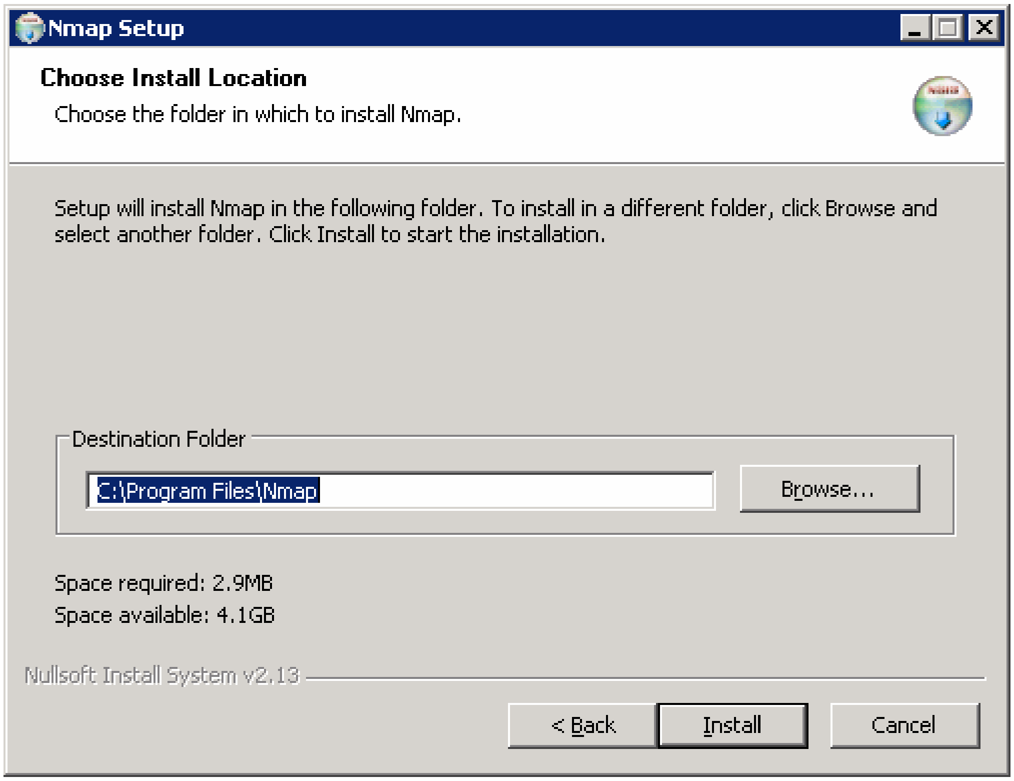
Position your mouse over a component to see its description.

Space required: 2.9MB

Nullsoft Install System v2.13

Figure 2-2: Nmap Setup, Components Selection

Use the default installation directory. Click **Next**.



Nmap Setup

Choose Install Location

Choose the folder in which to install Nmap.

Setup will install Nmap in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files\Nmap

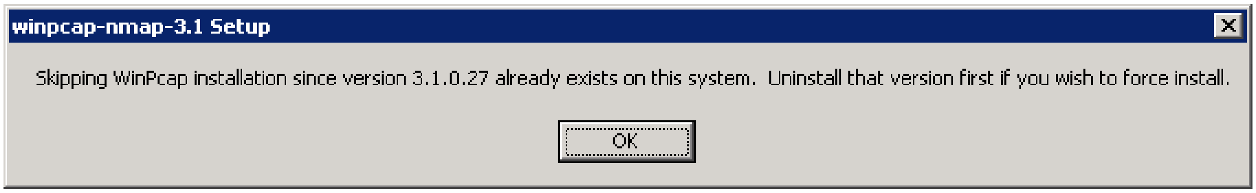
Space required: 2.9MB

Space available: 4.1GB

Nullsoft Install System v2.13

Figure 2-3: Nmap Installation Location

If you have previously installed other security tools that use WinPcap, a popup window may appear notifying you that WinPcap is already installed and will be skipped as part of the Nmap installation. If this notification appears, as shown in Figure 2-4, click **OK,** and skip to Figure 2-7 toward the end of this step. If this notification does not appear, proceed normally through this step.

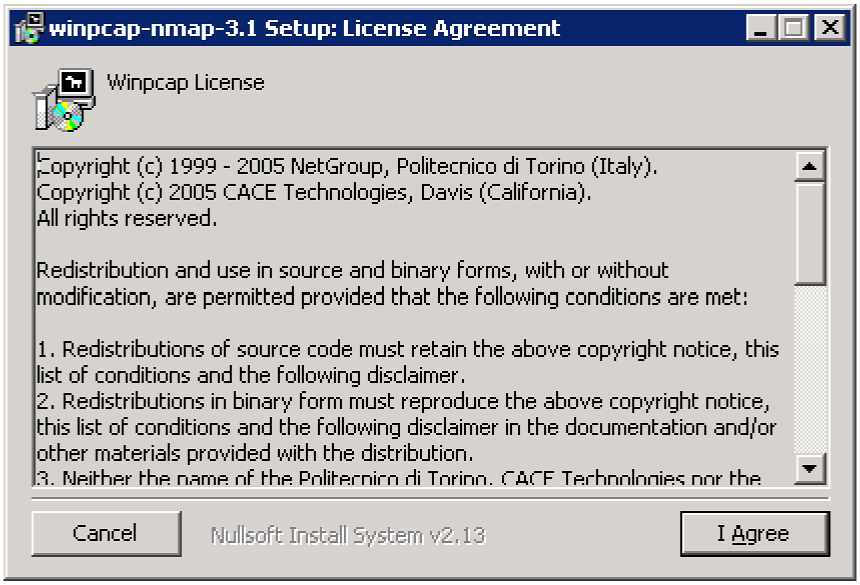


winpcap-nmap-3.1 Setup

Skipping WinPcap installation since version 3.1.0.27 already exists on this system. Uninstall that version first if you wish to force install.

Figure 2-4: WinPcap Installed Notification

Click **I Agree** to accept the license agreement for WinPcap.



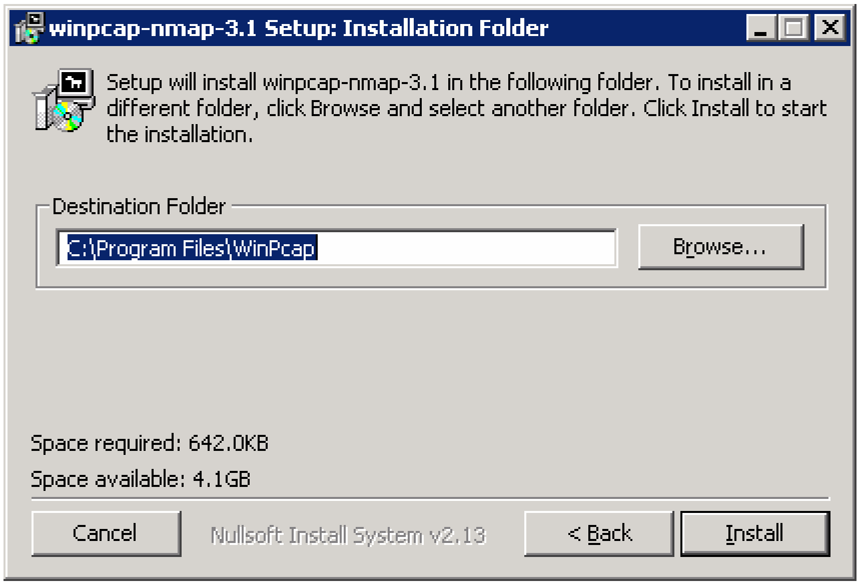
Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy).
Copyright (c) 2005 CACE Technologies, Davis (California).
All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Politecnico di Torino, CACE Technologies nor the

Figure 2-5: WinPcap License Agreement

Accept the default installation directory, or choose your own installation directory, and then click **Install**.



winpcap-nmap-3.1 Setup: Installation Folder

Setup will install winpcap-nmap-3.1 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

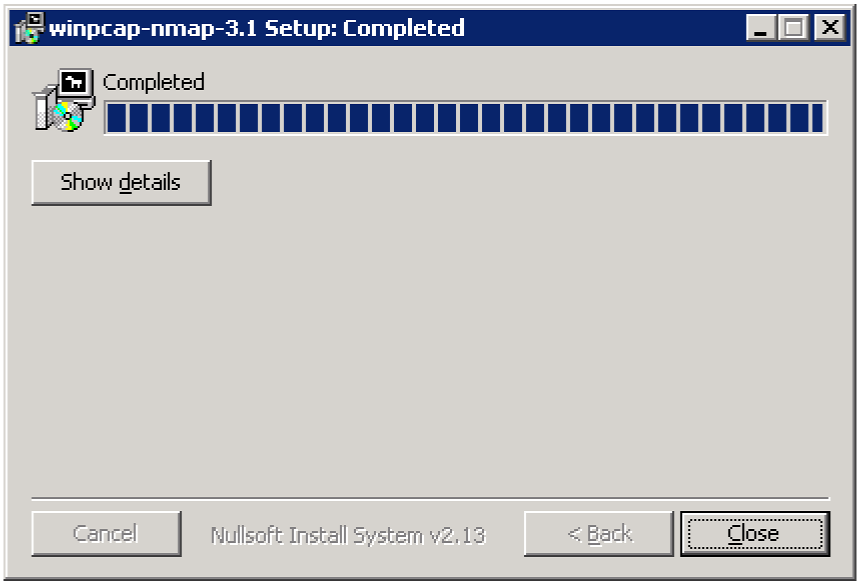
Destination Folder: C:\Program Files\WinPcap

Space required: 642.0KB
Space available: 4.1GB

Browse...
Cancel
Nullsoft Install System v2.13
< Back
Install

Figure 2-6: WinPcap Installation Location Dialog

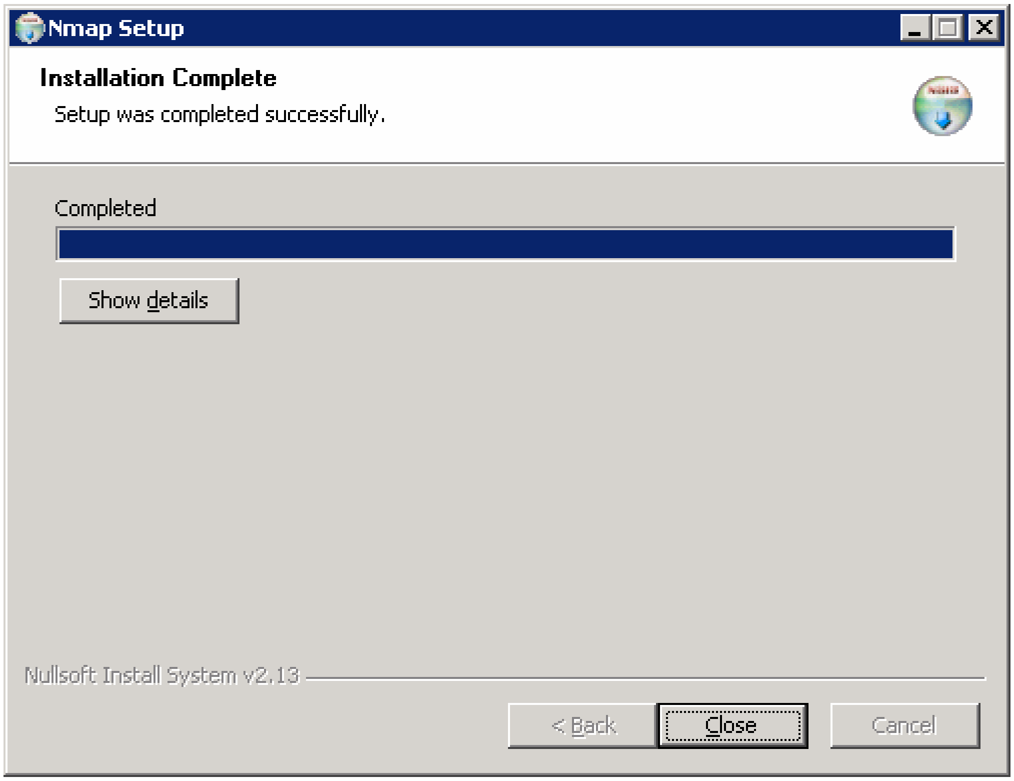
Click **Close** to complete the WinPcap installation. You are returned to the Nmap installation.


winpcap-nmap-3.1 Setup: Completed

Completed

Figure 2-7: WinPcap Installation Progress Indicator

When the Nmap installation is complete, click **Close** to end the installation wizard.


Nmap Setup

Installation Complete

Setup was completed successfully.

Completed

Figure 2-8: Nmap Installation Progress Indicator

Step 3: Run a Port Scan with Nmap

Nmap for Windows runs as a text-based command line program. To run Nmap, open the Microsoft Windows command-line terminal. Click the **Start** button, and then choose **Run**. Type **cmd** in the Open field, and then click **OK**. Once in the command prompt, navigate to the Nmap installation directory and run Nmap for R1 using **nmap** *address*.

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator> cd ..

C:\Documents and Settings> cd ..

C:\> cd Program Files

C:\Program Files> cd Nmap

C:\Program Files\Nmap> nmap 192.168.10.1

Starting Nmap 4.20 ( http://insecure.org ) at 2007-02-05 03:48 Eastern
Standard
Time
Interesting ports on 192.168.10.1:
Not shown: 1695 closed ports
PORT   STATE SERVICE
23/tcp open  telnet
80/tcp open  http
MAC Address: 00:19:06:23:43:80 (Cisco Systems)

Nmap finished: 1 IP address (1 host up) scanned in 3.843 seconds
```

Step 4: Prepare Router for SDM

Recall that a router must be prepared for SDM connectivity before the SDM management interface can configure a router.

Which protocols does SDM use to connect to a router? Provide the name and number layer of the OSI model at which each protocol operates.

Prepare the router for SDM connectivity with the following configuration.

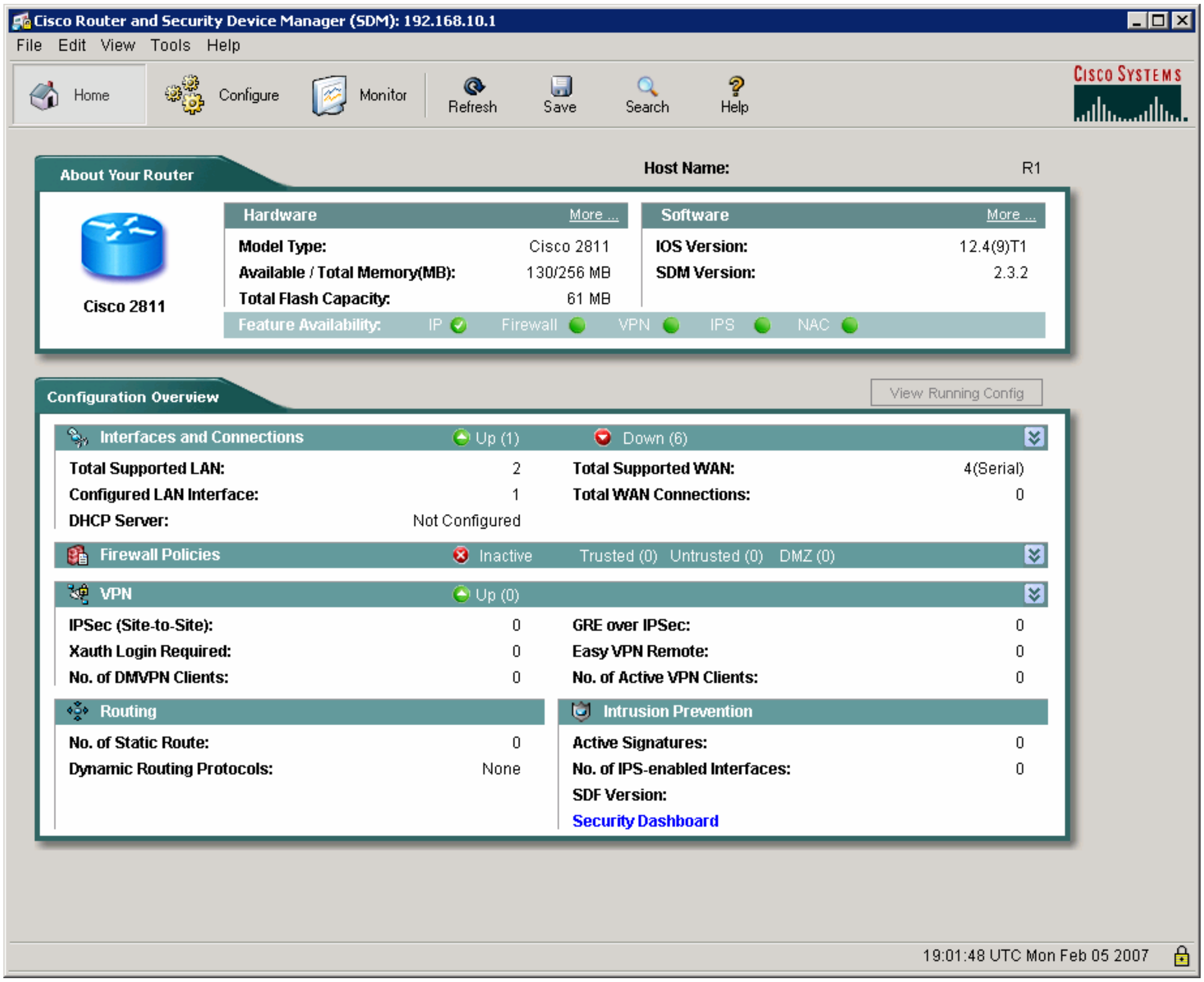
```
username ciscosdm privilege 15 password 7 030752180500324843
ip http authentication local
ip http secure-server
line vty 0 4
 transport input ssh
```

For more information on the commands used here, refer to Lab 3.1: : Configuring SDM on a Router. See Section 2 in this lab: Prepare the Router for SDM

Step 5: Use SDM One-Step Lockdown

One-step lockdown is an automatic procedure used to secure a router. It is only available through the SDM interface and not on the router itself in CLI. Thus, SDM must deliver the one-step lockdown functionality over SSH as an expanded set of configuration commands.

On the host, start up SDM to configure R1. At the menu bar at the top of the window, click **Edit**, and then choose **Preferences**.



Host Name: R1

Hardware

Model Type: Cisco 2811
Available / Total Memory(MB): 130/256 MB
Total Flash Capacity: 61 MB

Software

IOS Version: 12.4(9)T1
SDM Version: 2.3.2

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Interfaces and Connections ● Up (1) ● Down (6)

Total Supported LAN: 2
Configured LAN Interface: 1
DHCP Server: Not Configured
Total Supported WAN: 4(Serial)
Total WAN Connections: 0

Firewall Policies ☒ Inactive Trusted (0) Untrusted (0) DMZ (0)

VPN ● Up (0)

IPSec (Site-to-Site): 0
Xauth Login Required: 0
No. of DMVPN Clients: 0
GRE over IPSec: 0
Easy VPN Remote: 0
No. of Active VPN Clients: 0

Routing

No. of Static Route: 0
Dynamic Routing Protocols: None

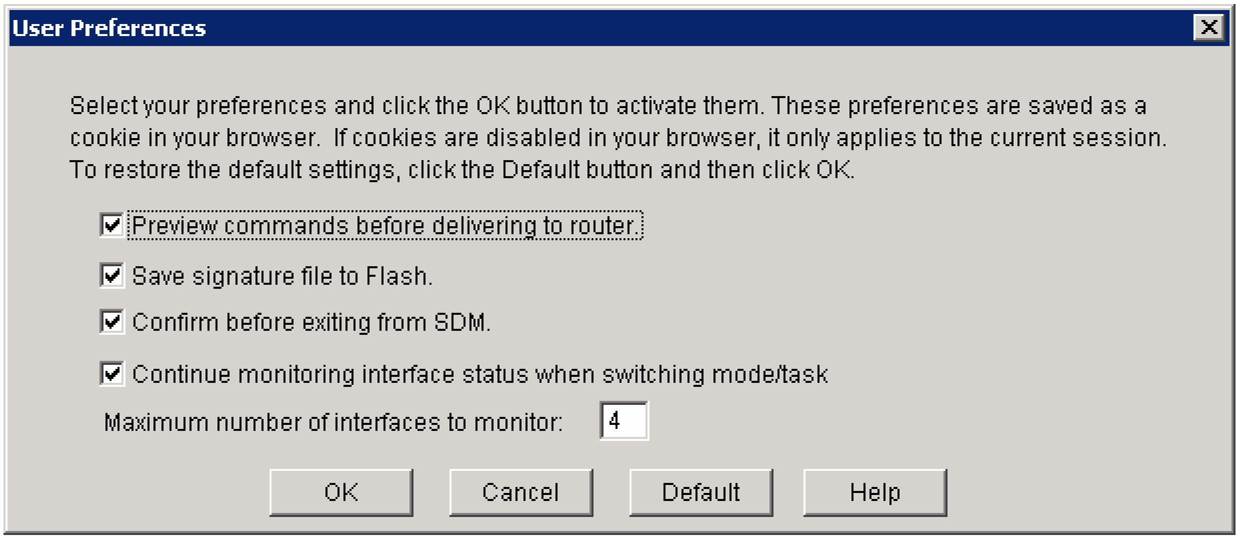
Intrusion Prevention

Active Signatures: 0
No. of IPS-enabled Interfaces: 0
SDF Version:

19:01:48 UTC Mon Feb 05 2007

Figure 5-1: SDM Home Screen

Make sure **Preview commands before delivering to router** is checked, and then click **OK**. From now on, you are able to preview exactly what configuration lines the SDM delivers to the router.



Select your preferences and click the OK button to activate them. These preferences are saved as a cookie in your browser. If cookies are disabled in your browser, it only applies to the current session. To restore the default settings, click the Default button and then click OK.

☑ Preview commands before delivering to router.

☑ Save signature file to Flash.

☑ Confirm before exiting from SDM.

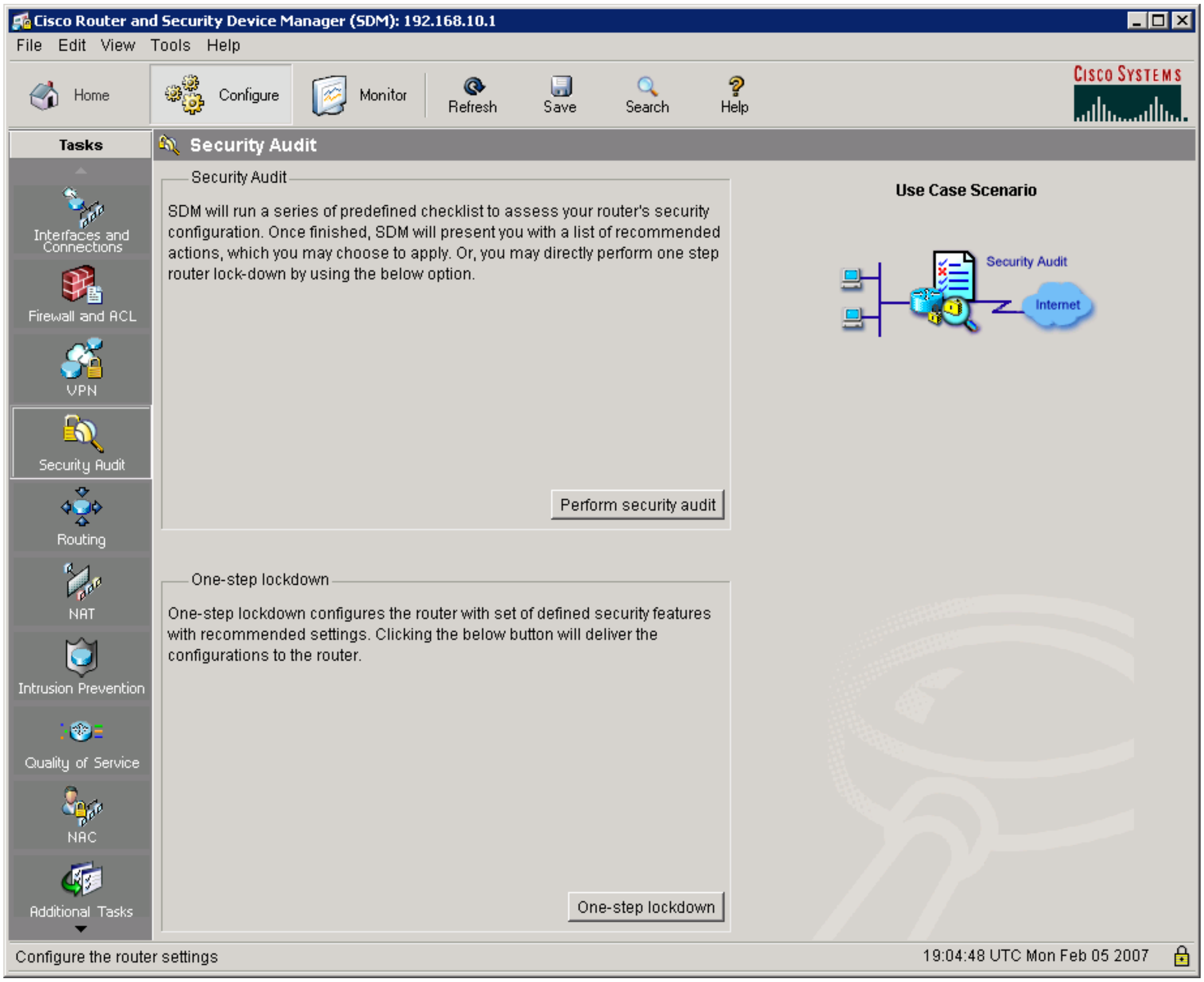
☑ Continue monitoring interface status when switching mode/task

Maximum number of interfaces to monitor: 4

Figure 5-2: SDM User Preferences

On the SDM home screen, click the **Configure** icon. Then on the Tasks toolbar, click the **Security Audit** icon.

Click the **One-step lockdown** button to begin the process.



Security Audit

SDM will run a series of predefined checklist to assess your router's security configuration. Once finished, SDM will present you with a list of recommended actions, which you may choose to apply. Or, you may directly perform one step router lock-down by using the below option.

One-step lockdown

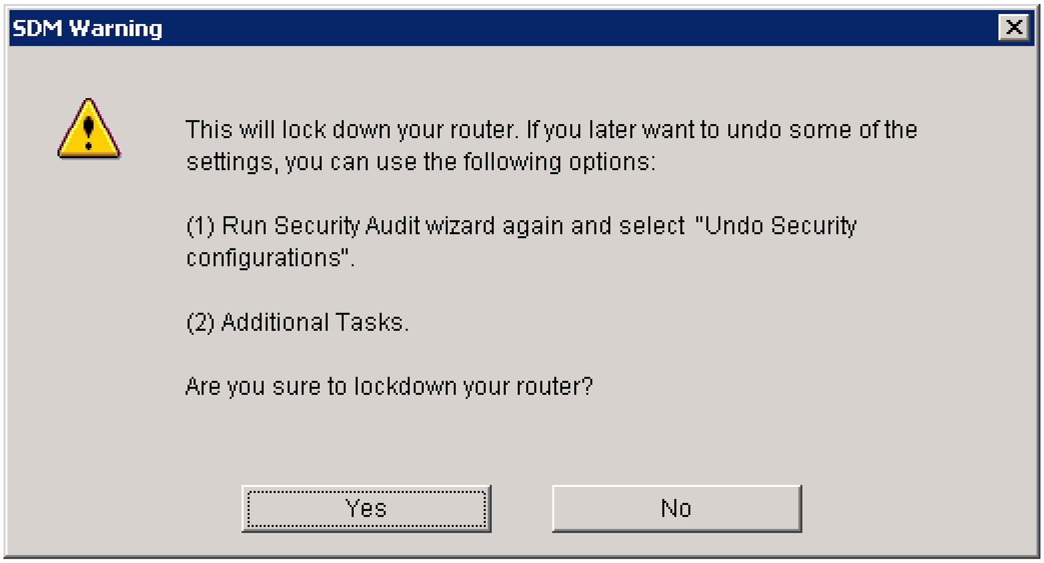
One-step lockdown configures the router with set of defined security features with recommended settings. Clicking the below button will deliver the configurations to the router.

Configure the router settings

19:04:48 UTC Mon Feb 05 2007

Figure 5-3: SDM Security Audit Screen

Click **Yes** to begin.



This will lock down your router. If you later want to undo some of the settings, you can use the following options:

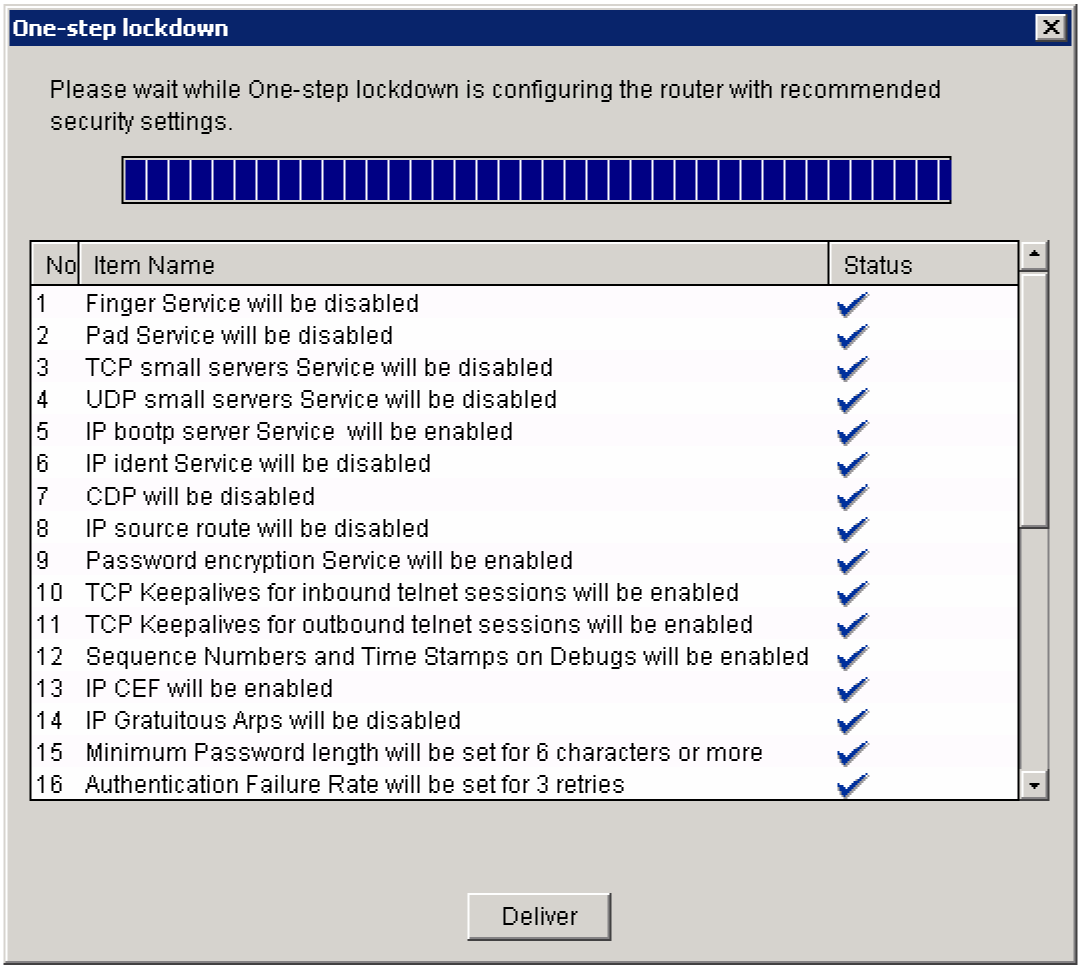
(1) Run Security Audit wizard again and select "Undo Security configurations".

(2) Additional Tasks.

Are you sure to lockdown your router?

Figure 5-4: SDM One-step Lockdown Warning

After analysis, SDM generates a list of changes to be made to your router's configuration. Click the **Deliver** button to have SDM add them to the router.

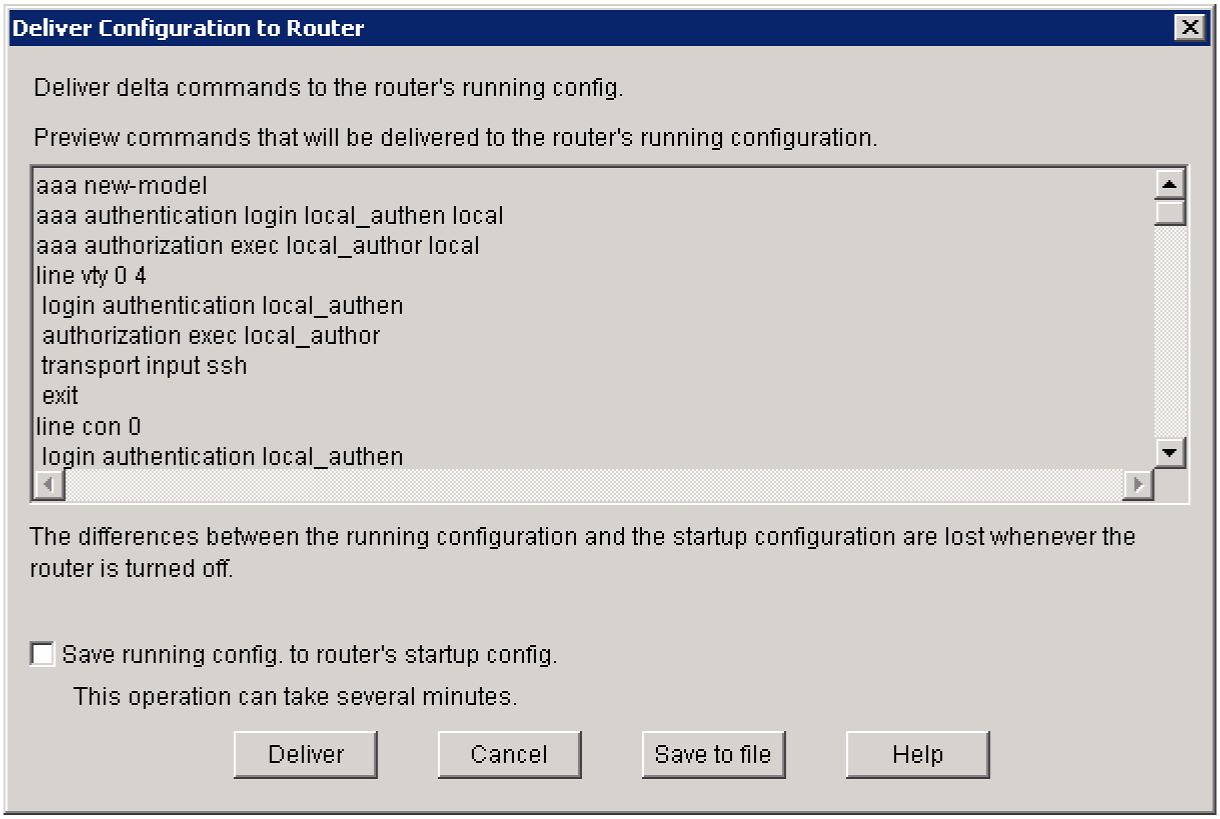


Please wait while One-step lockdown is configuring the router with recommended security settings.

No	Item Name	Status
1	Finger Service will be disabled	✓
2	Pad Service will be disabled	✓
3	TCP small servers Service will be disabled	✓
4	UDP small servers Service will be disabled	✓
5	IP bootp server Service will be enabled	✓
6	IP ident Service will be disabled	✓
7	CDP will be disabled	✓
8	IP source route will be disabled	✓
9	Password encryption Service will be enabled	✓
10	TCP Keepalives for inbound telnet sessions will be enabled	✓
11	TCP Keepalives for outbound telnet sessions will be enabled	✓
12	Sequence Numbers and Time Stamps on Debugs will be enabled	✓
13	IP CEF will be enabled	✓
14	IP Gratuitous Arps will be disabled	✓
15	Minimum Password length will be set for 6 characters or more	✓
16	Authentication Failure Rate will be set for 3 retries	✓

Figure 5-5: One-step Lockdown Configuration Tasks

Because you enabled the display of commands to be delivered to the router at the outset of this step, a preview window appears showing you the configuration changes SDM one-step lockdown intends to make. Examine the commands here or in the final configuration to see what security changes SDM will make. Click **Deliver** again when you are done reviewing.



Deliver delta commands to the router's running config.

Preview commands that will be delivered to the router's running configuration.

```
aaa new-model
aaa authentication login local_authen local
aaa authorization exec local_author local
line vty 0 4
 login authentication local_authen
 authorization exec local_author
 transport input ssh
 exit
line con 0
 login authentication local_authen
```

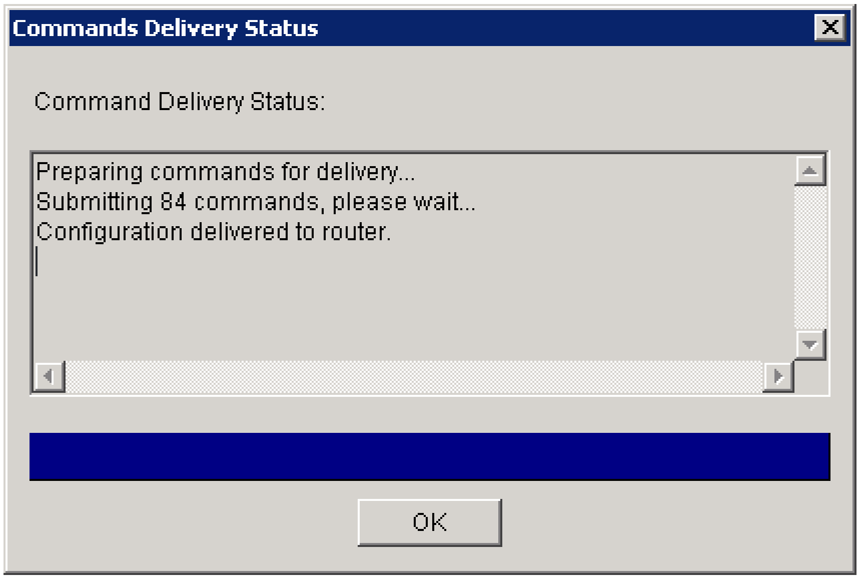
The differences between the running configuration and the startup configuration are lost whenever the router is turned off.

□ Save running config. to router's startup config.

This operation can take several minutes.

Figure 5-6: One-Step Lockdown Command Preview

Click **OK** once the commands are delivered.



Command Delivery Status:

```
Preparing commands for delivery...
Submitting 84 commands, please wait...
Configuration delivered to router.
```

Figure 5-7: SDM Command Delivery Status

Close SDM when the operation has completed.

Step 6: Use Nmap to See Changes

Run Nmap again to see that the router closed the telnet port (because of one-step lockdown) but can still be accessed through SSH.

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator> cd ..

C:\Documents and Settings> cd ..

C:\> cd Program Files

C:\Program Files> cd Nmap

C:\Program Files\Nmap> nmap 192.168.10.1

Starting Nmap 4.20 ( http://insecure.org ) at 2007-02-05 13:58 Eastern
Standard
Time
Interesting ports on 192.168.10.1:
Not shown: 1694 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
MAC Address: 00:19:06:23:43:80 (Cisco Systems)

Nmap finished: 1 IP address (1 host up) scanned in 6.047 seconds
```

For what reason does one-step lockdown disable access via the Telnet protocol?

Conclusion

Identify and explain the logic of one configuration change in each of the following categories:

1. System services

2. Authentication options

3. Secure protocols (HTTPS, SSH, and so forth.)

4. Terminal access

5. Interface protocols

6. Logging

If you are unfamiliar with particular protocols or services, use the product documentation available online at http://www.cisco.com/univercd/.

Final Configurations

```
R1# show run
no service pad
service tcp-keepalives-in
service tcp-keepalives-out
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
service sequence-numbers
!
hostname R1
!
```

```
security authentication failure rate 3 log
security passwords min-length 6
logging buffered 51200 debugging
logging console critical
!
aaa new-model
!
aaa authentication login local_authen local
aaa authorization exec local_author local
!
aaa session-id common
!
no ip source-route
ip tcp synwait-time 10
!
no ip bootp server
ip ssh time-out 60
ip ssh authentication-retries 2
!
crypto pki trustpoint TP-self-signed-1455051929
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1455051929
 revocation-check none
 rsakeypair TP-self-signed-1455051929
!
crypto pki certificate chain TP-self-signed-1455051929
 certificate self-signed 01
  3082023A 308201A3 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 31343535 30353139 3239301E 170D3037 30323035 31393030
  30395A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 34353530
  35313932 3930819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100C891 CD55482C 635B8206 52D2DD2E A7259989 EA7BAF48 E39F84DF A057CD84
  5294DE11 C5255AEA 9BD19262 0F9FD62F 692ACD8B 605D0B37 3ACA9BD7 581BD0DD
  006E5F36 5E55C5A3 FC5BFF9F AF7CD7E9 577F83A3 A496E4B3 6EA72B40 F29A6597
  50F46713 E43BF3D5 436F7E2D 9CBBC7ED 813AD448 73C358C0 E4B8059D 346418A0
  83AF0203 010001A3 62306030 0F060355 1D130101 FF040530 030101FF 300D0603
  551D1104 06300482 02523130 1F060355 1D230418 30168014 26532DF5 F2533C37
  09E52626 45CF92F0 3DB592A2 301D0603 551D0E04 16041426 532DF5F2 533C3709
  E5262645 CF92F03D B592A230 0D06092A 864886F7 0D010104 05000381 810033C2
  C04198B4 7DD7905C F750F7C2 58278CDB E601DE3E DF8A2A1E 8E89A9E5 A688AD9A
  AC7C718A 9FF34CE9 FA536240 CC502BA6 4D5C9D62 951451DD 008910D0 1DEA4047
  236EC3A9 CC10DA91 22F46C47 2518C510 D7F4B983 AA8B1162 ED841F91 DB238E68
  93792098 045326BE 68AB3C82 EC8AE642 A7456B3A AE7F8182 34E13367 3965
  quit
username ciscosdm privilege 15 password 7 030752180500324843
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip route-cache flow
 no mop enabled
 no shutdown
!
ip http server
ip http authentication local
ip http secure-server
!
logging trap debugging
no cdp run
```

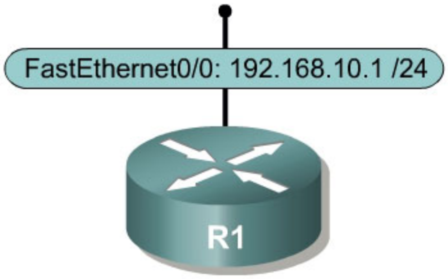
```
!
banner login ^CAuthorized access only!
 Disconnect IMMEDIATELY if you are not an authorized user!
^C
!
line con 0
 login authentication local_authen
line aux 0
 login authentication local_authen
line vty 0 4
 authorization exec local_author
 login authentication local_authen
 transport input ssh
end
```

Lab 5.2 Securing a Router with Cisco AutoSecure

Learning Objectives

- Implement Cisco AutoSecure on a router

Topology Diagram



Scenario

In this scenario, you will configure Cisco AutoSecure on a router. AutoSecure is a built-in tool in the Cisco IOS that uses a predefined set of commands for securing the router based on questions answered by the network administrator. The AutoSecure command set resembles some of the commands implemented by Cisco Service Device Manager (SDM) one-step lockdown in Lab 5.1.

Step 1: Configure the Physical Interface

Configure the R1 physical interface using the IP address shown in the topology diagram. You can force the interface into an “always up” state using the interface-level **no keepalive** command. Then use the **no shutdown** command to bring the interface up. Because you disabled keepalives, the interface status will display as link state (Layer 1) and line protocols state (Layer 2) “up,” even if it is not connected to an external device.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no keepalive
R1(config-if)# no shutdown
```

Normally, you would not use the **no keepalive** command on a routed interface.

Step 2: Configure AutoSecure

At the privileged EXEC prompt, issue the **auto secure** command to start AutoSecure. You may notice that this command is hidden from the Cisco IOS in-line help system. It is hidden because AutoSecure can lock out network administrators if executed by a user who has gained unauthorized access to a router. AutoSecure is a command-line wizard that enables a set of features in the router’s configuration. Its function is similar to SDM one-step lockdown; however, its functionality is text-based and more interactive.

```
R1# auto secure
                --- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security of
the router, but it will not make it absolutely resistant
to all security attacks ***

AutoSecure will modify the configuration of your device.
All configuration changes will be shown. For a detailed
explanation of how the configuration changes enhance security
and any possible side effects, please refer to Cisco.com for
Autosecure documentation.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for AutoSecure
```

Use the default of **no** to answer the question “Is this router connected to internet?” To enter in a default (the value in the square brackets), press Enter on your keyboard. After Enter is pressed, AutoSecure executes some prepackaged security precaution commands. These are commands that disable services that are typically not needed. AutoSecure also enables several security features.

```
Is this router connected to internet? [no]: no

Securing Management plane services...

Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol

Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp
```

What is the function of each of the following system services and IP servers?

1. Finger

2. TCP small servers

3. UDP small servers

4. Password encryption
5. TCP keepalives

6. CDP

7. BOOTP Server

8. HTTP Server

9. Gratuitous ARP

The following prompt appears, requesting that you create a security banner:

```
Here is a sample Security Banner to be shown
at every access to device. Modify it to suit your
enterprise requirements.

Authorized Access only
  This system is the property of So-&-So-Enterprise.
  UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.
  You must have explicit permission to access this
  device. All activities performed on this device
  are logged. Any violations of access policy will result
  in disciplinary action.

Enter the security banner {Put the banner between
k and k, where k is any character}:
~CCNP Router
UNAUTHORIZED ACCESS PROHIBITED~
```

What should your security banner emphasize and why?

To create a security banner, you need to enter a delimiting character, followed by your message, followed by the delimiting character. The character must be a character that your message will not contain. Once this character is found again in the field, the message context will terminate. In the example below, the tilde character (~) is used as the delimiter.

If you have not previously configured enable passwords and enable secrets, or if both the enable pass and the enable secret password are the same, AutoSecure will force you to create them. AutoSecure also enforces a 6-character minimum length on passwords, so create them based on that requirement. This lab will use “password” for the enable password and “secret” for the enable secret to meet the minimum length practices.

```
Enable secret is either not configured or
 is the same as enable password
Enter the new enable secret: secret
Confirm the enable secret : secret
Enter the new enable password: password
Confirm the enable password: password
```

Create a new user in the local user database, because AutoSecure enables AAA and uses local authentication. Use a username and password of “ciscouser”.

```
Configuration of local user database
Enter the username: ciscouser
Enter the password: ciscouser
Confirm the password: ciscouser
Configuring AAA local authentication
Configuring Console, Aux and VTY lines for
local authentication, exec-timeout, and transport
```

The router will also enable some login enhancements, which it will need some parameters for. Use a blocking period of 10 seconds, a maximum failure number of 5, and a maximum time period for crossing failed login attempts of 10.

```
Securing device against Login Attacks
Configure the following parameters

Blocking Period when Login Attack detected: 10

Maximum Login failures with the device: 5

Maximum time period for crossing the failed login attempts: 10
```

The router will configure a Secure Shell (SSH) server, which will require a domain name. Use “cisco.com” as the domain name.

```
Configure SSH server? [yes]: yes
Enter the domain-name: cisco.com
```

Why does AutoSecure enable Secure Shell?

AutoSecure disables some unneeded or potentially vulnerable services on each physical interface. You are prompted to enable Context-Based Access Control and TCP intercept. For this lab, type **no** to not configure these services.

```
Configuring interface specific AutoSecure services
Disabling the following ip services on all interfaces:

 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
Disabling mop on Ethernet interfaces

Securing Forwarding plane services...

Enabling CEF (This might impact the memory requirements for your platform)
Enabling unicast rpf on all interfaces connected
to internet

Configure CBAC Firewall feature? [yes/no]: no
Tcp intercept feature is used prevent tcp syn attack
on the servers in the network. Create autosec_tcp_intercept_list
to form the list of servers to which the tcp traffic is to
be observed


Enable tcp intercept feature? [yes/no]: no
```

From your reading, what function does “enabling unicast rpf on all interfaces connected to the internet” serve?

The last step AutoSecure does is verify the configuration that it is going to add. After AutoSecure shows you the running configuration it has generated, AutoSecure asks you to verify that you want to apply the running configuration. Use the default of **yes**.

```
This is the configuration generated:

no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
service password-encryption
service tcp-keepalives-in
service tcp-keepalives-out
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd
banner motd ^CCCNP Router
UNAUTHORIZED ACCESS PROHIBITED^C
security passwords min-length 6
security authentication failure rate 10 log
enable secret 5 $1$d7wX$kb5JYyFOQmSRWVpW8iitA.
enable password 7 095C4F1A0A1218000F
username ciscouser password 7 02050D4808091A32495C
aaa new-model
aaa authentication login local_auth local
line con 0
 login authentication local_auth
 exec-timeout 5 0
 transport output telnet
line aux 0
 login authentication local_auth
 exec-timeout 10 0
 transport output telnet
line vty 0 4
 login authentication local_auth
 transport input telnet
line tty 1
 login authentication local_auth
 exec-timeout 15 0
login block-for 10 attempts 5 within 10
ip domain-name cisco.com
crypto key generate rsa general-keys modulus 1024
ip ssh time-out 60
ip ssh authentication-retries 2
line vty 0 4
 transport input ssh telnet
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
logging facility local2
logging trap debugging
service sequence-numbers
logging console critical
logging buffered
interface FastEthernet0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface FastEthernet0/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface Serial0/0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
interface Serial0/0/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
interface Serial0/1/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
interface Serial0/1/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
ip cef
access-list 100 permit udp any any eq bootpc
!
end
```

When the router asks you to accept this configuration so it can be applied to the router, answer **yes**.

```
Apply this configuration to running-config? [yes]: yes

Applying the config generated to running-config
The name for the keys will be: R1.cisco.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

*Feb  6 01:03:52.694: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Feb  6 01:03:57.250 UTC: %AUTOSEC-1-MODIFIED: AutoSecure configuration has
been Modified on this device
```

How does the router generate the name for the public crypto keys shown in the preceding configuration text?

Final Configuration

```
R1# show run
no service pad
service tcp-keepalives-in
service tcp-keepalives-out
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
service sequence-numbers
!
hostname R1
!
security authentication failure rate 10 log
security passwords min-length 6
logging buffered 4096 debugging
logging console critical
```

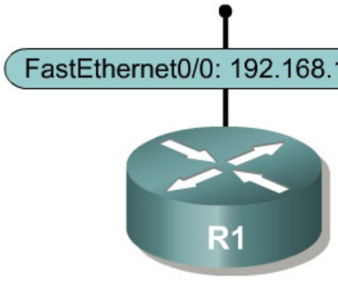
```
enable secret 5 $1$d7wX$kb5JYyFOQmSRWVpW8iitA.
enable password 7 095C4F1A0A1218000F
!
aaa new-model
!
aaa authentication login local_auth local
!
no ip source-route
no ip gratuitous-arps
!
ip cef
!
no ip bootp server
ip domain name cisco.com
ip ssh time-out 60
ip ssh authentication-retries 2
login block-for 10 attempts 5 within 10
!
username ciscouser password 7 02050D4808091A32495C
archive
 log config
  logging enable
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no keepalive
 no mop enabled
 no shutdown
!
no ip http server
no ip http secure-server
!
logging trap debugging
logging facility local2
access-list 100 permit udp any any eq bootpc
no cdp run
!
banner motd ^CCCNP Router
UNAUTHORIZED ACCESS PROHIBITED^C
!
line con 0
 exec-timeout 5 0
 login authentication local_auth
 transport output telnet
line aux 0
 exec-timeout 15 0
 login authentication local_auth
 transport output telnet
line vty 0 4
 login authentication local_auth
 transport input telnet ssh
end
```

Lab 5.3 Disabling Unneeded Services

Learning Objectives

- Identify and disable unneeded and insecure services on a router
- Enable TCP keepalives

Topology Diagram



Scenario

In this lab, you will disable unneeded services on a router. SDM one-step lockdown or AutoSecure will disable many of the same services. This lab only requires one router.

Step 1: Configure the Physical Interface

Because this lab uses only one router, you will simulate an active FastEthernet connection by activating the interface and applying the **no keepalive** command to initiate an “always up” state, regardless of the existence of a device at the remote end. Normally, you would not use the **no keepalive** command on a routed interface, except in special circumstances. In this lab, you will use it only for simulation purposes.

Configure the R1 physical interface using the IP address shown in the topology diagram. Use the **no keepalive** command in interface configuration mode, and then use the **no shutdown** command to activate the interface. Because you have disabled keepalives, the interface status displays as link state (Layer 1) and line protocols state (Layer 2) “up,” even if it is not connected to an external device.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no keepalive
R1(config-if)# no shutdown
```

Step 2: Ensure Services Are Disabled

Some services are disabled by default on more recent Cisco IOS releases, so you do not necessarily have to disable them. However, it is helpful to know the commands in case they are enabled and affect security. These commands are especially useful if you have older versions of the IOS that you are using.

The **no ip finger** command replaces the **no service finger** command. Both disable the finger service, which allows remote users or systems to identify users connected to the local router's terminal lines or who have active running processes.

```
R1(config)# no ip finger
```

How could the enabling of the finger service pose a security risk?

The **no service udp-small-servers** and **no service tcp-small-servers** commands disable UDP and TCP small servers, such as echo and discard. The small servers are not needed in most environments.

```
R1(config)# no service udp-small-servers
R1(config)# no service tcp-small-servers
```

The TCP and UDP small servers are enabled by default on Cisco IOS Release 11.2 and earlier. They are disabled by default on Cisco IOS Release 11.3 and later.

It is recommended that you do not enable these services, unless it is absolutely necessary. These services could be exploited indirectly to gain information about the target system, or exploited directly with a fraggle attack, which uses UDP echo.[1] Also, if a sender transmits a volume of fake requests for UDP diagnostic services on the router, the requests could consume all CPU resources.

Step 3: Manage Router Access

Name two popular TCP protocols that network administrators use to manage to network devices.

[1] http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080149ad6.shtml#topic5

Recall that such management applications as telnet and SSH connect to the vty port on a router. A Cisco router has five vtys configured by default, numbered 0 through 4, to which users connect to access the command-line interface. When one vty is in use, the next vty port is used. If all vty ports are being used, other users cannot connect to the device in this way.

Describe how individuals with malicious intent could exploit vty port functionality.

Describe at least two ways to prevent these type of attacks from occurring.

Enabling TCP keep alives causes the router to generate periodic keep alive messages, letting it detect and drop broken Telnet connections. This frees up hung telnet sessions. This functionality also has the additional benefit of making the router more secure by preventing a hacker from exploiting a hung telnet session. To enable TCP keepalive packets on idle connections, use the **service tcp-keepalives-in** and **service tcp-keepalives-out** commands in global configuration mode.

```
R1(config)# service tcp-keepalives-in
R1(config)# service tcp-keepalives-out
```

Step 4: Disable CDP

Cisco Discovery Protocol (CDP) is a great troubleshooting tool, especially on poorly documented networks. However, it can also leave your network susceptible to reconnaissance attacks.

CDP is used for some network management functions, but is dangerous because it allows any system on a directly connected segment to learn that the router is a Cisco device, and to determine the model number and the Cisco IOS software version being run. This reconnaissance information can be used to design attacks against the router.

To disable the CDP service globally, use the **no cdp run** command in global configuration mode. To disable CDP on a per-interface basis, issue the **no cdp enable** command in interface configuration mode.

```
R1(config)# no cdp run
```

Step 5: Disable Other Unused Services

Disable the packet assembler/disassembler (PAD) on the router by using the **no service pad** command in global configuration mode. PAD translates between packets and character streams in legacy networks. You should not need this service in most current IP networks.

```
R1(config)# no service pad
```

The BOOTP service is used in networks that have a centralized Cisco IOS software deployment: One router can be used by other routers to load its operating system. However, the BOOTP service is seldom used, and it gives a hacker an opportunity to steal a Cisco IOS image. Therefore, in most situations, you should disable it using the following command:

```
R1(config)# no ip bootp server
```

The most recent Cisco IOS software releases issue the Hypertext Transfer Protocol (HTTP) to support remote configuration and monitoring. In general, HTTP access is equivalent to interactive access to the router. The authentication protocol used for HTTP is equivalent to sending a clear-text password across the network. Unfortunately, there is no effective provision in HTTP for challenge-based or one-time passwords. This makes HTTP a relatively risky choice for router management across the public Internet.

If you choose to use HTTP for router management, use the **ip http access-class** command to restrict access to IP addresses. You should also use the **ip http authentication** *method* command to configure authentication. As with interactive logins, the best choice for HTTP authentication is to issue a TACACS+ or RADIUS server.[2]

In the following example, you choose not to use the IOS web interface. Disable the Cisco IOS HTTP server with the **no ip http server** command in global configuration mode.

```
R1(config)# no ip http server
```

The IP protocol supports source routing options that allow the sender of an IP datagram to control the route that a datagram takes toward its ultimate destination, and generally the route that any reply takes. These options are rarely used for legitimate purposes in real networks. Some older IP

[2] http://www.cisco.com/warp/public/707/21.html#http

implementations do not process source-routed packets properly, and it is possible to send them datagrams with source routing options in order to crash machines that run these implementations. The **no ip source-route** command discards packets that contain source routing information. You can disable this if a network is not using source-routing information.

```
R1(config)# no ip source-route
```

Gratuitous Address Resolution Protocols (ARPs) are unsolicited ARP requests and replies that can be generated for several reasons, such as when detecting IP address conflicts or updating ARP tables after an address change. However, attackers can use these packets to spoof a valid network device; for example, an attacker could send out a packet that claims to be the default router. If you choose to do so, you can disable Gratuitous ARP with the global configuration command **no ip gratuitous-arps**.

```
R1(config)# no ip gratuitous-arps
```

Step 6: Disabling Unneeded Interface Services

Some commands are used on a per-interface basis to mitigate certain types of hacker attacks or reconnaissance. Issue the following commands to the R1 FastEthernet0/0 interface.

The **no ip redirects** command disables IP redirects so that the router does not send out ICMP redirect messages. These messages occur when a router routes a packet out the interface that it came in on. The contents of the message tells the packet sender to send it directly to where the router would have sent it.

```
R1(config)# interface fastethernet0/0
R1(config-if)# no ip redirects
```

The **no ip proxy-arp** command disables proxy ARPs from the router, which means that a router can respond to an ARP request for an address on a remote subnet (with its own MAC address) and take responsibility for the packets getting to their destination.

```
R1(config-if)# no ip proxy-arp
```

The **no ip unreachables** command prevents the router from sending Internet Control Message Protocol (ICMP) unreachable messages when it has not learned a route to a destination. Normally, these are helpful for troubleshooting, but they can also be involved in reconnaissance or DoS attacks.

```
R1(config-if)# no ip unreachables
```

Similarly, directed broadcasts can be used in reconnaissance and DoS attacks. You can prevent this by using the **no ip directed-broadcast** command. Although directed broadcasts are disabled by default in recent Cisco IOS releases, this command is included in the lab because it is a significant security point. This command makes the router discard packets with a destination address that is the broadcast address for a specific network. This packet can be used in a DoS attack. If there is a hacker located at 192.168.1.1 who wants to attack a host at 192.168.2.2, the hacker can ping 192.168.3.255, which is sourced from 192.168.2.2. Every host in the 192.168.3.0 /24 subnet will respond to that ICMP echo request and direct the response to the spoofed source. In the given case, the spoofed source is the victim of the attack.

```
R1(config-if)# no ip directed-broadcast
```

Unnecessary ICMP messages can be sent in response to ICMP mask request messages. Use the **no ip mask-reply** command to disable ICMP mask reply messages.

```
R1(config-if)# no ip mask-reply
```

Maintenance Operation Protocol (MOP) is an old DECnet protocol that is not needed on most current IP networks. To disable it, issue the **no mop enable** command. This is enabled by default on Ethernet interfaces only.

```
R1(config-if)# no mop enabled
```

Final Configuration

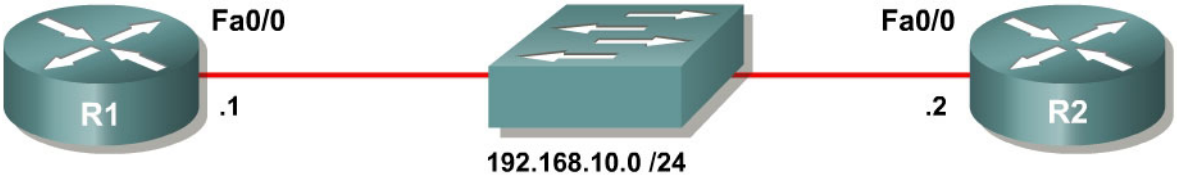
```
R1#show run
service tcp-keepalives-in
service tcp-keepalives-out
!
hostname R1
!
no ip source-route
no ip gratuitous-arps
!
no ip bootp server
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no keepalive
 no mop enabled
 no shutdown
!
no ip http server
!
no cdp run
end
```

Lab 5.4 Enhancing Router Security

Learning Objectives

- Implement Cisco IOS login enhancements
- Enforce a minimum password length
- Modify command privilege levels
- Configure a banner
- Configure a router to use SSH
- Enable password encryption

Topology Diagram



Scenario

In previous labs in this module, you have automatically configured numerous security features on routers using One-Step Lockdown and AutoSecure. This lab provides details on manual security configuration.

In this scenario, the CIO of your corporation has a new security policy that must be rolled out across the network. The policy has strict rules regarding security banners, privilege levels, login security, password lengths, and password storage. Additionally, you must only allow secure shell (SSH) access to the VTY lines on routers in the network.

In this lab, you will configure your routers to conform to the following security policies:

- Enforce a minimum password length of eight characters.

- Display a security banner stating that unauthorized use is prohibited and prosecutable and that the use of this device is monitored and will be used as evidence.

- Permit only secure management methods. No management passwords must be sent or stored as clear text. Anti-replay measures must be taken.

- Login procedures must be guarded against denial of service (DoS) attacks.

R2 acts as a host whose only function is to access R1. You will do most of the configuration on R1.

Step 1: Configure the Physical Interfaces

Configure the R1 and R2 physical interfaces using the IP address shown in the topology diagram, and activate the interfaces with the **no shutdown** command.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown

R2(config)# interface fastethernet0/0
R2(config-if)# ip address 192.168.10.2 255.255.255.0
R2(config-if)# no shutdown
```

If you use a switch to connect the Fast Ethernet interfaces on your routers, erase the configuration in non-volatile RAM with the **erase start** command and reload the switch. All ports default to VLAN 1, which provides connectivity between the Fast Ethernet interfaces on R1 and R2

Step 2: Telnet to R1

On R2, issue the **telnet** *host* command from an exec prompt, where *host* is the R1 IP address.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open


Password required, but none set

[Connection to 192.168.10.1 closed by foreign host]
R2#
```

In troubleshooting, what does the above output help you determine?

Notice that R1 accepts the Telnet connection, but then terminates the connection because no line password is set. This can be easily fixed on R1 by setting up line authentication. For this lab, use the local username database. Another option is to configure communication to a RADIUS or TACACS+ server for AAA services.

Create a user account on the router with the username and password of “cisco” by using the **username** *name* **password** *password* command in global configuration mode.

```
R1(config)# username cisco password cisco
```

To configure the virtual teletype terminal (VTY) lines, enter the **line vty** *low* [*high*] command in global configuration mode. The *low* and *high* keywords refer to the bounds of the lines you are configuring. On a router, you configure the five terminal lines simultaneously in most scenarios, so the low number would be 0—the first line—and the high number would be 4—the last line. At the line configuration prompt, enter the **login local** command to enable local authentication.

```
R1(config)# line vty 0 4
R1(config-line)# login local
```

Now, connect to R1 from R2.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open


User Access Verification

Username: cisco
Password: cisco
```

This time the router allows your Telnet connection. Attempt to enter privileged exec mode on R1 through the Telnet session.

```
R1> enable
% No password set
R1> exit

[Connection to 192.168.10.1 closed by foreign host]
R2#
```

To fix the closed connection, configure an enable password with the **enable secret** *password* command on R1. Make the enable password “cisco”.

```
R1(config)# enable secret cisco
```

From R2, connect again to R1 via Telnet.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open


User Access Verification

Username: cisco
Password: cisco
R1> enable
Password: cisco
R1# exit
```

```
[Connection to 192.168.10.1 closed by foreign host]
R2#
```

Why does Cisco IOS not allow Telnet connections or entering privileged mode remotely unless the proper passwords are set up?

Step 3: Configure Cisco IOS Login Enhancements

What happens when you fail Telnet authentication on R1 repeatedly?

Test your answer by attempting to telnet from R2 to R1 using incorrect usernames and/or passwords.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open


User Access Verification

Username: cisco
Password: wrongpass
% Login invalid

Username: cisco
Password: guesspass
% Login invalid

Username: baduser
Password: badpass
% Login invalid

[Connection to 192.168.10.1 closed by foreign host]
R2#
```

The router disconnects the Telnet session after three failed tries, but does not employ any further security beyond this. This router is still vulnerable to a dictionary attack—a hacker running a program that continually tries new passwords.

Fortunately, Cisco recently enhanced login security features to mitigate this type of attack. To view which login features are configured, use the **show login** command. Each of the following output lines corresponds to one of the new login features.

```
R1# show login
     No login delay has been applied.
     No Quiet-Mode access list has been configured.

     Router NOT enabled to watch for login Attacks
```

The **login block-for** *seconds* **attempts** *tries* **within** *seconds* command, issued in global configuration mode, allows the activation of all the other login security features. The first *seconds* parameter specifies the amount of time that login is not allowed after a violation occurs. The *tries* parameter represents the number of failed login attempts permitted during a configurable amount of time. The final *seconds* argument represents the time period over which the number of failed attempts must occur for the user to be locked out of the device. Configure R1 using the following, which blocks all login attempts for 30 seconds if there are two failed login attempts within a 15 second time period.

```
R1(config)# login block-for 30 attempts 2 within 15
```

Test this configuration by attempting to telnet to R1 from R2. Intentionally fail authentication twice and when kicked off, attempt to telnet again. Look at the messages logged on R1 and their timestamps.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open


User Access Verification

Username: test
Password: test
% Login invalid

Username: guess
Password: guess
% Login invalid

[Connection to 192.168.10.1 closed by foreign host]
R2# telnet 192.168.10.1
Trying 192.168.10.1 ...
% Connection refused by remote host

R2#

R1#
*Feb  7 08:27:01.259: %SEC_LOGIN-1-QUIET_MODE_ON: Still timeleft for watching
failures is 11 secs, [user: guess] [Source: 192.168.10.2] [localport: 23]
[Reason: Login Authentication Failed - BadUser] [ACL: sl_def_acl] at 08:27:01
UTC Wed Feb 7 2007
*Feb  7 08:27:31.259: %SEC_LOGIN-5-QUIET_MODE_OFF: Quiet Mode is OFF, because
block period timed out at 08:27:31 UTC Wed Feb 7 2007
R1#
```

When the router reaches the login limit specified by the **login block-for** command, it enters Quiet Mode. Normally, access attempts are blocked during

Quiet Mode, as seen in the output above. However, you can set up an access list that permits trusted hosts to access the router, even in Quiet Mode. Issue the **login quiet-mode access-class** *acl* command in global configuration mode on R1. The access list specifies the source of trusted connections. The following allows hosts on network 192.168.20.0/24 to access the terminal lines (VTY) on R1.

```
R1(config)# login quiet-mode access-class 1
R1(config)# access-list 1 permit 192.168.20.0 0.0.0.255
```

You can also configure the delay between login attempts. The default is 1 second when **login block-for** is enabled. Change the delay to 3 seconds using the **login delay** *seconds* command in global configuration mode. Also, issue the **login on-failure log** command to have the router log failures.

```
R1(config)# login delay 3
R1(config)# login on-failure log
```

Verify the configuration by failing login to R1. Notice the delay between attempts on R2. Verify the login configuration with the **show login** command on R1.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open


User Access Verification

Username: test
Password: test
% Login invalid

Username: this
Password: isatest
% Login invalid

[Connection to 192.168.10.1 closed by foreign host]
R2#

R1# show login
     A login delay of 3 seconds is applied.
     Quiet-Mode access list 1 is applied.
     All failed login is logged.

     Router enabled to watch for login Attacks.
     If more than 2 login failures occur in 15 seconds or less,
     logins will be disabled for 30 seconds.

     Router presently in Normal-Mode.
     Current Watch Window
         Time remaining: 2 seconds.
         Login failures for current window: 0.
     Total login failures: 10.

R1#
*Feb  7 08:43:28.239: %SEC_LOGIN-4-LOGIN_FAILED: Login failed [user: test]
[Source: 192.168.10.2] [localport: 23] [Reason: Login Authentication Failed -
BadUser] at 08:43:28 UTC Wed Feb 7 2007
```

```
*Feb  7 08:43:34.475: %SEC_LOGIN-4-LOGIN_FAILED: Login failed [user: this]
[Source: 192.168.10.2] [localport: 23] [Reason: Login Authentication Failed -
BadUser] at 08:43:34 UTC Wed Feb 7 2007

*Feb  7 08:43:34.475: %SEC_LOGIN-1-QUIET_MODE_ON: Still timeleft for watching
failures is 1 secs, [user: this] [Source: 192.168.10.2] [localport: 23]
[Reason: Login Authentication Failed - BadUser] [ACL: 1] at 08:43:34 UTC Wed
Feb 7 2007

R1# show login
     A login delay of 3 seconds is applied.
     Quiet-Mode access list 1 is applied.
     All failed login is logged.

     Router enabled to watch for login Attacks.
     If more than 2 login failures occur in 15 seconds or less,
     logins will be disabled for 30 seconds.

     Router presently in Quiet-Mode.
     Will remain in Quiet-Mode for 26 seconds.
     Restricted logins filtered by applied ACL 1.

R1#

*Feb  7 08:44:04.475: %SEC_LOGIN-5-QUIET_MODE_OFF: Quiet Mode is OFF, because
block period timed out at 08:44:04 UTC Wed Feb 7 2007

R1# show login
     A login delay of 3 seconds is applied.
     Quiet-Mode access list 1 is applied.
     All failed login is logged.

     Router enabled to watch for login Attacks.
     If more than 2 login failures occur in 15 seconds or less,
     logins will be disabled for 30 seconds.

     Router presently in Normal-Mode.
     Current Watch Window
         Time remaining: 11 seconds.
         Login failures for current window: 0.
     Total login failures: 12.
```

Step 4: Enforce a Minimum Password Length

Many companies require that passwords have a minimum length or use certain character combinations. You can configure a minimum password length on a router with the global configuration command **security passwords min-length** *size*.

Configure R1 to enforce a minimum password length of eight characters. Then attempt to add a user with a password length that is less than eight characters. Then add the user by lengthening the password to eight characters. Remove the user when you are done.

Note that in a production environment, you probably would want a password length that is at least eight characters.

```
R1(config)# security passwords min-length 8
R1(config)# username cisco2 password cis
% Password too short - must be at least 8 characters. Password configuration
failed
R1(config)# username cisco2 password ciscocisco
R1(config)# no username cisco2 password ciscocisco
R1(config)# no security passwords min-length 8
```

Step 5: Modify Command Privilege Levels

By default, the user EXEC prompt on a router is privilege level 1. When you type the **enable** command, the command-line interface (CLI) promotes the user's command privileges to user level 15. You can verify this with the **show privilege** command. To return to the user exec prompt, use the **disable** command.

```
R1> show privilege
Current privilege level is 1

R1> enable
Password: cisco

R1# show privilege
Current privilege level is 15
```

By default, commands are either privilege level 1 or 15, depending on how secure the command needs to be. Many commands, such as **show ip route**, can be performed at privilege level 1. Others, such as **show running-configuration** and **clear ip route ***, can be maliciously used, so they are restricted to privilege level 15. The privilege level includes all commands at that privilege level, plus every level below it. Try to perform a **clear ip route** at both privilege level 1 and 15.

```
R1> clear ip route *
           ^
% Invalid input detected at '^' marker.

R1> enable
Password: cisco
R1# clear ip route *
```

When you enter the **enable** command, the CLI prompts you for an enable password. Enter **enable ?**, and then try enabling privilege level 5.

```
R1> enable ?
  <0-15>  Enable level
  view    Set into the existing view
  <cr>

R1> enable 5
% No password set
R1>
```

You can set different privilege level passwords by using the **enable secret level** *level password* command in global configuration mode. Set the password for privilege level 5 to “cisco5”. Then, issue **enable 5** to get to privilege level 5.

```
R1(config)# enable secret level 5 cisco5

R1> enable 5
Password: cisco5

R1# show privilege
Current privilege level is 5
```

In some situations, you may want to given different access privileges to different users. This would be applicable, for instance, at an Internet Service Provider where technical support users have different access privileges.

You can change command privilege levels for specific command sequences with the **privilege** *prompt* **level** *privilege command* command in global configuration mode. The *prompt* argument is the prompt at which that command must be entered, such as EXEC, global configuration, or a particular submode. The *privilege* parameter is the minimum privilege level for this command, and *command* is the command that is modified.

For example, to allow privilege level 5 to shutdown an interface, you must change privileges on several commands. Think about the steps that someone must go through to deactivate an interface, starting at the exec prompt. The user must first be in configuration mode, and then go to interface configuration mode. Finally, the user issues the **shutdown** command on the interface.

Set the minimum privilege level for each of these commands to 5 as follows:

```
R1(config)# privilege exec level 5 configure terminal
R1(config)# privilege configure level 5 interface
R1(config)# privilege interface level 5 shutdown
```

Write down all of the commands necessary for this final command to be executed.

Using either the **enable** *level* command from the user exec prompt or the **disable** *level* command from the privileged exec prompt, change your user privilege level to 5 and attempt to deactivate the interface.

```
R1# disable 5
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ?
Configure commands:
  atm        Enable ATM SLM Statistics
  beep       Configure BEEP (Blocks Extensible Exchange Protocol)
  call       Configure Call parameters
  default    Set a command to its defaults
  dss        Configure dss parameters
  end        Exit from configure mode
  exit       Exit from configure mode
  help         Description of the interactive help system
  interface    Select an interface to configure
  netconf      Configure NETCONF
  no           Negate a command or set its defaults
  oer          Optimized Exit Routing configuration submodes
  sasl         Configure SASL

R1(config)# interface fastethernet0/0
R1(config-if)# ?
Interface configuration commands:
  default   Set a command to its defaults
  exit      Exit from interface configuration mode
  help      Description of the interactive help system
  no        Negate a command or set its defaults
  shutdown  Shutdown the selected interface

R1(config-if)# shutdown
*Feb  8 06:24:19.791: %LINK-5-CHANGED: Interface FastEthernet0/0, changed
state to administratively down
*Feb  8 06:24:20.791: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to down
R1(config-if)# no shutdown
*Feb  8 06:24:25.735: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Feb  8 06:24:26.735: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up
```

Notice the limited available command set with the **?** command. Also note that the **no shutdown** command is available when the **shutdown** command is available.

Go back to privilege level 15 and view the privilege level 5 commands that have been configured.

```
R1# show running-config | include privilege
privilege interface level 5 shutdown
privilege configure level 5 interface
privilege exec level 5 configure terminal
privilege exec level 5 configure
```

The reason there is an extra command beyond what was configured is because of the way the CLI parser receives input. With a command that has multiple keywords, such as **configure terminal**, each keyword in the sequence must be allowed. This is why there is an entry for both **configure** and **configure terminal**.

Would a user at privilege level 5 currently be able to use the **configure memory** command?

Step 6: Create a Banner

You can create a banner for users connecting to the router with the **banner** *character* command in global configuration mode. The *character* parameter is any delimiting character that you select to inform the router that the banner is complete. In the example below, a tilde (~) is used.

```
R1(config)# banner ~
Enter TEXT message.  End with the character '~'.
CCNP Lab Router
UNAUTHORIZED ACCESS PROHIBITED

Unauthorized users who attempt to connect to and perform unauthorized
operations will be prosecuted. Your actions are being monitored. The
monitoring information retrieved will be used against you in court.
~
```

Try connecting to R1 from R2.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open

CCNP Lab Router
UNAUTHORIZED ACCESS PROHIBITED

Unauthorized users who attempt to connect to and perform unauthorized
operations will be prosecuted. Your actions are being monitored. The
monitoring information retrieved will be used against you in court.

User Access Verification

Username:
```

In a production environment, the banner reflects the company security policy.

Step 7: Enable SSH

Secure Shell (SSH) is a terminal connection protocol that is similar to an encrypted version of Telnet. Since SSH is encrypted with configurable encryption methods, it is not vulnerable to packet sniffing like plain-text traffic.

This step requires a Cisco IOS image that includes the encryption suite CRYPTO. SSH includes public key encryption methods similar to those used in IPsec virtual private networks (VPNs).

First, set the domain name for the router using the **ip domain-name** *name* command in global configuration mode. In this case, set the R1 domain name to “cisco.com”.

```
R1(config)# ip domain-name cisco.com
```

Next, generate RSA encryption keys with the **crypto key generate rsa** command in global configuration mode. When prompted for the number of bits in the modulus, enter 1024. Notice that the name of the keys is the hostname of the router concatenated with the domain name you configured.

```
R1(config)# crypto key generate rsa
The name for the keys will be: R1.cisco.com
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#
*Feb  8 07:10:14.027: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

Once you have generated the RSA keys, SSH is enabled automatically. On R1, you can view the crypto keys generated with the **show crypto key mypubkey rsa** command.

```
R1# show crypto key mypubkey rsa
% Key pair was generated at: 07:10:14 UTC Feb 8 2007
Key name: R1.cisco.com
 Storage Device: not specified
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88087
<OUTPUT OMITTED>
```

By default, all virtual terminal lines (VTYs) allow connections using all available connection protocols, such as Telnet, SSH, rlogin, and so on. For enhanced security, force the VTYs to accept only SSH traffic by using the **transport input** *type* command in line configuration mode. The *type* parameter lists the incoming and outgoing connection types allowed.

```
R1(config)# line vty 0 4
R1(config-line)# transport input ssh
```

On R2, you can try to connect to R1 with SSH by using the **ssh –l** *username hostname* command. Use the username configured in the local username database on R1.

```
R2# ssh -l cisco 192.168.10.1

Password: cisco

CCNP Lab Router
UNAUTHORIZED ACCESS PROHIBITED

Unauthorized users who attempt to connect to and perform unauthorized
operations will be prosecuted. Your actions are being monitored. The
monitoring information retrieved will be used against you in court.

R1>exit

[Connection to 192.168.10.1 closed by foreign host]
R2#
```

Step 8: Encrypt Passwords

The **show running-config** command displays usernames and passwords unencrypted. Other types of passwords, such as line passwords, also show up as unencrypted. Secret passwords do not show up unencrypted, because they already have the MD5 algorithm performed on them.

```
R1# show running-config | include username
username cisco password 0 cisco
```

To secure your router from access attacks in which a user sees an unencrypted password and uses it to raise their privilege level to something higher than the user should have, issue the service password-encryption command in global configuration mode. Display the username entries in the running configuration.

```
R1(config)# service password-encryption

R1# show running-config | include username
username cisco password 7 070C285F4D06
```

The passwords in the running configuration that were previously unencrypted now display in encrypted form. This encrypted form is easily reversible and is not the same as the MD5 hash used for secret passwords. It is only intended to prevent "shoulder surfing," which is learning passwords from eavesdropping on someone else's workstation. Since it is reversible, you should not send out configurations with passwords encrypted in this form, because once someone has a hardcopy of the encrypted password, they can easily retrieve the plain-text version.

To see how easy it is to decode a Cisco level 7 password, go to google.com and type in, "cisco level 7 password". There will be several sites that have the ability to decode Cisco level 7 passwords. Cut and paste the level 7 password from above into this URL: http://www.securitystats.com/tools/ciscocrack.php You will see that the weakly encrypted password is easily decoded to yield the plain text password "cisco".

Final Configuration

```
R1# show run
service password-encryption
!
hostname R1
!
security passwords min-length 8
enable secret level 5 5 $1$aKRq$uPRFZlcoQz7LI8PMqreul/
enable secret 5 $1$dGMq$3r5OinUfI.faiFqHRjqfT/
!
ip domain name cisco.com
login block-for 30 attempts 2 within 15
login delay 3
login quiet-mode access-class 1
login on-failure log
!
```

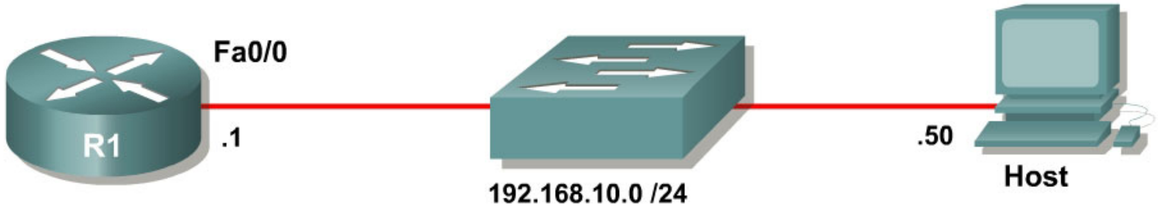
```
username cisco password 7 070C285F4D06
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
!
access-list 1 permit 192.168.20.0 0.0.0.255
!
banner motd ^C
CCNP Lab Router
UNAUTHORIZED ACCESS PROHIBITED

Unauthorized users who attempt to connect to and perform unauthorized
operations will be prosecuted. Your actions are being monitored. Any
monitoring information retrieved will be used against you in court.
^C
privilege interface level 5 shutdown
privilege configure level 5 interface
privilege exec level 5 configure terminal
privilege exec level 5 configure
!
line vty 0 4
 login local
 transport input ssh
end

R2# show run
hostname R2
!
interface FastEthernet0/0
 ip address 192.168.10.2 255.255.255.0
 no shutdown
end
```

Lab 5.5 Configuring Logging

Learning Objectives

- Configure a router to log to a Syslog server
- Use Kiwi Syslog Daemon as a Syslog server
- Configure local buffering on a router

Topology Diagram



Scenario

In this lab, you will configure a router to log system messages and notifications to a Syslog server. You will also view the logs on the Syslog server.

Step 1: Configure the interface

Configure the router interface shown in the topology diagram.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
```

Apply the IP address shown in the topology diagram to the host. If you do not know how to set up an IP address on a host, consult Lab 3.1: Configuring SDM on a Router.

Verify that you have connectivity between R1 and the host with the **ping** command.

```
R1# ping 192.168.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Step 2: Install the Kiwi Syslog Daemon

This lab uses the Kiwi Syslog Daemon, which is a free Syslog server for use with Microsoft Windows. If it is not currently installed on the host, download the installer from http://www.kiwisyslog.com. If it is installed, skip to step 3.

As of the time of this writing, there are two versions of this software: a free version and a licensed version. This lab uses only the features found in the free version. When prompted to install the program as a service or application, you can choose whether you want the Syslog daemon started automatically as a system service or triggered by user action, like a normal Windows application. This lab was written using the service installation.

Step 3: Run the Kiwi Syslog Service Manager

Open the Kiwi Syslog Daemon Manager, which can be accessed either by the icon on the host's desktop labeled **Kiwi Syslog Daemon** or by clicking on the **Start** button and choosing **Programs > Kiwi Enterprises > Kiwi Syslog Daemon > Kiwi Syslog Daemon**.

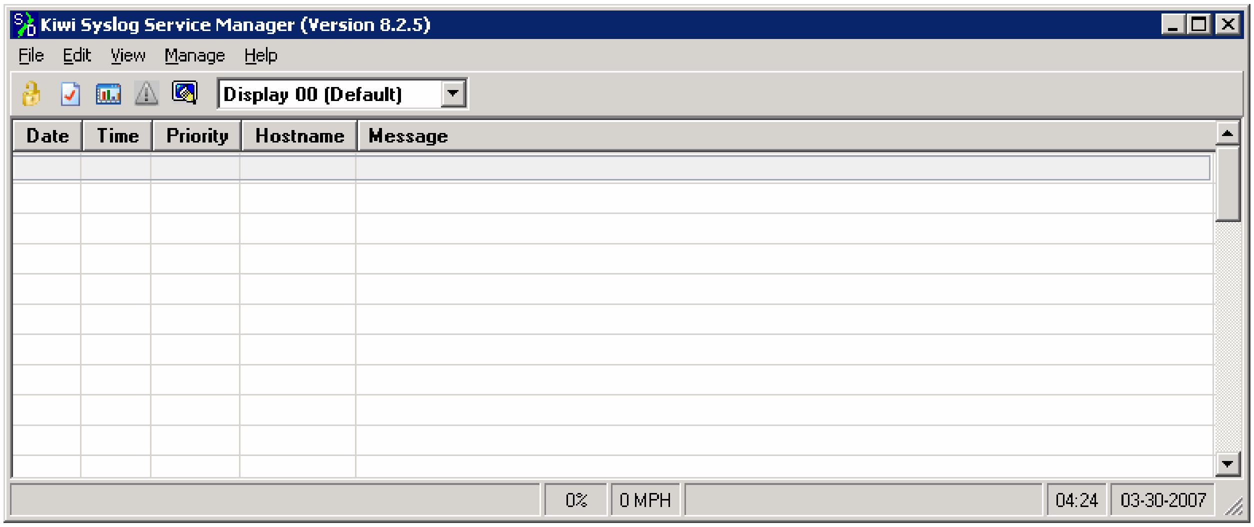

Kiwi Syslog Service Manager (Version 8.2.5)
File Edit View Manage Help
Display 00 (Default)
Date Time Priority Hostname Message
0%
0 MPH
04:24
03-30-2007

Figure 3-1: Kiwi Syslog Daemon Manager Main Window

If this is your first time running the program after installing it, choose **Manage > Install the Syslogd service**. You need to start the service if you just installed it, or if you are not sure the service is running. Start the service with **Manage > Start the Syslogd service**. You can check if the service is running by selecting **Manage > Ping the Syslogd service**.

Step 4: Configure the Router for Logging

Configuring a router to use a Syslog server is a relatively simple process and only requires a few commands in global configuration mode.

First, configure the IP address of a Syslog server with the **logging host** *hostname* command. In this lab, use an IP address instead of a hostname.

```
R1(config)# logging host 192.168.10.50
```

Set the Syslog severity level with the global configuration command **logging trap** *level*. You can specify the severity level by either using a keyword or an integer from 0 to 7.

```
R1(config)# logging trap ?
  <0-7>          Logging severity level
  alerts         Immediate action needed           (severity=1)
  critical       Critical conditions               (severity=2)
  debugging      Debugging messages                (severity=7)
  emergencies    System is unusable                (severity=0)
  errors         Error conditions                  (severity=3)
  informational  Informational messages            (severity=6)
  notifications  Normal but significant conditions (severity=5)
  warnings       Warning conditions                (severity=4)
  <cr>
```

The severity levels in order are as follows:

- Emergencies – 0
- Alerts – 1
- Critical – 2
- Errors – 3
- Warnings – 4
- Notifications – 5
- Informational – 6 (default)
- Debugging –7

Each severity level includes the severity levels with lower numbers. This may seem a little counter-intuitive, so predict what will happen in this example.

Predict which severity levels of messages would be logged if you issued the following command:

```
Router(config)# logging trap critical
```

The default level is 6, informational logging. To demonstrate the command, set the logging trap level to informational. Note that the command does not show up in the running configuration, because it is the default.

```
R1(config)# logging trap informational
```

Generate some logging messages for your log server by configuring your device to log users entering and exiting privileged mode with the **logging userinfo** command. When you have completed entering commands, enter the **end** command to exit configuration mode. This user action generates a Syslog message that the router was just configured. There may also be another Syslog message stating that logging to the host just started.

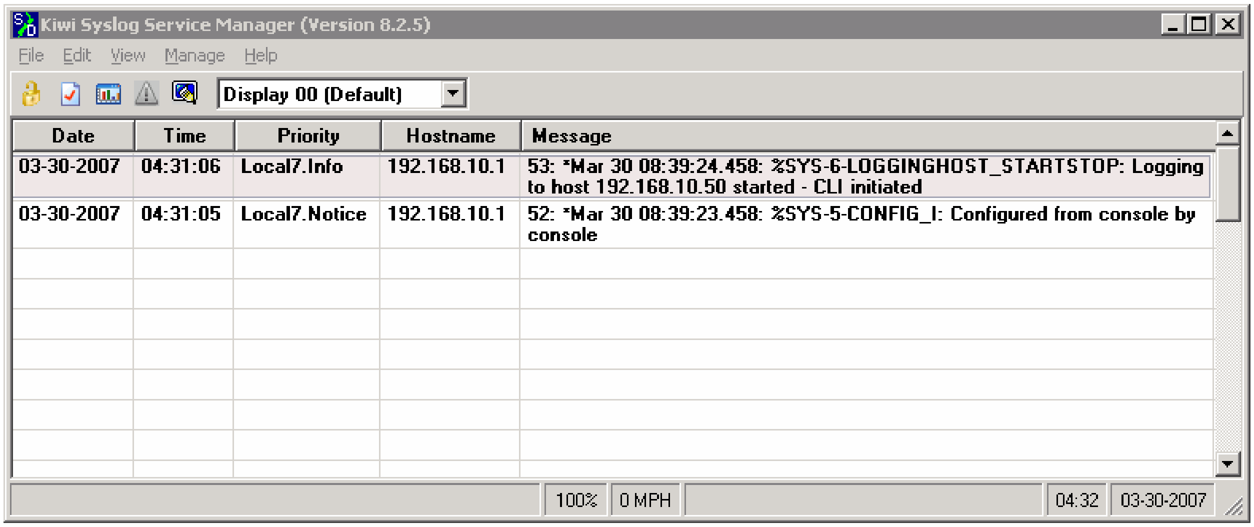
```
R1(config)# logging userinfo
R1(config)# end
R1#

*Mar 30 08:39:23.458: %SYS-5-CONFIG_I: Configured from console by console
*Mar 30 08:39:24.458: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host
192.168.10.50 started - CLI initiated
```

You may also want to verify logging settings with the **show logging** command.

```
R1# show logging
Syslog logging: enabled (11 messages dropped, 1 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 46 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: disabled, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.

    Trap logging: level informational, 50 message lines logged
        Logging to 192.168.10.50(global) (udp port 514, audit disabled, link
up), 2 message lines logged, xml disabled,
               filtering disabled
```

Step 5: Verify Logging

On the host, look at the Kiwi Syslog Daemon Manager. The log messages that were just created will be displayed.



Kiwi Syslog Service Manager (Version 8.2.5)
File Edit View Manage Help
Display 00 (Default)
100%
0 MPH
04:32
03-30-2007

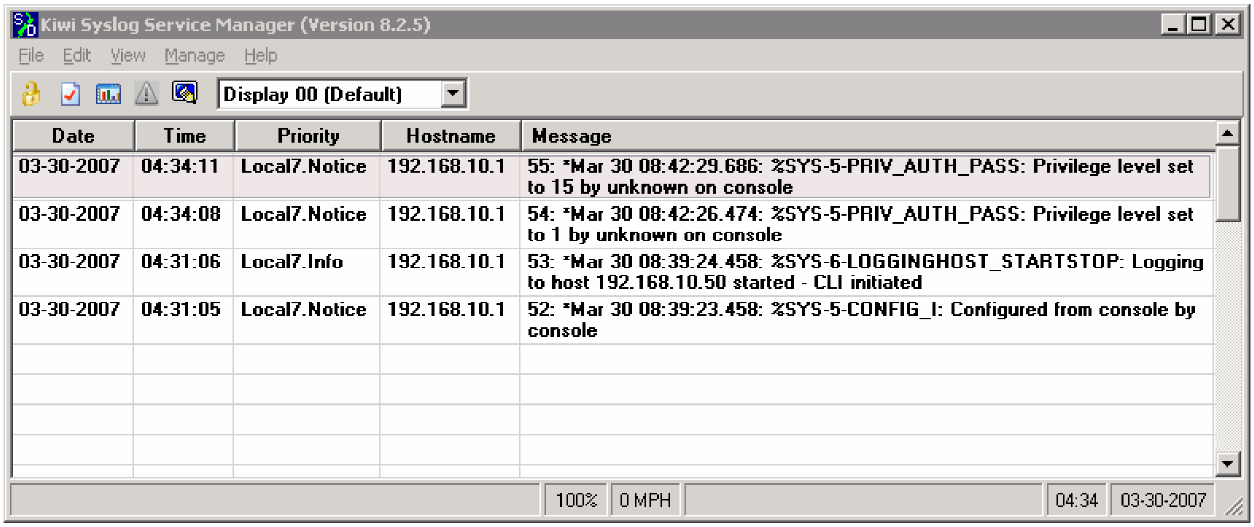
Date	Time	Priority	Hostname	Message
03-30-2007	04:31:06	Local7.Info	192.168.10.1	53: *Mar 30 08:39:24.458: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 192.168.10.50 started - CLI initiated
03-30-2007	04:31:05	Local7.Notice	192.168.10.1	52: *Mar 30 08:39:23.458: %SYS-5-CONFIG_I: Configured from console by console

Figure 5-1: Informational Log Messages from R1

On the router, exit privileged EXEC mode, and then reenter it using the **enable** command.

Log messages will appear on the router as well as in the Kiwi Syslog Daemon Manager. The reason log messages are generated is because of the **logging userinfo** command you issued earlier.

```
R1# disable
*Mar 30 08:42:26.474: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 1 by
unknown on console
R1> enable
*Mar 30 08:42:29.686: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 15 by
unknown on console
R1#
```



Kiwi Syslog Service Manager (Version 8.2.5)
File Edit View Manage Help
Display 00 (Default)
100%
0 MPH
04:34
03-30-2007

Date	Time	Priority	Hostname	Message
03-30-2007	04:34:11	Local7.Notice	192.168.10.1	55: *Mar 30 08:42:29.686: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 15 by unknown on console
03-30-2007	04:34:08	Local7.Notice	192.168.10.1	54: *Mar 30 08:42:26.474: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 1 by unknown on console
03-30-2007	04:31:06	Local7.Info	192.168.10.1	53: *Mar 30 08:39:24.458: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 192.168.10.50 started - CLI initiated
03-30-2007	04:31:05	Local7.Notice	192.168.10.1	52: *Mar 30 08:39:23.458: %SYS-5-CONFIG_I: Configured from console by console

Figure 5-2: User Information Log Messages from R1

Why it is better to have centralized logging servers rather than only have the routers log locally?

Step 6: Configure Buffered Logging

In cases where you have a small network and do not have a centralized logging server, you may consider buffering logs to a local memory buffer. The commands coincide with those used for the Syslog server.

Issue the **logging buffered** [*bytes*] [*severity-level*] command on R1 to begin buffering to the local buffer. Use the informational level and set the buffer size to 32 KB. Exit global configuration mode, which generates a log message.

```
R1(config)# logging buffered 32768 informational
R1(config)# exit
R1#

*Mar 30 14:44:56.968: %SYS-5-CONFIG_I: Configured from console by console
```

Issue the **show logging** command to get general information about the buffer and view the buffer log.

```
R1# show logging
Syslog logging: enabled (11 messages dropped, 1 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 54 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level informational, 1 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.

    Trap logging: level informational, 58 message lines logged
        Logging to 192.168.10.50(global) (udp port 514, audit disabled, link
up), 6 message lines logged, xml disabled,
               filtering disabled

Log Buffer (32768 bytes):

*Mar 30 14:44:56.968: %SYS-5-CONFIG_I: Configured from console by console
```

Exit privileged EXEC mode and then reenter it. This generates some user information messages that are saved to the memory buffer.

```
R1# disable
```

```
*Mar 30 14:45:22.272: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 1 by
unknown on console

R1> enable
R1#
*Mar 30 14:45:23.200: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 15 by
unknown on console
```

Display the contents of the internal buffer again with the **show logging** command.

```
R1# show logging
Syslog logging: enabled (11 messages dropped, 1 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 56 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level informational, 3 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.

    Trap logging: level informational, 60 message lines logged
        Logging to 192.168.10.50(global) (udp port 514, audit disabled, link
up), 8 message lines logged, xml disabled,
               filtering disabled

Log Buffer (32768 bytes):

*Mar 30 14:44:56.968: %SYS-5-CONFIG_I: Configured from console by console
*Mar 30 14:45:22.272: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 1 by
unknown on console
*Mar 30 14:45:23.200: %SYS-5-PRIV_AUTH_PASS: Privilege level set to 15 by
unknown on console
```

Final Configuration

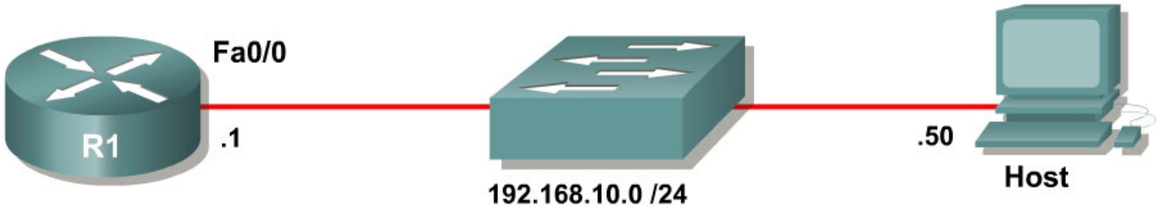
```
R1# show run
!
hostname R1
!
logging userinfo
logging buffered 32768 informational
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
!
logging 192.168.10.50
!
end
```

Lab 5.6a Configuring AAA and TACACS+

Learning Objectives

- Install CiscoSecure ACS
- Configure CiscoSecure ACS as a TACACS+ server
- Enable AAA on a router using a remote TACACS+ server

Topology Diagram



Scenario

In this lab, you will set up CiscoSecure ACS as a TACACS+ server. You will also set up R1 to use authentication, authorization, and accounting (AAA) services for line authentication.

Step 1: Configure the Interface

Configure the router interface shown in the topology diagram.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
```

Configure the IP address of 192.168.10.50/24 on the host.

Verify that you have connectivity between R1 and the host with the **ping** command.

```
R1# ping 192.168.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Step 2: Install CiscoSecure ACS

If you have already installed CiscoSecure ACS, go to step 3.

This step guides you through installing the 90-day trial version of CiscoSecure ACS. After you download the trial and extract it, run Setup.exe.

Note: At the time of this writing, CiscoSecure ACS only runs on Microsoft Windows Server Editions. You cannot run CiscoSecure ACS on Microsoft Windows XP.

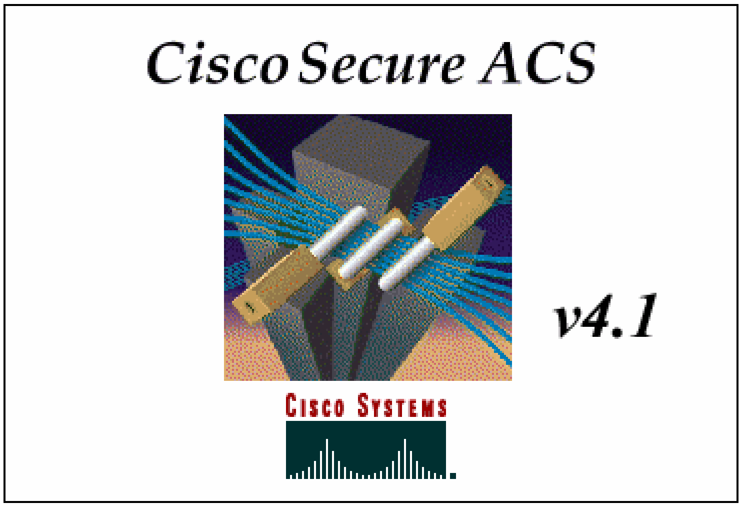
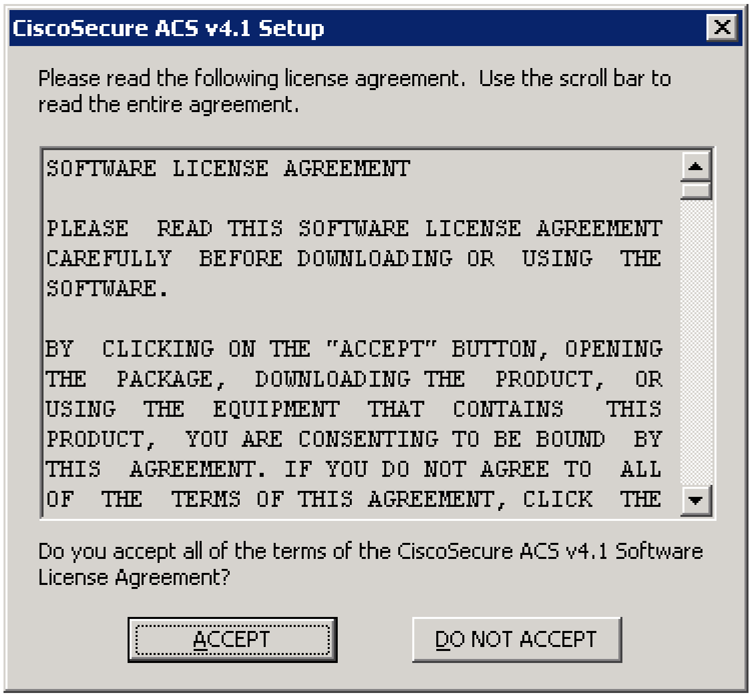


Figure 2-1: CiscoSecure ACS Splash Screen

After reading the terms of the license agreement, click **ACCEPT**.



Please read the following license agreement. Use the scroll bar to read the entire agreement.

SOFTWARE LICENSE AGREEMENT

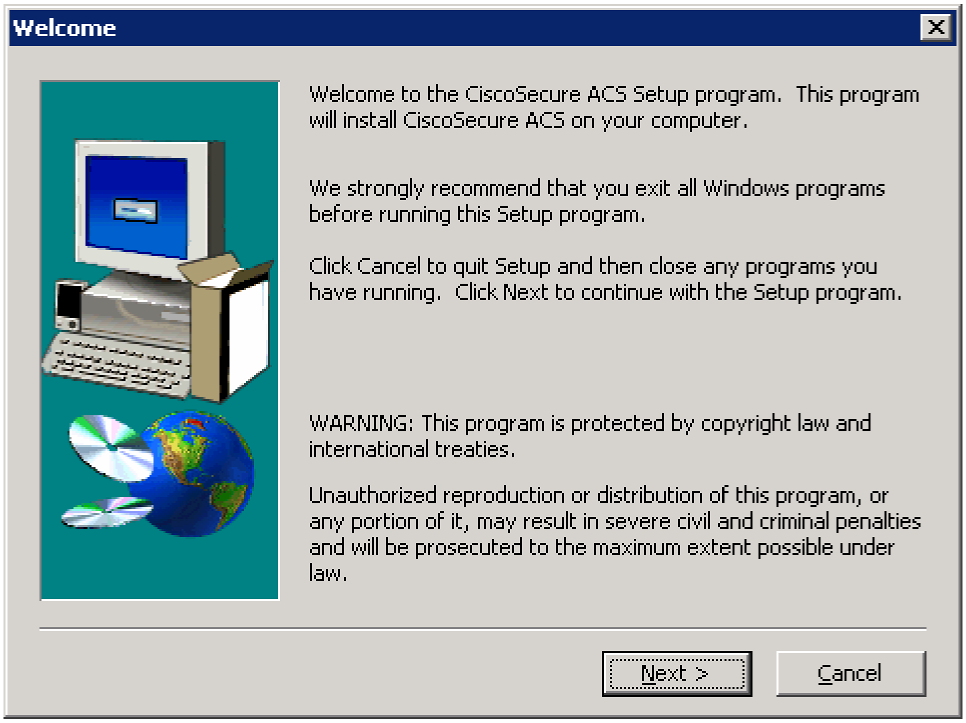
PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE DOWNLOADING OR USING THE SOFTWARE.

BY CLICKING ON THE "ACCEPT" BUTTON, OPENING THE PACKAGE, DOWNLOADING THE PRODUCT, OR USING THE EQUIPMENT THAT CONTAINS THIS PRODUCT, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE

Do you accept all of the terms of the CiscoSecure ACS v4.1 Software License Agreement?

Figure 2-2: CiscoSecure ACS License Agreement

Click **Next** to continue the installation process.



Welcome

Welcome to the CiscoSecure ACS Setup program. This program will install CiscoSecure ACS on your computer.

We strongly recommend that you exit all Windows programs before running this Setup program.

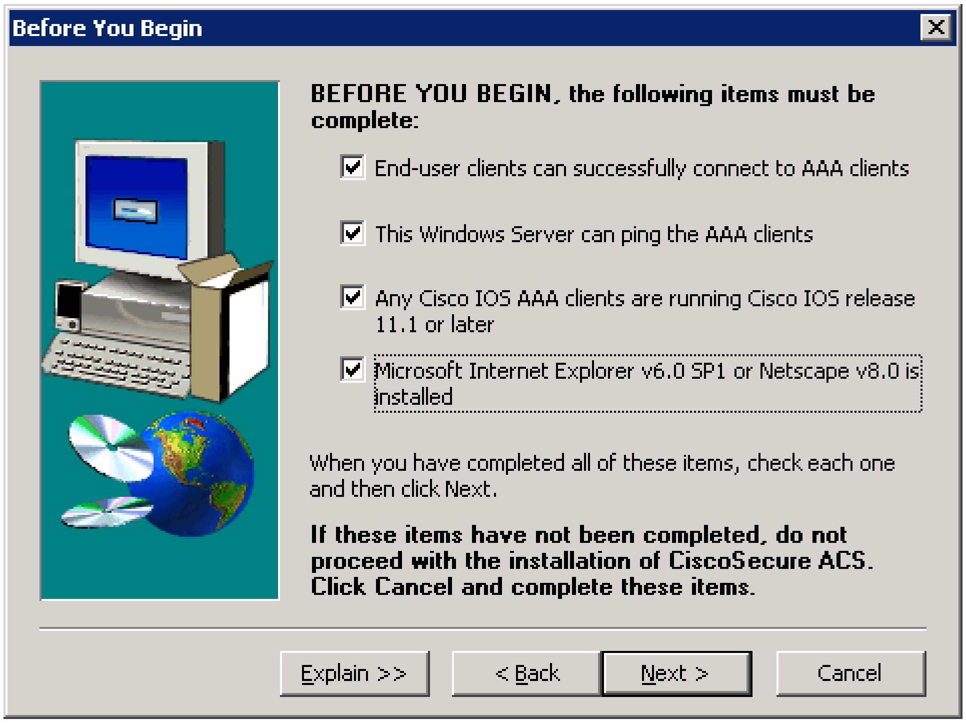
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties and will be prosecuted to the maximum extent possible under law.

Figure 2-3: CiscoSecure ACS Installation Wizard

Verify that all the requirements in the checklist are satisfied and check all the options before clicking **Next**.


Before You Begin

BEFORE YOU BEGIN, the following items must be complete:

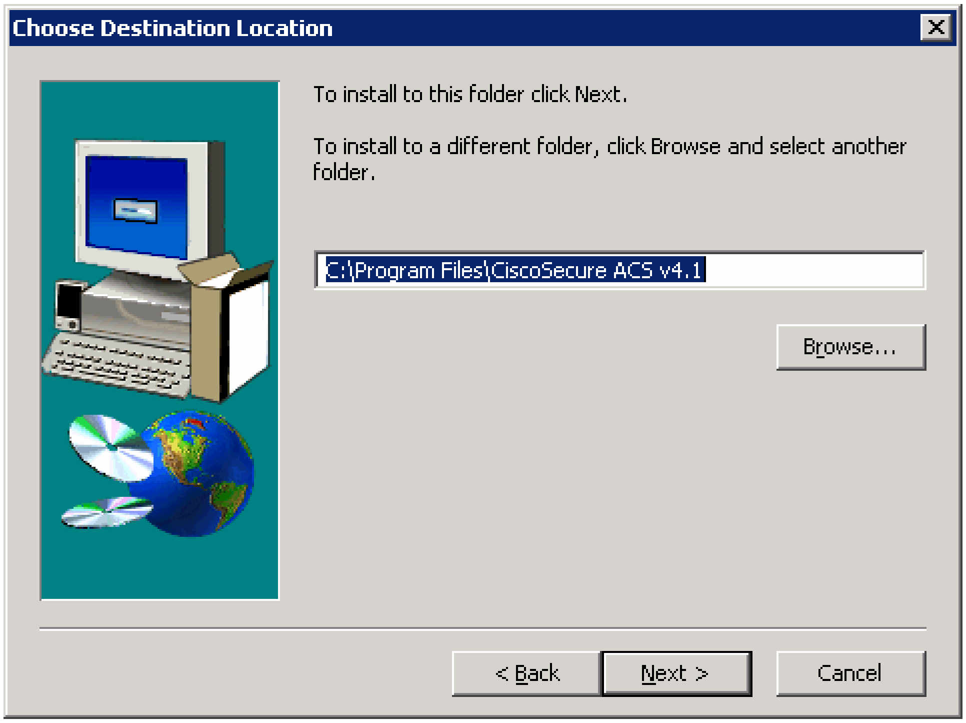
- ☑ End-user clients can successfully connect to AAA clients
- ☑ This Windows Server can ping the AAA clients
- ☑ Any Cisco IOS AAA clients are running Cisco IOS release 11.1 or later
- ☑ Microsoft Internet Explorer v6.0 SP1 or Netscape v8.0 is installed

When you have completed all of these items, check each one and then click Next.

If these items have not been completed, do not proceed with the installation of CiscoSecure ACS. Click Cancel and complete these items.

Figure 2-4: CiscoSecure ACS Pre-Installation Checklist

Use the default installation folder and click **Next**.


Choose Destination Location

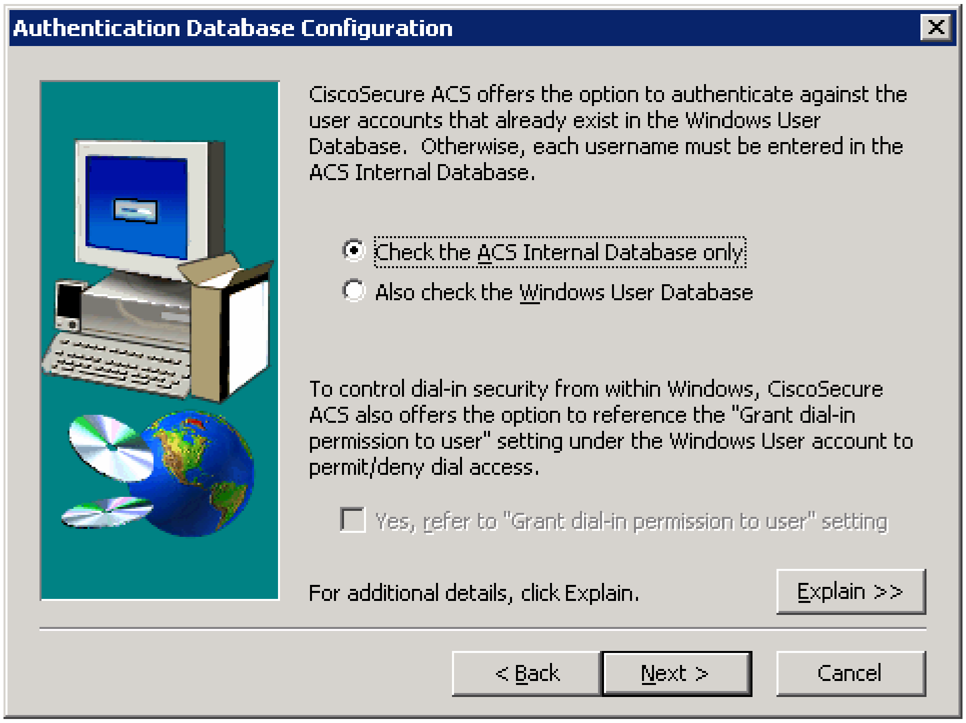
To install to this folder click Next.

To install to a different folder, click Browse and select another folder.

C:\Program Files\CiscoSecure ACS v4.1

Figure 2-5: CiscoSecure ACS Installation Location

CiscoSecure has the ability to check the Windows User Database. However, for this lab, choose to authenticate using the internal database only. Click **Next**.



Authentication Database Configuration

CiscoSecure ACS offers the option to authenticate against the user accounts that already exist in the Windows User Database. Otherwise, each username must be entered in the ACS Internal Database.

● Check the ACS Internal Database only

○ Also check the Windows User Database

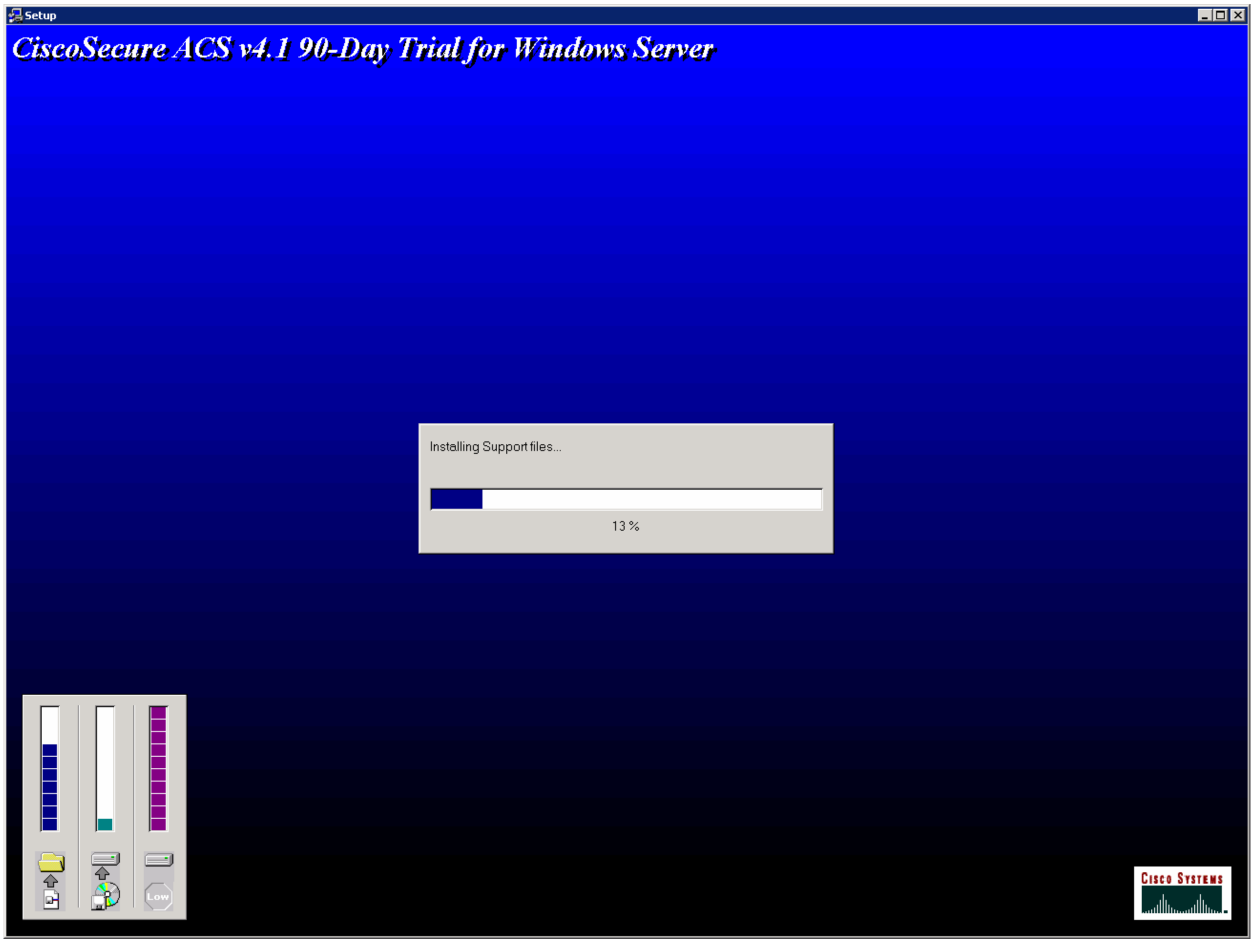
To control dial-in security from within Windows, CiscoSecure ACS also offers the option to reference the "Grant dial-in permission to user" setting under the Windows User account to permit/deny dial access.

□ Yes, refer to "Grant dial-in permission to user" setting

For additional details, click Explain.

Figure 2-6: CiscoSecure ACS Authentication Database Options

The installer will begin copying files and registry keys. This process may take a few minutes.

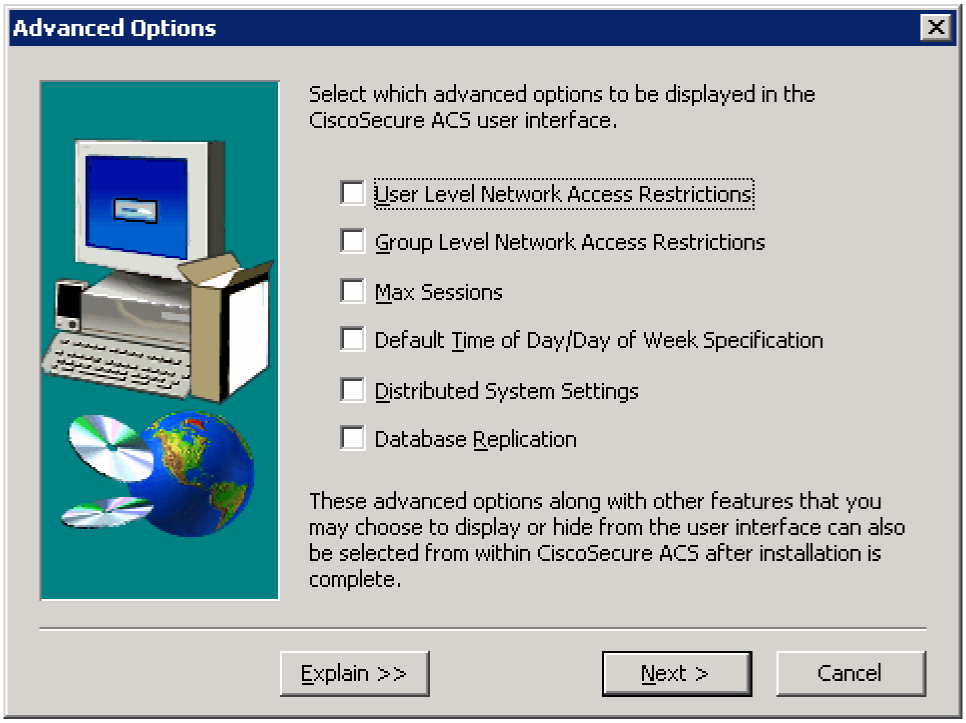


Installing Support files...

13 %

Figure 2-7: CiscoSecure ACS Installation Progress Indicator

At the end of the installer, you are prompted whether you want to see any advanced configuration options in the user interface. You do not need to select any of these. Click **Next** after reviewing the options.


Advanced Options

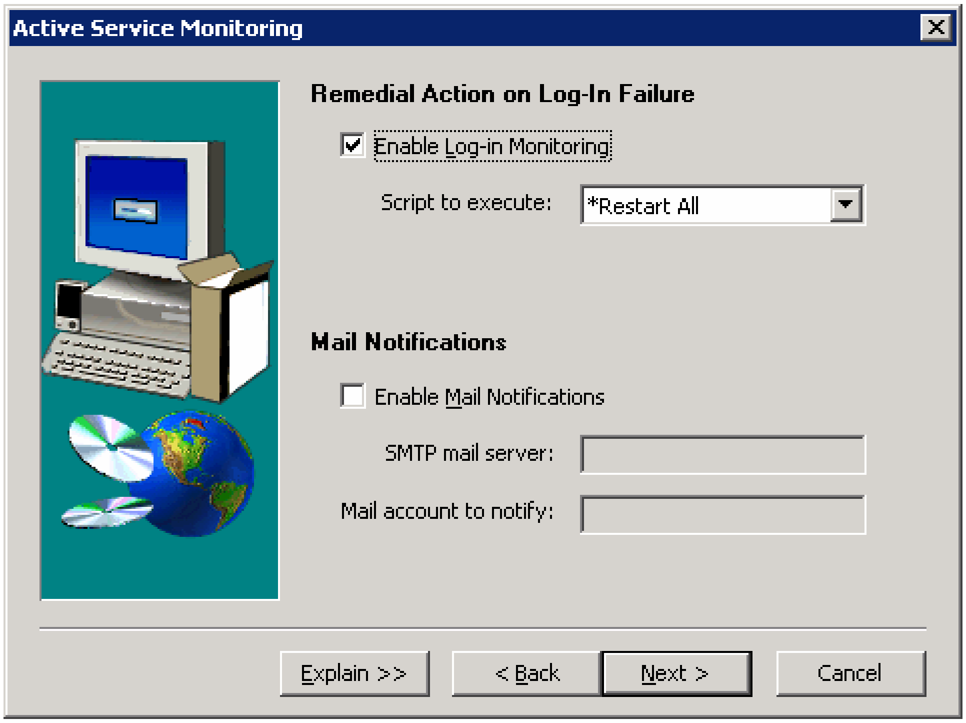
Select which advanced options to be displayed in the CiscoSecure ACS user interface.

- □ User Level Network Access Restrictions
- □ Group Level Network Access Restrictions
- □ Max Sessions
- □ Default Time of Day/Day of Week Specification
- □ Distributed System Settings
- □ Database Replication

These advanced options along with other features that you may choose to display or hide from the user interface can also be selected from within CiscoSecure ACS after installation is complete.

Figure 2-8: CiscoSecure ACS Advanced Configuration Options

Keep the default settings in the next step of the installation wizard and click **Next**.


Active Service Monitoring

Remedial Action on Log-In Failure

☑ Enable Log-in Monitoring

Script to execute: *Restart All

Mail Notifications

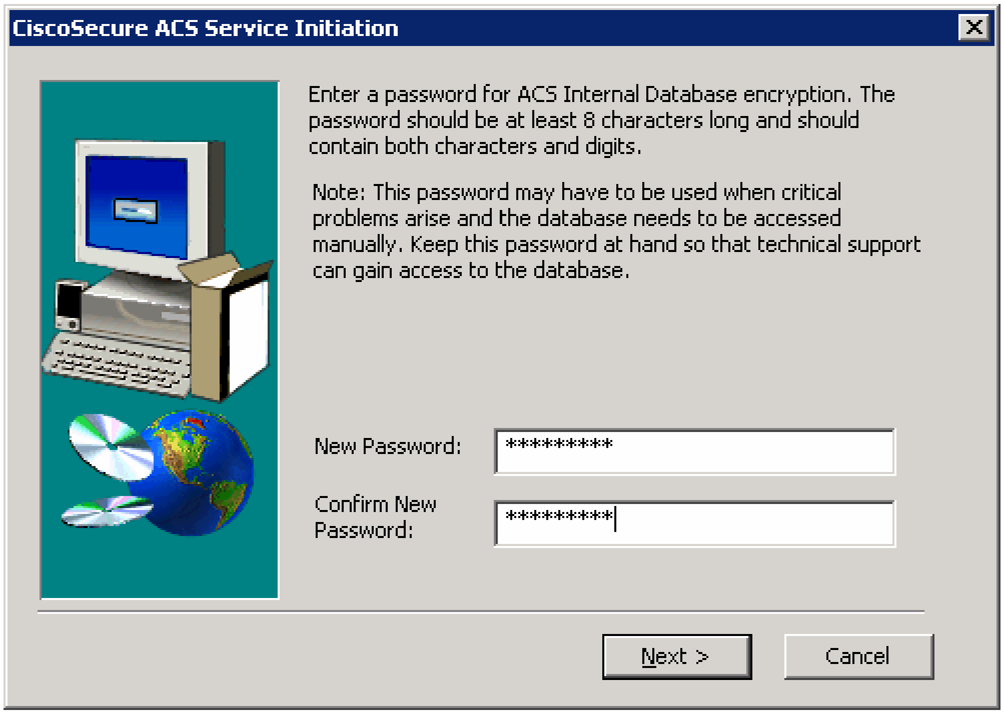
□ Enable Mail Notifications

SMTP mail server:

Mail account to notify:

Figure 2-9: CiscoSecure ACS Log-In

You must create a password for ACS internal database encryption. It must be at least eight characters and contain both letters and numbers. In the example below, “ciscoacs4” is used as the password. After configuring the password, click **Next**.



CiscoSecure ACS Service Initiation

Enter a password for ACS Internal Database encryption. The password should be at least 8 characters long and should contain both characters and digits.

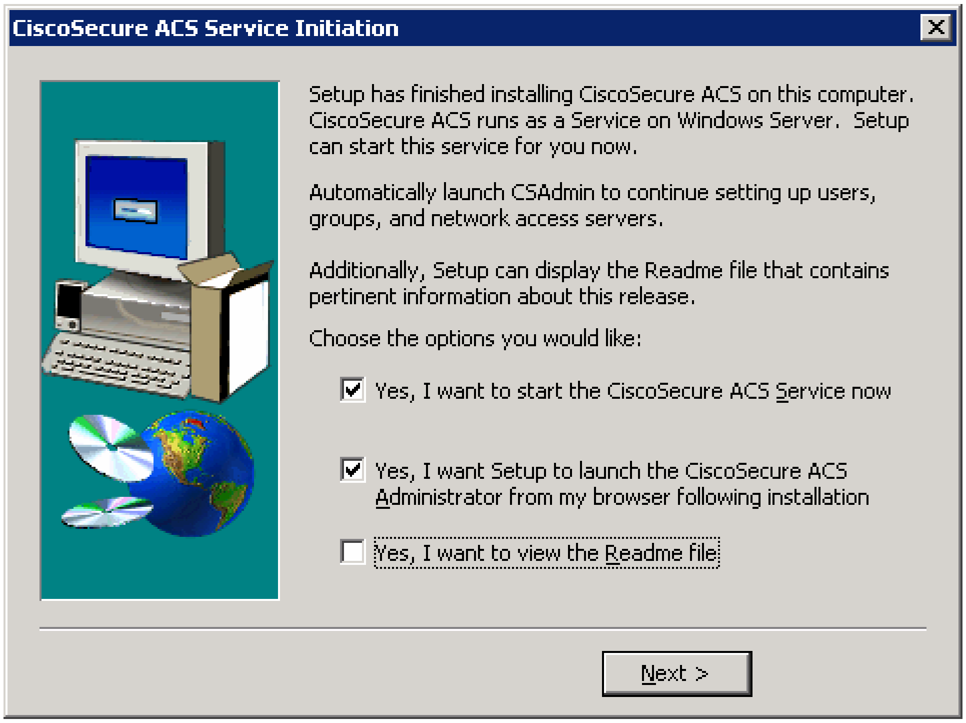
Note: This password may have to be used when critical problems arise and the database needs to be accessed manually. Keep this password at hand so that technical support can gain access to the database.

New Password: *********

Confirm New Password: *********

Figure 2-9: CiscoSecure ACS Password Configuration

Choose to start the ACS service on the host now. You should also select the option to start the administration window after the installer ends to verify the installation. Click **Next** after selecting the options.



CiscoSecure ACS Service Initiation

Setup has finished installing CiscoSecure ACS on this computer. CiscoSecure ACS runs as a Service on Windows Server. Setup can start this service for you now.

Automatically launch CSAdmin to continue setting up users, groups, and network access servers.

Additionally, Setup can display the Readme file that contains pertinent information about this release.

Choose the options you would like:

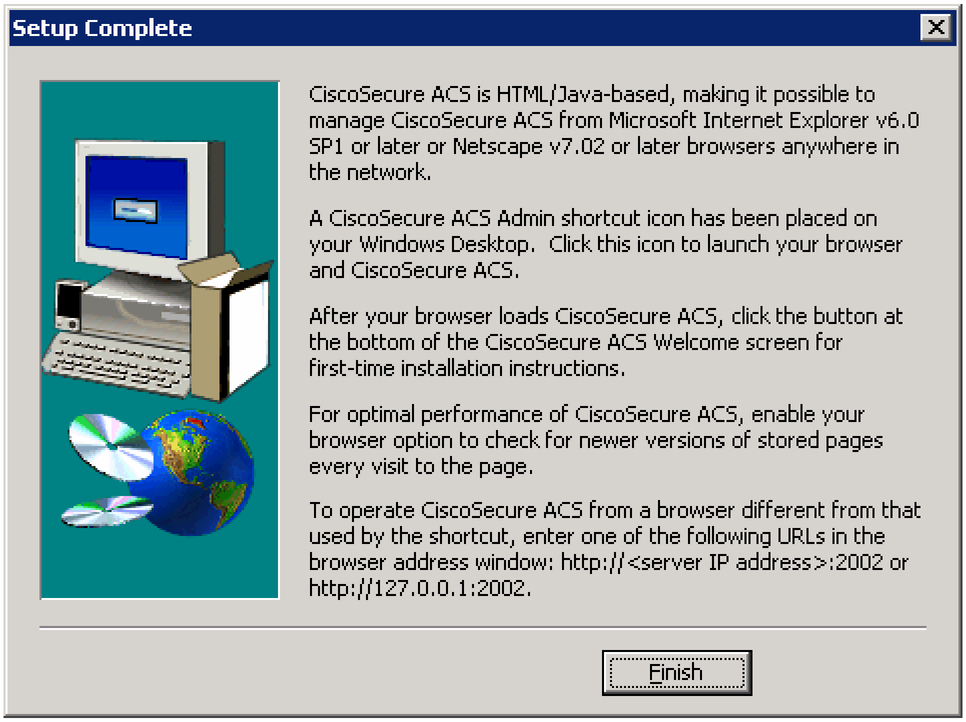
☑ Yes, I want to start the CiscoSecure ACS Service now

☑ Yes, I want Setup to launch the CiscoSecure ACS Administrator from my browser following installation

☐ Yes, I want to view the Readme file

Figure 2-10: CiscoSecure ACS Service Configuration

Read the instructions and click **Finish**. You should also make sure your computer is compliant with all ACS access requirements, complying with the supported versions of Internet Explorer and the Java Runtime Environment.



Setup Complete

CiscoSecure ACS is HTML/Java-based, making it possible to manage CiscoSecure ACS from Microsoft Internet Explorer v6.0 SP1 or later or Netscape v7.02 or later browsers anywhere in the network.

A CiscoSecure ACS Admin shortcut icon has been placed on your Windows Desktop. Click this icon to launch your browser and CiscoSecure ACS.

After your browser loads CiscoSecure ACS, click the button at the bottom of the CiscoSecure ACS Welcome screen for first-time installation instructions.

For optimal performance of CiscoSecure ACS, enable your browser option to check for newer versions of stored pages every visit to the page.

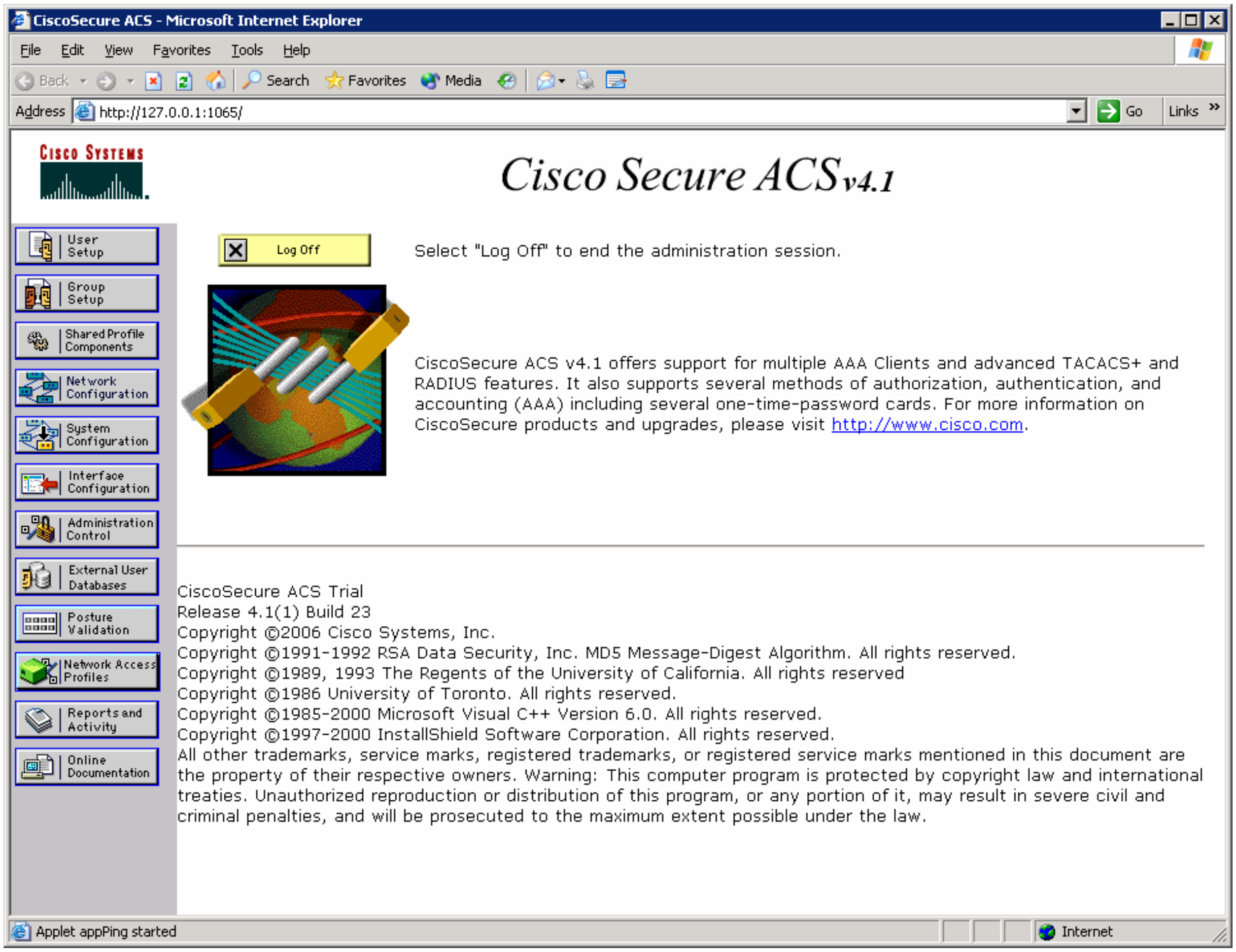
To operate CiscoSecure ACS from a browser different from that used by the shortcut, enter one of the following URLs in the browser address window: http://<server IP address>:2002 or http://127.0.0.1:2002.

Figure 2-11: CiscoSecure ACS Installation Complete Window

If the CiscoSecure ACS administrative screen comes up when the installer ends, it was successfully installed.

Step 3: Configure Users in CiscoSecure ACS

If CiscoSecure ACS is not open, start it by clicking the **Start** button and choosing **Programs > CiscoSecure ACS v4.1 Trial > ACS Admin**.



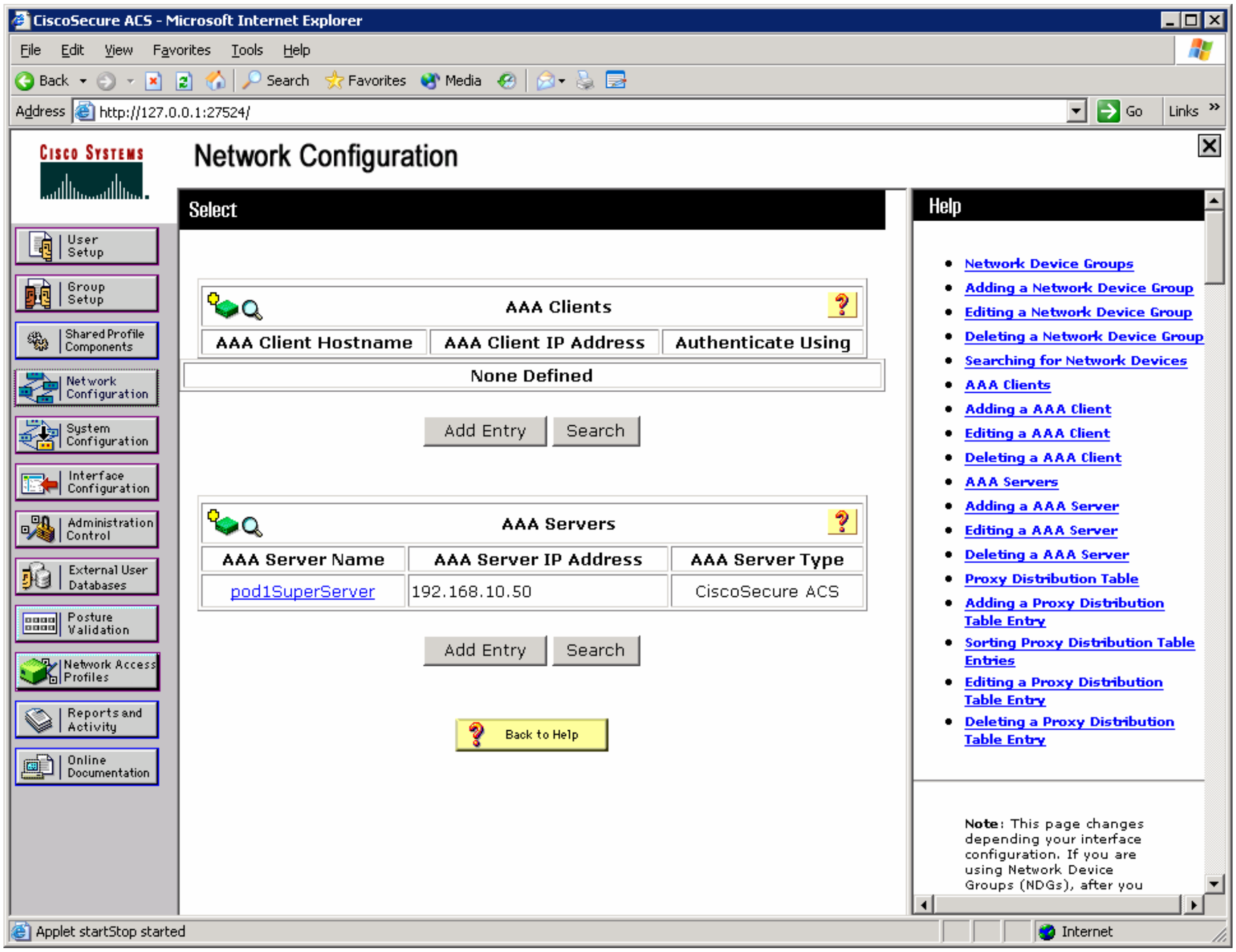
Select "Log Off" to end the administration session.

CiscoSecure ACS v4.1 offers support for multiple AAA Clients and advanced TACACS+ and RADIUS features. It also supports several methods of authorization, authentication, and accounting (AAA) including several one-time-password cards. For more information on CiscoSecure products and upgrades, please visit http://www.cisco.com.

CiscoSecure ACS Trial
Release 4.1(1) Build 23
Copyright ©2006 Cisco Systems, Inc.
Copyright ©1991-1992 RSA Data Security, Inc. MD5 Message-Digest Algorithm. All rights reserved.
Copyright ©1989, 1993 The Regents of the University of California. All rights reserved
Copyright ©1986 University of Toronto. All rights reserved.
Copyright ©1985-2000 Microsoft Visual C++ Version 6.0. All rights reserved.
Copyright ©1997-2000 InstallShield Software Corporation. All rights reserved.
All other trademarks, service marks, registered trademarks, or registered service marks mentioned in this document are the property of their respective owners. Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 3-1: ACS Home Page

Click the **Network Configuration** button on the left side. On this screen, you can configure AAA clients directly. Click **Add Entry** under the heading AAA Clients.



AAA Clients

AAA Client Hostname	AAA Client IP Address	Authenticate Using
None Defined		

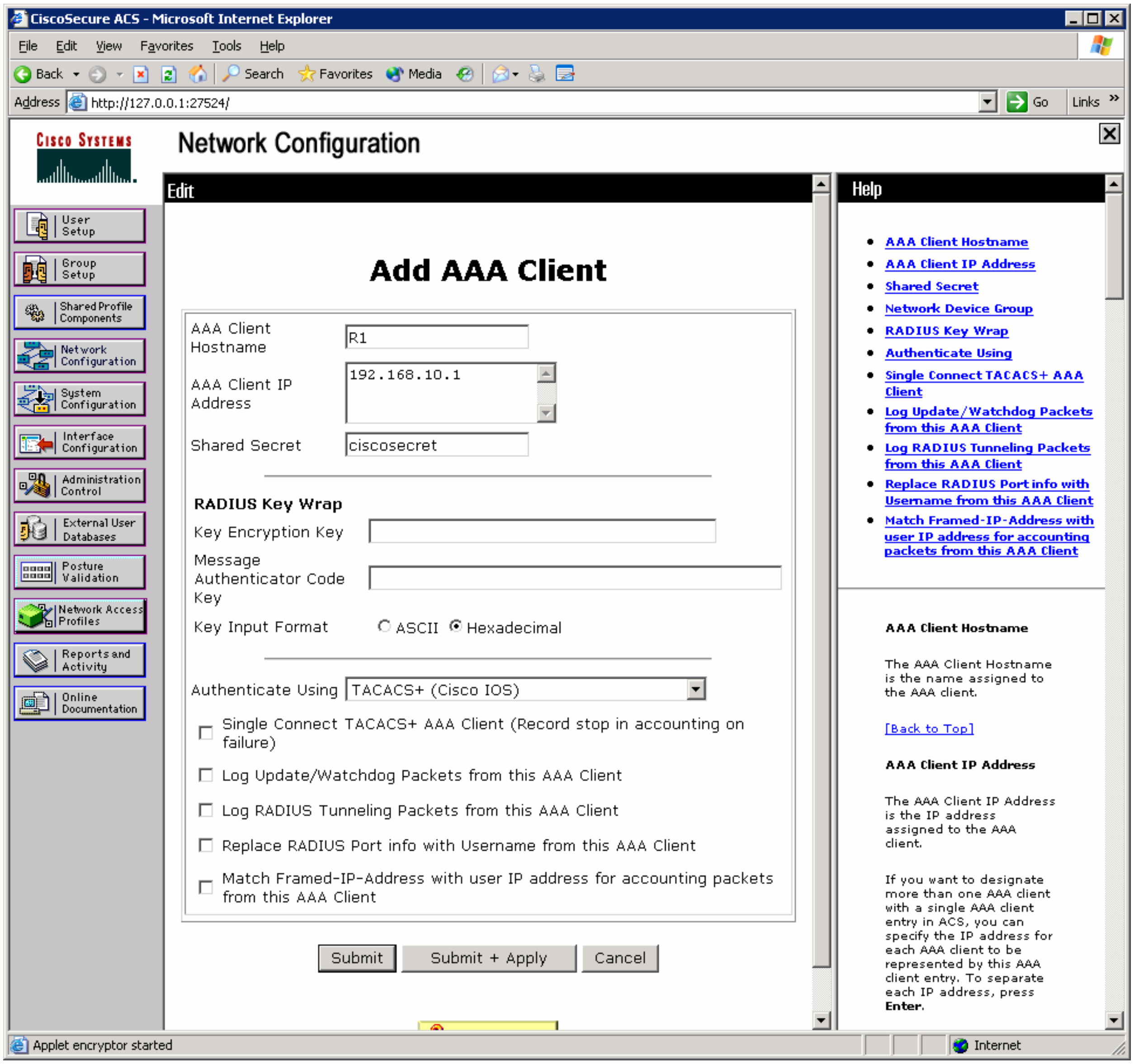
AAA Servers

AAA Server Name	AAA Server IP Address	AAA Server Type
pod1SuperServer	192.168.10.50	CiscoSecure ACS

Note: This page changes depending your interface configuration. If you are using Network Device Groups (NDGs), after you

Figure 3-2: ACS Network Configuration Page

Configure R1 as a TACACS+ client as shown below, and then click **Submit + Apply**.



Add AAA Client

AAA Client Hostname: R1

AAA Client IP Address: 192.168.10.1

Shared Secret: ciscosecret

RADIUS Key Wrap

Key Encryption Key:

Message Authenticator Code Key:

Key Input Format ○ ASCII ● Hexadecimal

Authenticate Using: TACACS+ (Cisco IOS)

□ Single Connect TACACS+ AAA Client (Record stop in accounting on failure)

□ Log Update/Watchdog Packets from this AAA Client

□ Log RADIUS Tunneling Packets from this AAA Client

□ Replace RADIUS Port info with Username from this AAA Client

□ Match Framed-IP-Address with user IP address for accounting packets from this AAA Client

- AAA Client Hostname
- AAA Client IP Address
- Shared Secret
- Network Device Group
- RADIUS Key Wrap
- Authenticate Using
- Single Connect TACACS+ AAA Client
- Log Update/Watchdog Packets from this AAA Client
- Log RADIUS Tunneling Packets from this AAA Client
- Replace RADIUS Port info with Username from this AAA Client
- Match Framed-IP-Address with user IP address for accounting packets from this AAA Client

AAA Client Hostname

The AAA Client Hostname is the name assigned to the AAA client.

[Back to Top]

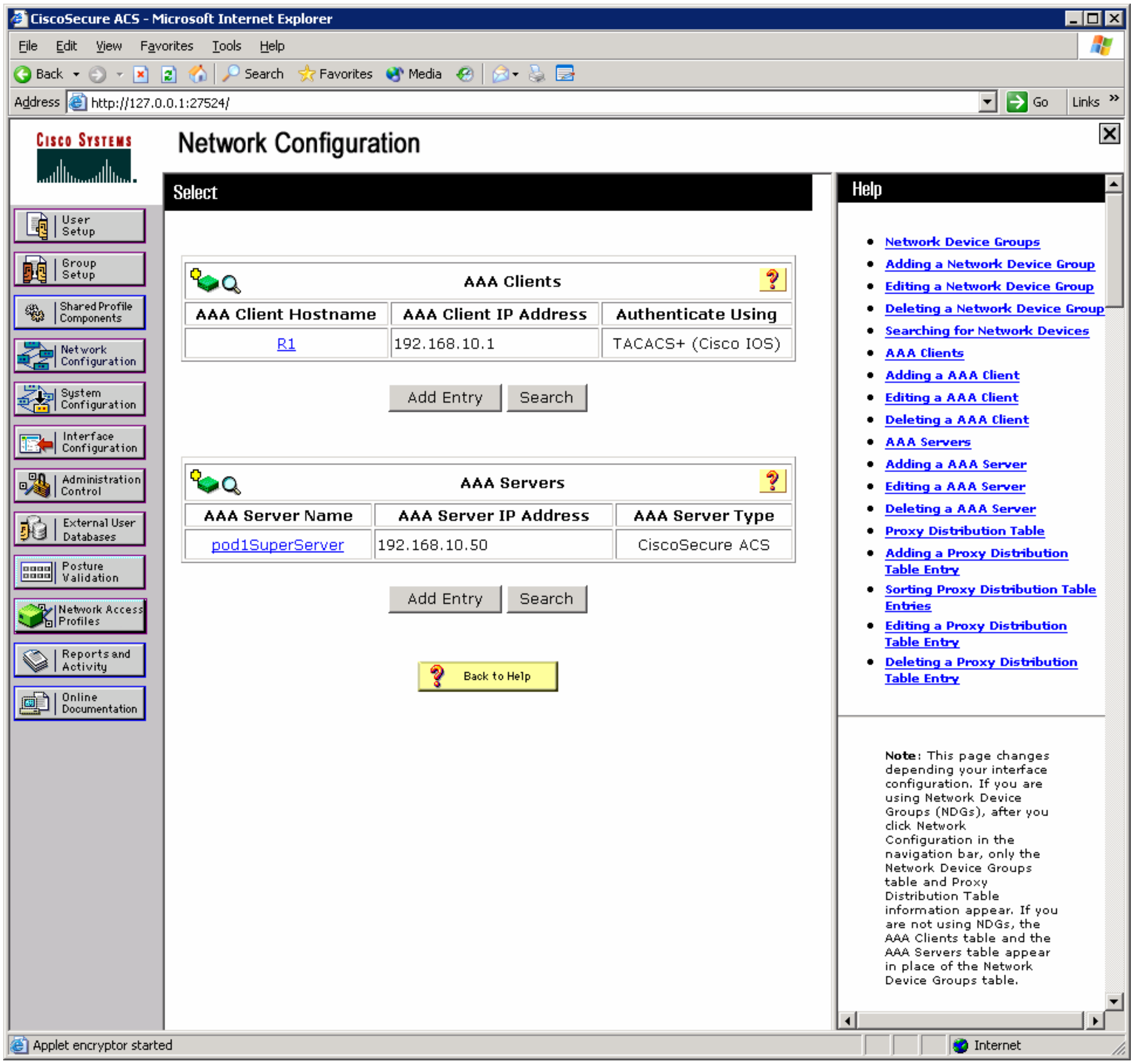
AAA Client IP Address

The AAA Client IP Address is the IP address assigned to the AAA client.

If you want to designate more than one AAA client with a single AAA client entry in ACS, you can specify the IP address for each AAA client to be represented by this AAA client entry. To separate each IP address, press **Enter**.

Figure 3-3: ACS AAA Client Configuration

You should now be able to see R1 listed as a AAA client on the network configuration screen.



Network Configuration

AAA Clients

AAA Client Hostname	AAA Client IP Address	Authenticate Using
R1	192.168.10.1	TACACS+ (Cisco IOS)

AAA Servers

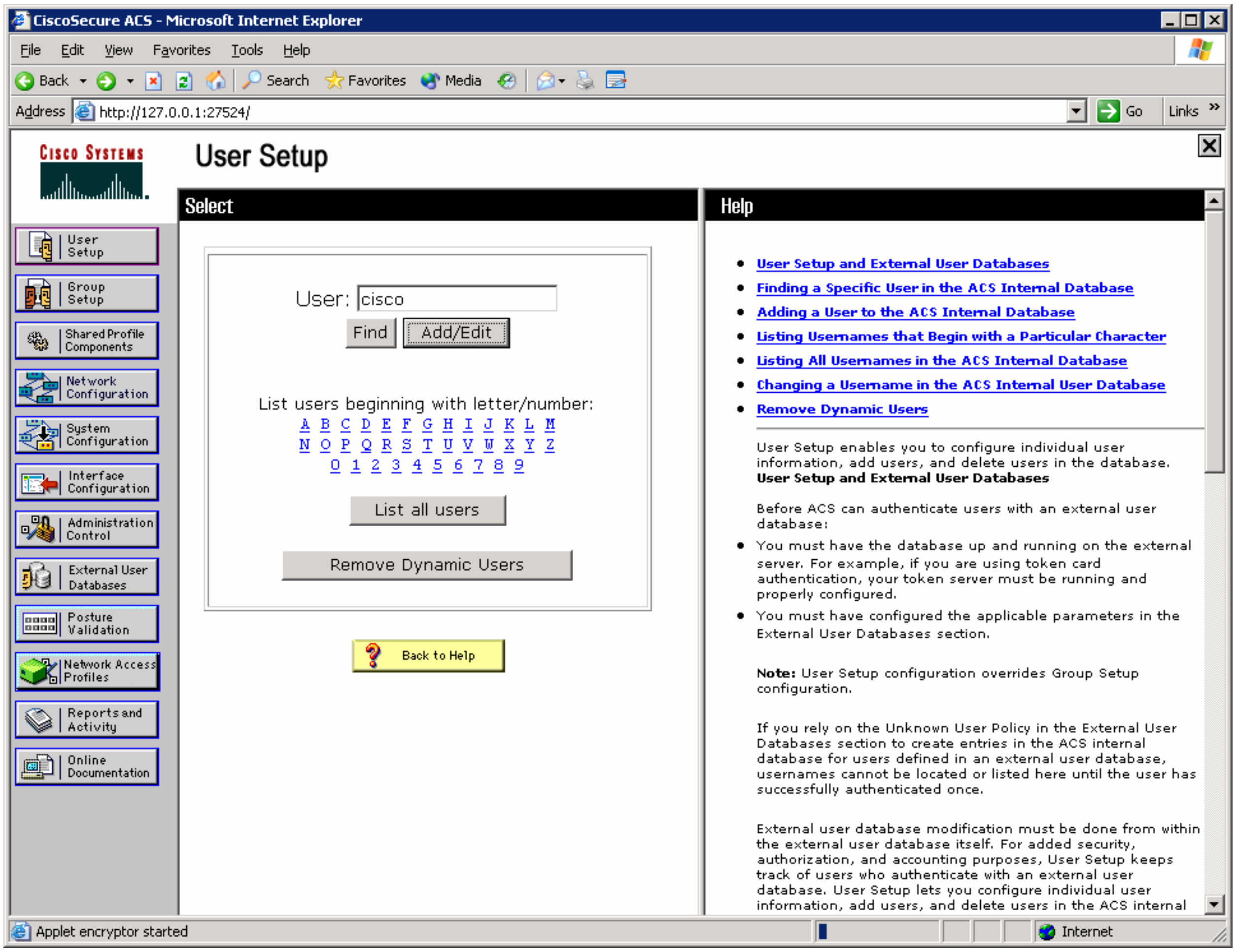
AAA Server Name	AAA Server IP Address	AAA Server Type
pod1SuperServer	192.168.10.50	CiscoSecure ACS

- Network Device Groups
- Adding a Network Device Group
- Editing a Network Device Group
- Deleting a Network Device Group
- Searching for Network Devices
- AAA Clients
- Adding a AAA Client
- Editing a AAA Client
- Deleting a AAA Client
- AAA Servers
- Adding a AAA Server
- Editing a AAA Server
- Deleting a AAA Server
- Proxy Distribution Table
- Adding a Proxy Distribution Table Entry
- Sorting Proxy Distribution Table Entries
- Editing a Proxy Distribution Table Entry
- Deleting a Proxy Distribution Table Entry

Note: This page changes depending your interface configuration. If you are using Network Device Groups (NDGs), after you click Network Configuration in the navigation bar, only the Network Device Groups table and Proxy Distribution Table information appear. If you are not using NDGs, the AAA Clients table and the AAA Servers table appear in place of the Network Device Groups table.

Figure 3-4: ACS Network Configuration Page, with Changes Applied

Click the **User Setup** button on the left side. Add a user named “cisco,” and then click **Add/Edit**.



User Setup

User: cisco

List users beginning with letter/number:

- User Setup and External User Databases
- Finding a Specific User in the ACS Internal Database
- Adding a User to the ACS Internal Database
- Listing Usernames that Begin with a Particular Character
- Listing All Usernames in the ACS Internal Database
- Changing a Username in the ACS Internal User Database
- Remove Dynamic Users

User Setup enables you to configure individual user information, add users, and delete users in the database.
User Setup and External User Databases

Before ACS can authenticate users with an external user database:

- You must have the database up and running on the external server. For example, if you are using token card authentication, your token server must be running and properly configured.
- You must have configured the applicable parameters in the External User Databases section.

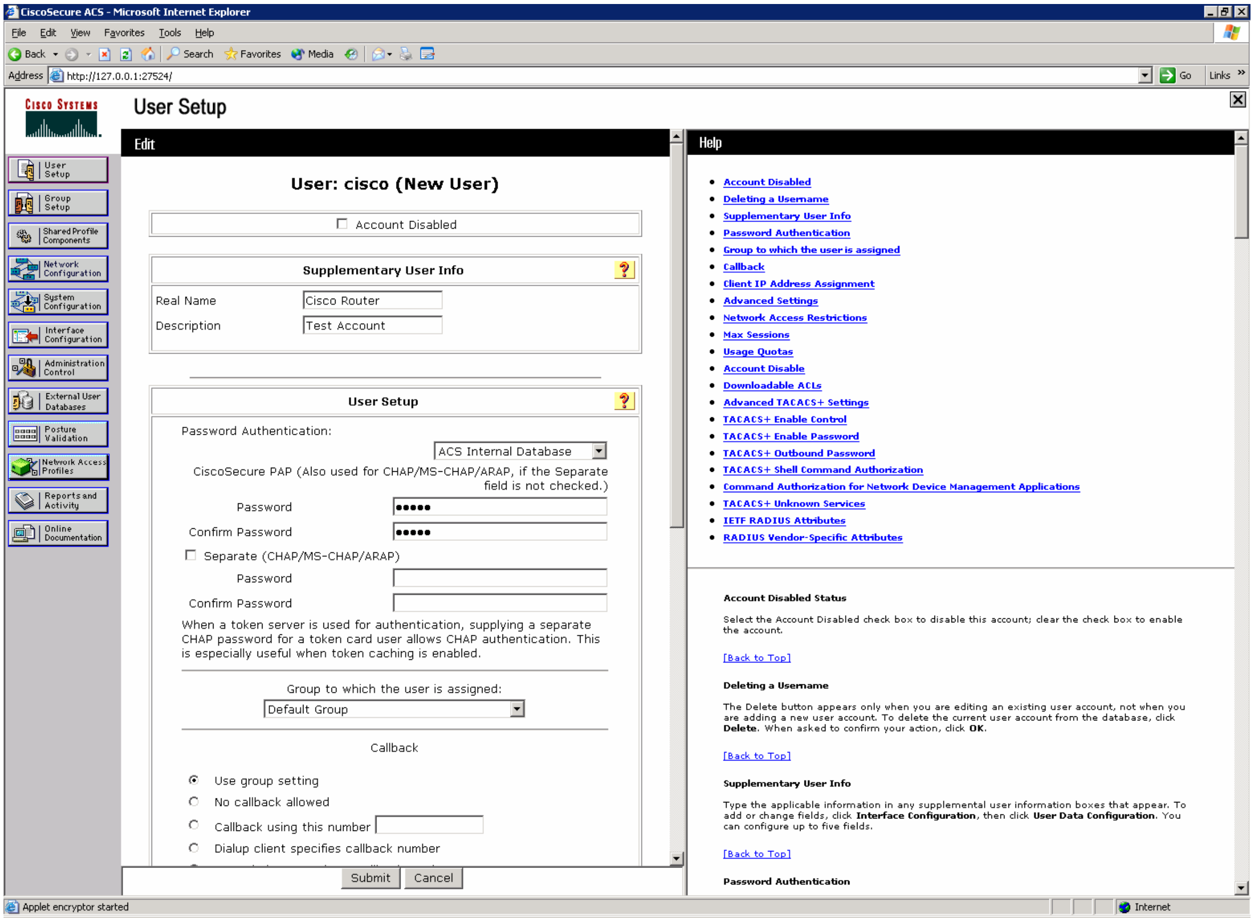
Note: User Setup configuration overrides Group Setup configuration.

If you rely on the Unknown User Policy in the External User Databases section to create entries in the ACS internal database for users defined in an external user database, usernames cannot be located or listed here until the user has successfully authenticated once.

External user database modification must be done from within the external user database itself. For added security, authorization, and accounting purposes, User Setup keeps track of users who authenticate with an external user database. User Setup lets you configure individual user information, add users, and delete users in the ACS internal

Figure 3-5: ACS User Configuration Page

Assign the real name to be your own name, and set the password to “cisco.” Click **Submit**.



User Setup

Edit

User: cisco (New User)

□ Account Disabled

Supplementary User Info

Real Name: Cisco Router

Description: Test Account

User Setup

Password Authentication: ACS Internal Database

CiscoSecure PAP (Also used for CHAP/MS-CHAP/ARAP, if the Separate field is not checked.)

Password: •••••

Confirm Password: •••••

□ Separate (CHAP/MS-CHAP/ARAP)

Password:

Confirm Password:

When a token server is used for authentication, supplying a separate CHAP password for a token card user allows CHAP authentication. This is especially useful when token caching is enabled.

Group to which the user is assigned: Default Group

Callback

● Use group setting

○ No callback allowed

○ Callback using this number

○ Dialup client specifies callback number

Submit Cancel

Help

- Account Disabled
- Deleting a Username
- Supplementary User Info
- Password Authentication
- Group to which the user is assigned
- Callback
- Client IP Address Assignment
- Advanced Settings
- Network Access Restrictions
- Max Sessions
- Usage Quotas
- Account Disable
- Downloadable ACLs
- Advanced TACACS+ Settings
- TACACS+ Enable Control
- TACACS+ Enable Password
- TACACS+ Outbound Password
- TACACS+ Shell Command Authorization
- Command Authorization for Network Device Management Applications
- TACACS+ Unknown Services
- IETF RADIUS Attributes
- RADIUS Vendor-Specific Attributes

Account Disabled Status

Select the Account Disabled check box to disable this account; clear the check box to enable the account.

[Back to Top]

Deleting a Username

The Delete button appears only when you are editing an existing user account, not when you are adding a new user account. To delete the current user account from the database, click **Delete**. When asked to confirm your action, click **OK**.

[Back to Top]

Supplementary User Info

Type the applicable information in any supplemental user information boxes that appear. To add or change fields, click **Interface Configuration**, then click **User Data Configuration**. You can configure up to five fields.

[Back to Top]

Password Authentication

Figure 3-6: ACS Add New User Page

Why would a company want to use a centralized authentication server rather than configuring users and passwords on each individual router?

Step 4: Configure AAA Services on R1

On R1, enable AAA with the **aaa new-model** command in global configuration mode. Then set up the default login authentication list with the **aaa authentication login default** *method1* [*method2*] [*method3*] command. You may create a list of authentication methods. Configure the list to first use TACACS+ for the authentication service, and then enter the **none** keyword. If no TACACS+ server can be reached and authentication cannot be performed, the router globally allows access without authentication. This is a safeguard measure in case the router starts up without connectivity to an active TACACS+

server. You could alternatively configure local authentication as the backup authentication method instead.

```
R1(config)# aaa new-model
R1(config)# aaa authentication login default group tacacs+ none
```

Note: If you do not set up a default login authentication list, you could get locked out of the router and need to use the password recovery procedure for your specific router.

Specify a TACACS+ server using the **tacacs-server host** *hostname* **key** *key* command. The *hostname* parameter accepts either a hostname or an IP address. The *key* is a secret password shared between the TACACS+ server and the AAA client and used to encrypt the TCP connection between the authenticator and the TACACS+ authentication server.

```
R1(config)# tacacs-server host 192.168.10.50 key ciscosecret
```

Next, create a unique authentication list for Telnet access to the router. This does not have the fallback of no authentication, so if there is no access to the TACACS+ server, Telnet access is disabled. To create an authentication list that is not the default list, use the global configuration command **aaa authentication login** *name method1* [*method2*] [*method3*]. Name the authentication method list "telnet_lines." To apply the list to virtual terminal lines (VTYs) on the router, issue the **login authentication** *name* command in line configuration mode.

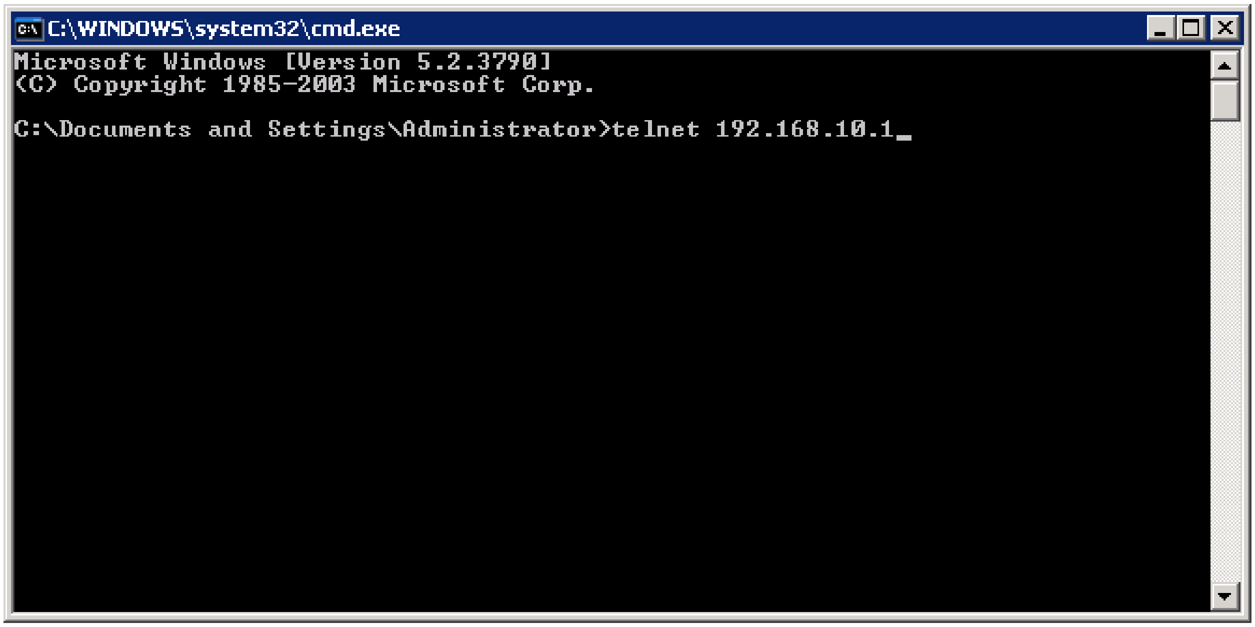
```
R1(config)# aaa authentication login telnet_lines group tacacs+
R1(config)# line vty 0 4
R1(config-line)# login authentication telnet_lines
```

Given the configuration described above, if you enter a username and password pair stored in the ACS authentication database, and the router can reach and use the authentication methods available through TACACS+, would the user be permitted to access the router?

If you enter a username and password pair not stored in the ACS authentication database and the router can reach and use the authentication methods available through TACACS+, would the user be permitted to access the router?

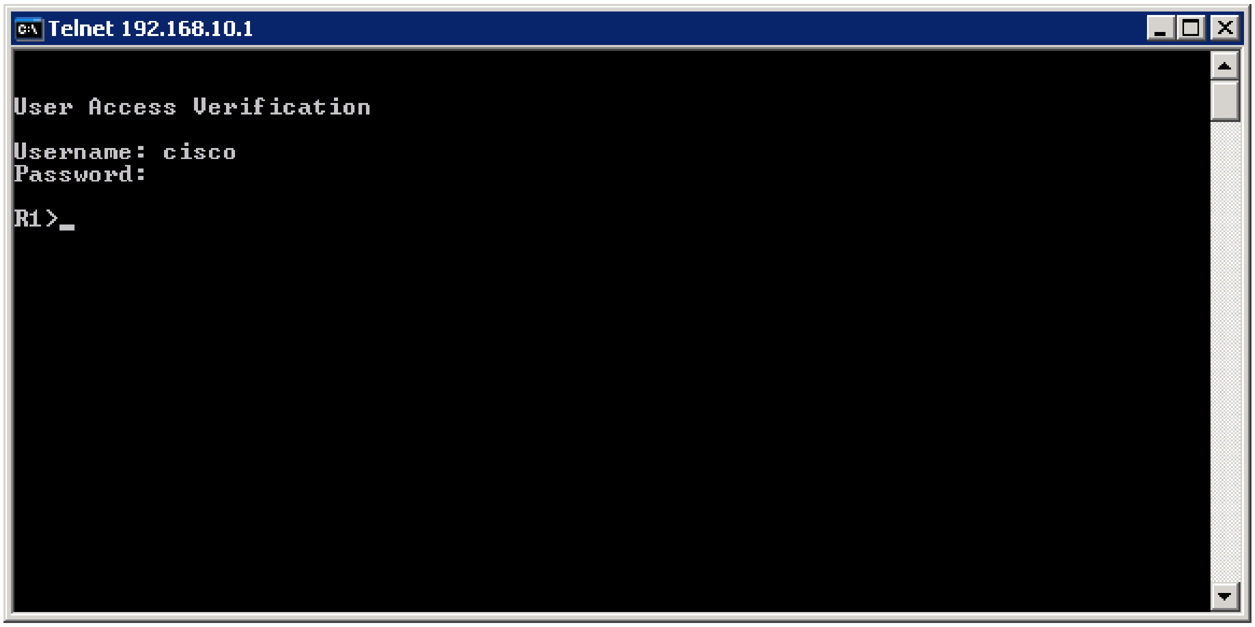
If you entered a username and password pair stored in the ACS authentication database, but the router could not reach a TACACS+ server, would the user be permitted to access the router?

You can test your configuration by opening a Telnet session from the host to R1. Click the **Start** button and choose **Run**. Enter the **cmd** command in the Run dialog box, and click **OK**. At the command prompt, issue the **telnet** *host* command. At the login prompt, use the login credentials created earlier: the username and password are both “cisco.”



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>telnet 192.168.10.1_
```

Figure 4-1: Host Telnets to R1



```
User Access Verification

Username: cisco
Password:

R1>_
```

Figure 4-2: Test AAA Authentication Using Telnet

If your session with the router console port times out, you may have to log in using the default authentication list.

Which authentication database does the current default authentication list query?

Why is it advisable to assign redundant authentication methods when using AAA?

Final Configuration

```
R1# show run
hostname R1
!
aaa new-model
!
aaa authentication login default group tacacs+ none
aaa authentication login telnet_lines group tacacs+
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
```

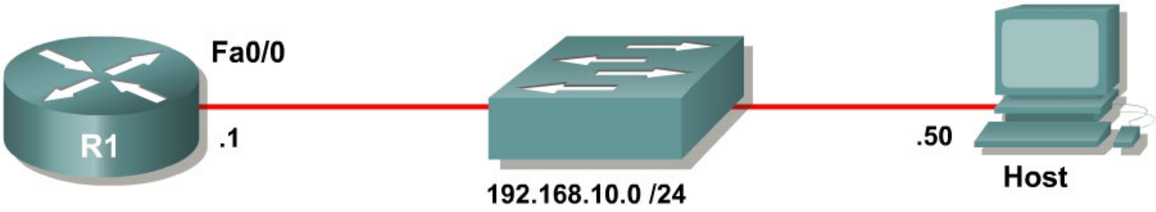
```
!
tacacs-server host 192.168.10.50 key ciscosecret
!
line vty 0 4
 login authentication telnet_lines
end
```

Lab 5.6b Configuring AAA and RADIUS

Learning Objectives

- Install CiscoSecure ACS
- Configure CiscoSecure ACS as a RADIUS server
- Enable AAA on a router using a remote RADIUS server

Topology Diagram



Scenario

In this lab, you will set up CiscoSecure ACS as a RADIUS server. You will also set up R1 to use authentication, authorization, and accounting (AAA) with reference to the RADIUS server. Because RADIUS is an open, standards-based protocol, many implementations are available. This lab shows how to configure CiscoSecure ACS. However, you could use a different RADIUS software solution. If you are using another RADIUS solution, configure the server similarly to the configuration used for ACS. The router configuration is the same regardless of the software server used.

Step 1: Configure the Interface

Configure the router interface shown in the topology diagram.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
```

Configure the IP address of 192.168.10.50/24 on the host.

Verify that you have connectivity between R1 and the host with the **ping** command.

```
R1# ping 192.168.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Step 2: Install CiscoSecure ACS

If you have already installed CiscoSecure ACS, go to step 3.

This step guides you through installing the 90-day trial version of CiscoSecure ACS. After you download the trial and extract it, run Setup.exe.

Note: At the time of this writing, CiscoSecure ACS only runs on Microsoft Windows Server Editions. You cannot run CiscoSecure ACS on Microsoft Windows XP.

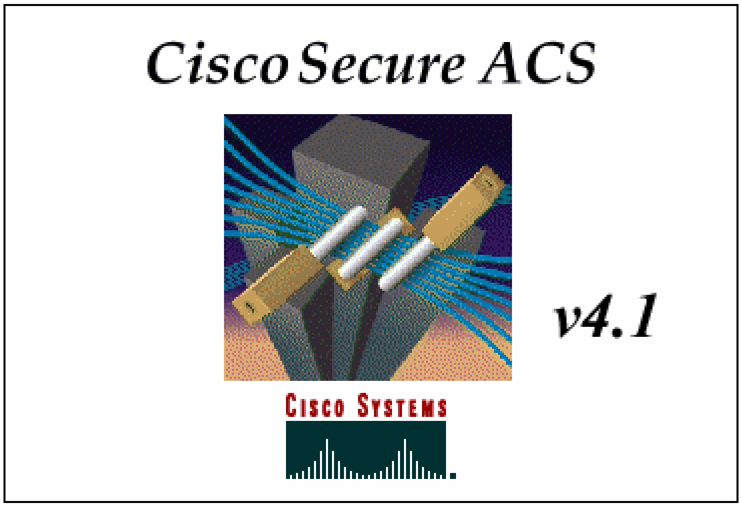
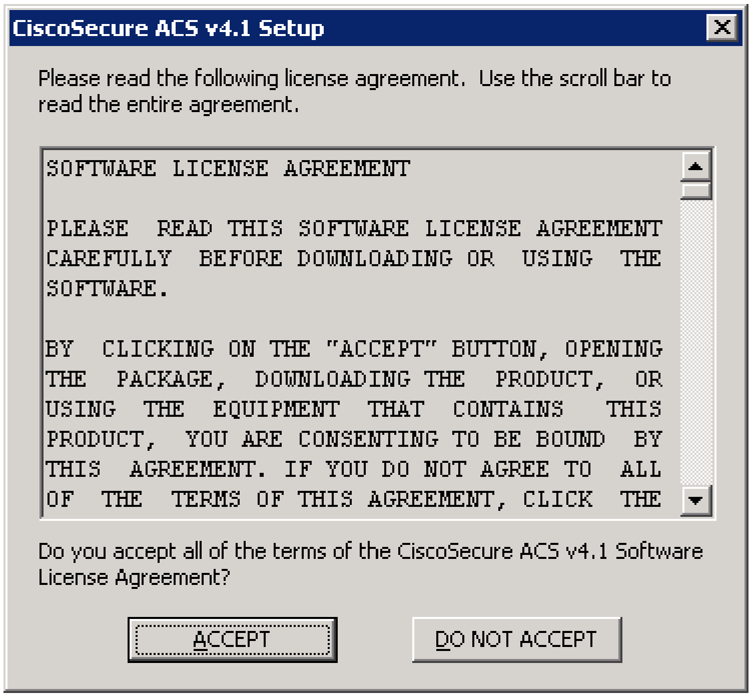


Figure 2-1: CiscoSecure ACS Splash Screen

After reading the terms of the license agreement, click **ACCEPT**.



CiscoSecure ACS v4.1 Setup

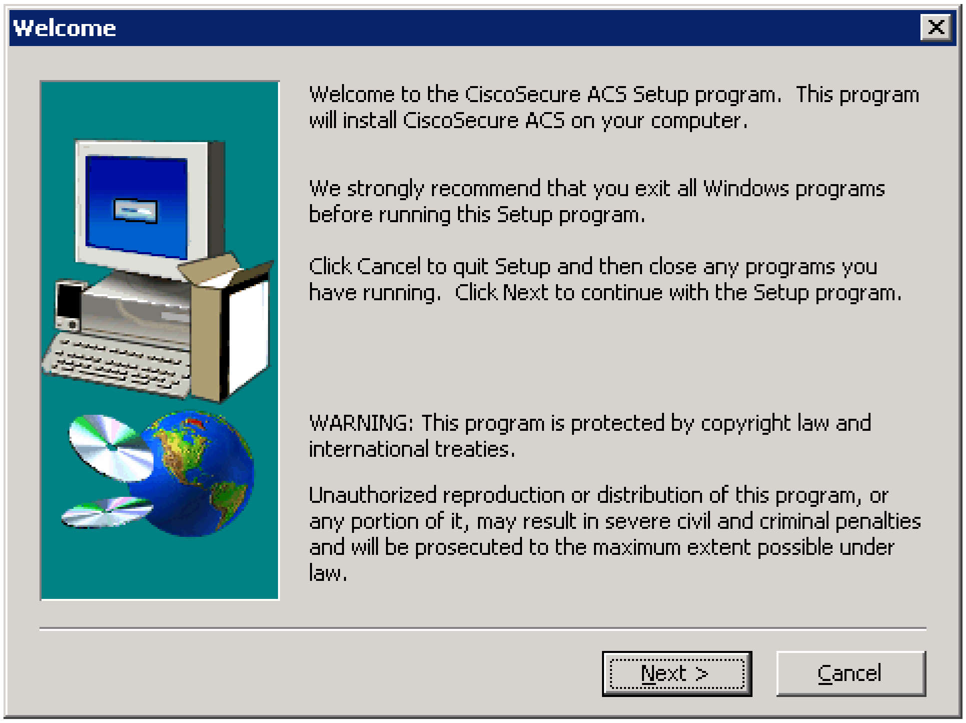
Please read the following license agreement. Use the scroll bar to read the entire agreement.

```
SOFTWARE LICENSE AGREEMENT

PLEASE  READ THIS SOFTWARE LICENSE AGREEMENT
CAREFULLY  BEFORE DOWNLOADING OR  USING  THE
SOFTWARE.

BY  CLICKING ON THE "ACCEPT" BUTTON, OPENING
THE  PACKAGE,  DOWNLOADING THE  PRODUCT,  OR
USING  THE  EQUIPMENT  THAT  CONTAINS   THIS
PRODUCT,  YOU ARE CONSENTING TO BE BOUND  BY
THIS  AGREEMENT. IF YOU DO NOT AGREE TO  ALL
OF  THE  TERMS OF THIS AGREEMENT, CLICK  THE
```

Do you accept all of the terms of the CiscoSecure ACS v4.1 Software License Agreement?

Figure 2-2: CiscoSecure ACS License Agreement

Click **Next** to continue the installation process.



Welcome

Welcome to the CiscoSecure ACS Setup program. This program will install CiscoSecure ACS on your computer.

We strongly recommend that you exit all Windows programs before running this Setup program.

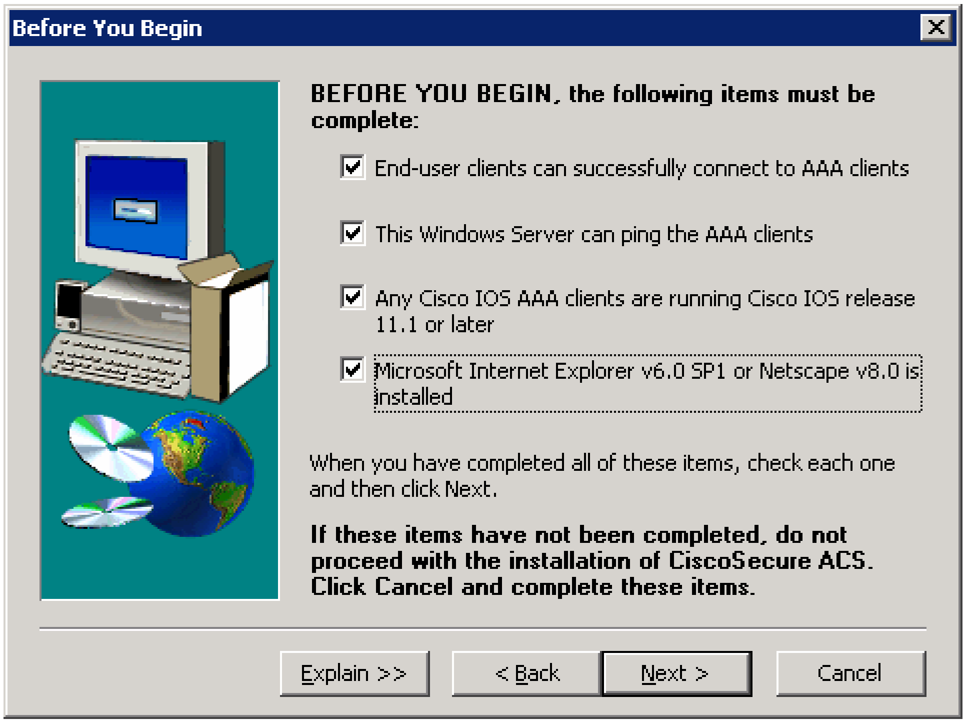
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties and will be prosecuted to the maximum extent possible under law.

Figure 2-3: CiscoSecure ACS Installation Wizard

Verify that all the requirements in the checklist are satisfied and check all the options before clicking **Next**.



Before You Begin

BEFORE YOU BEGIN, the following items must be complete:

☑ End-user clients can successfully connect to AAA clients

☑ This Windows Server can ping the AAA clients

☑ Any Cisco IOS AAA clients are running Cisco IOS release 11.1 or later

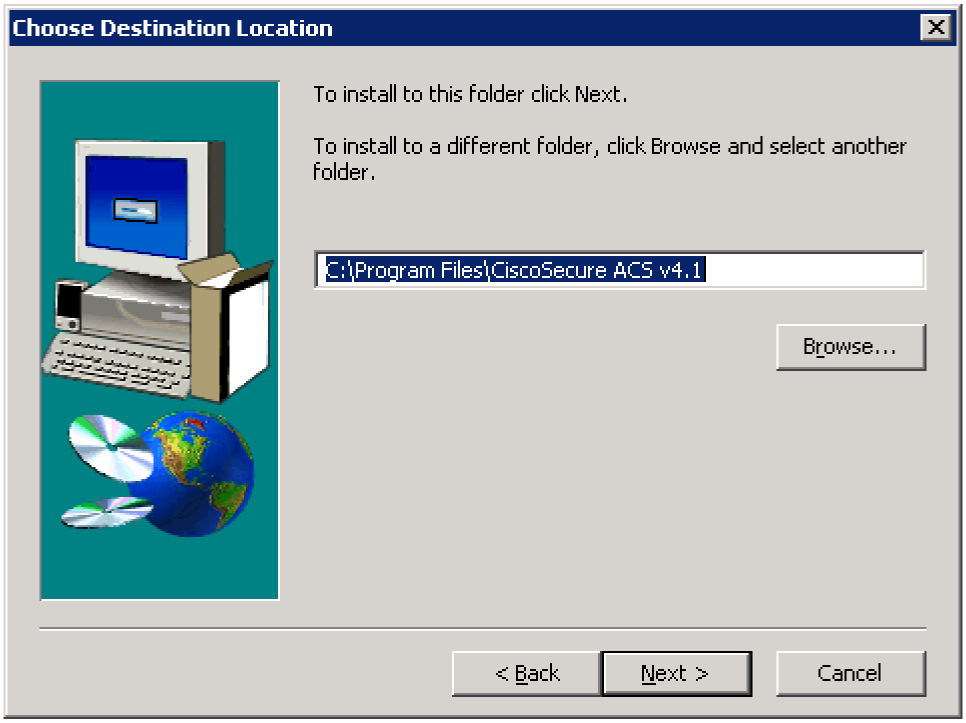
☑ Microsoft Internet Explorer v6.0 SP1 or Netscape v8.0 is installed

When you have completed all of these items, check each one and then click Next.

If these items have not been completed, do not proceed with the installation of CiscoSecure ACS. Click Cancel and complete these items.

Figure 2-4: CiscoSecure ACS Pre-Installation Checklist

Use the default installation folder and click **Next**.



Choose Destination Location

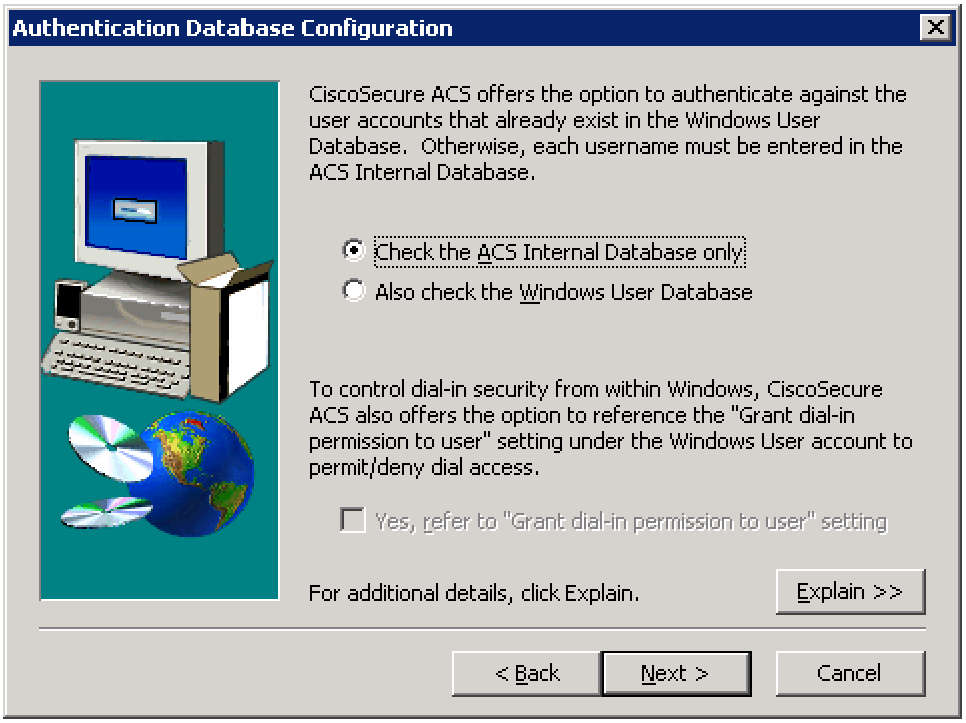
To install to this folder click Next.

To install to a different folder, click Browse and select another folder.

C:\Program Files\CiscoSecure ACS v4.1

Figure 2-5: CiscoSecure ACS Installation Location

CiscoSecure has the ability to check the Windows User Database. However, for this lab, choose to authenticate using the internal database only. Click **Next**.



Authentication Database Configuration

CiscoSecure ACS offers the option to authenticate against the user accounts that already exist in the Windows User Database. Otherwise, each username must be entered in the ACS Internal Database.

● Check the ACS Internal Database only
○ Also check the Windows User Database

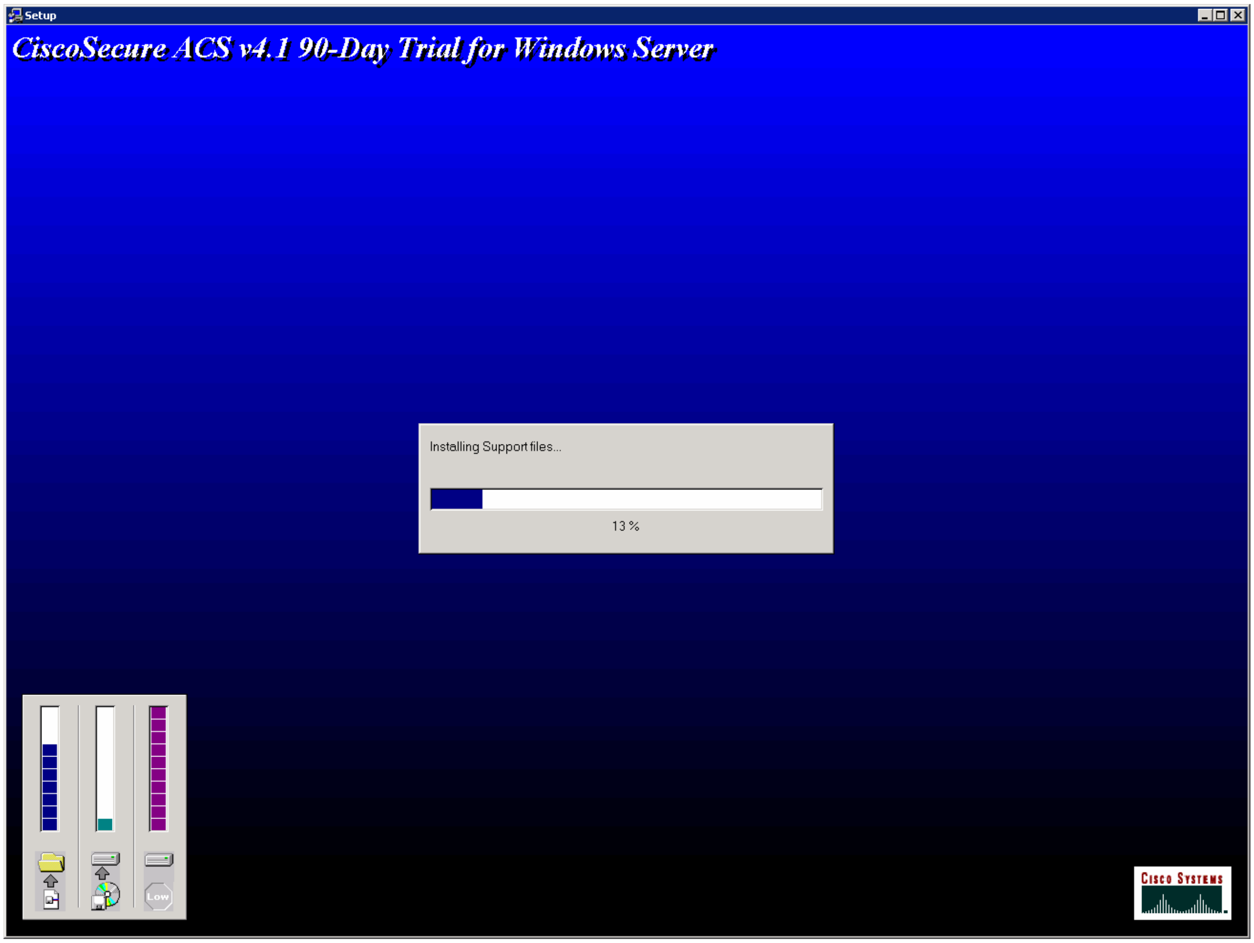
To control dial-in security from within Windows, CiscoSecure ACS also offers the option to reference the "Grant dial-in permission to user" setting under the Windows User account to permit/deny dial access.

□ Yes, refer to "Grant dial-in permission to user" setting

For additional details, click Explain.

Figure 2-6: CiscoSecure ACS Authentication Database Options

The installer will then begin copying files and registry keys. This process may take a few minutes.

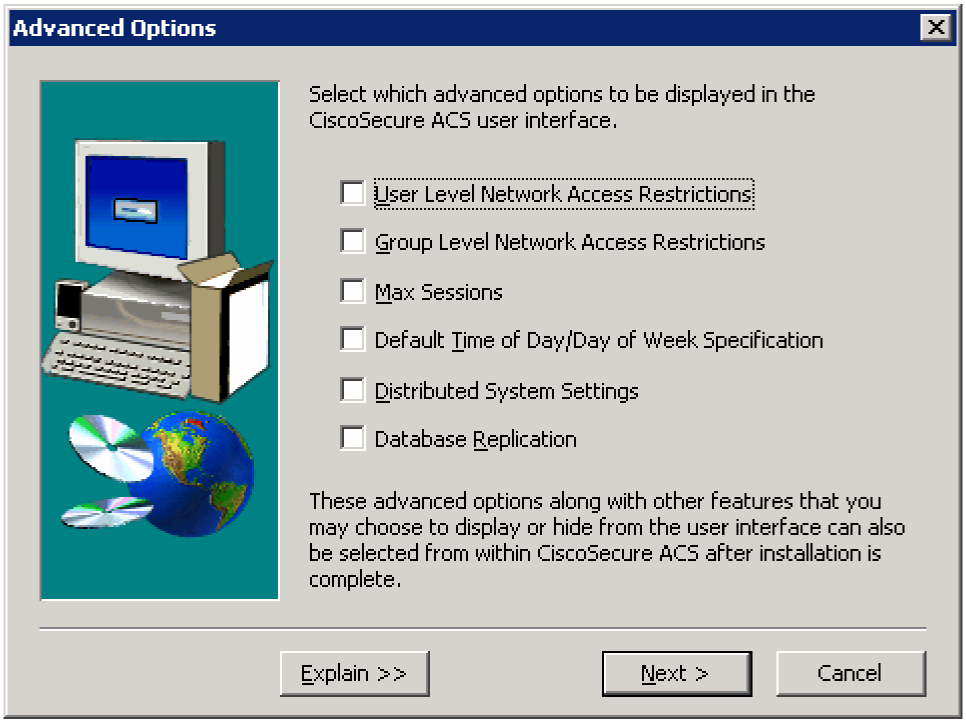


Installing Support files...

13 %

Figure 2-7: CiscoSecure ACS Installation Progress Indicator

At the end of the installer, you are prompted whether you want to see any advanced configuration options in the user interface. You do not need to select any of these. Click **Next** after reviewing the options.



Advanced Options

Select which advanced options to be displayed in the CiscoSecure ACS user interface.

□ User Level Network Access Restrictions

□ Group Level Network Access Restrictions

□ Max Sessions

□ Default Time of Day/Day of Week Specification

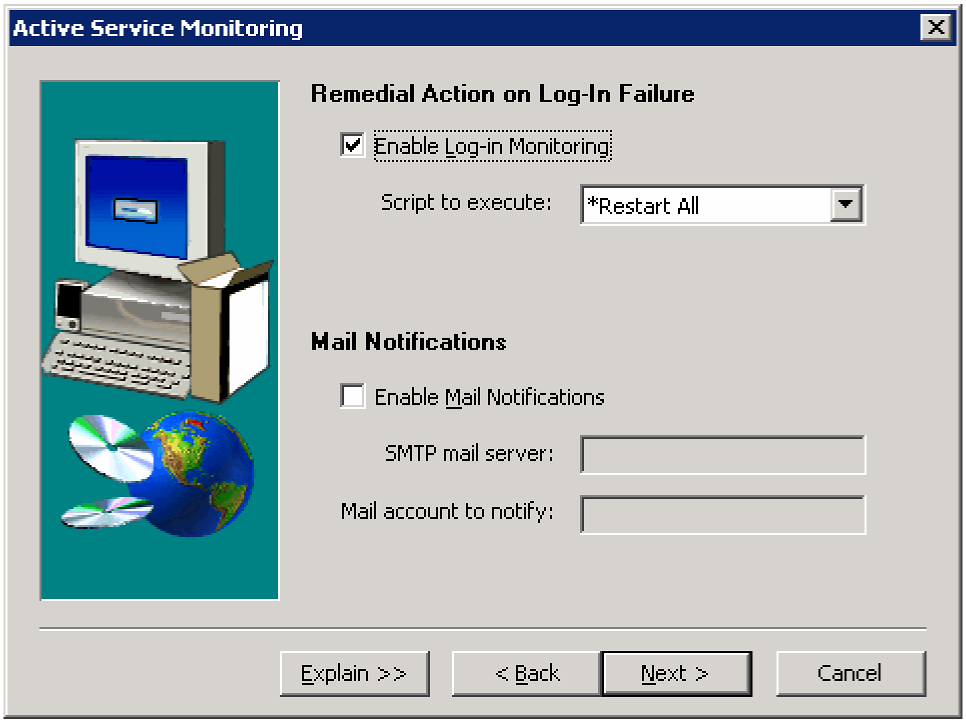
□ Distributed System Settings

□ Database Replication

These advanced options along with other features that you may choose to display or hide from the user interface can also be selected from within CiscoSecure ACS after installation is complete.

Figure 2-8: CiscoSecure ACS Advanced Configuration Options

Keep the default settings in the next step of the installation wizard and click **Next**.



Active Service Monitoring

Remedial Action on Log-In Failure

☑ Enable Log-in Monitoring

Script to execute: *Restart All

Mail Notifications

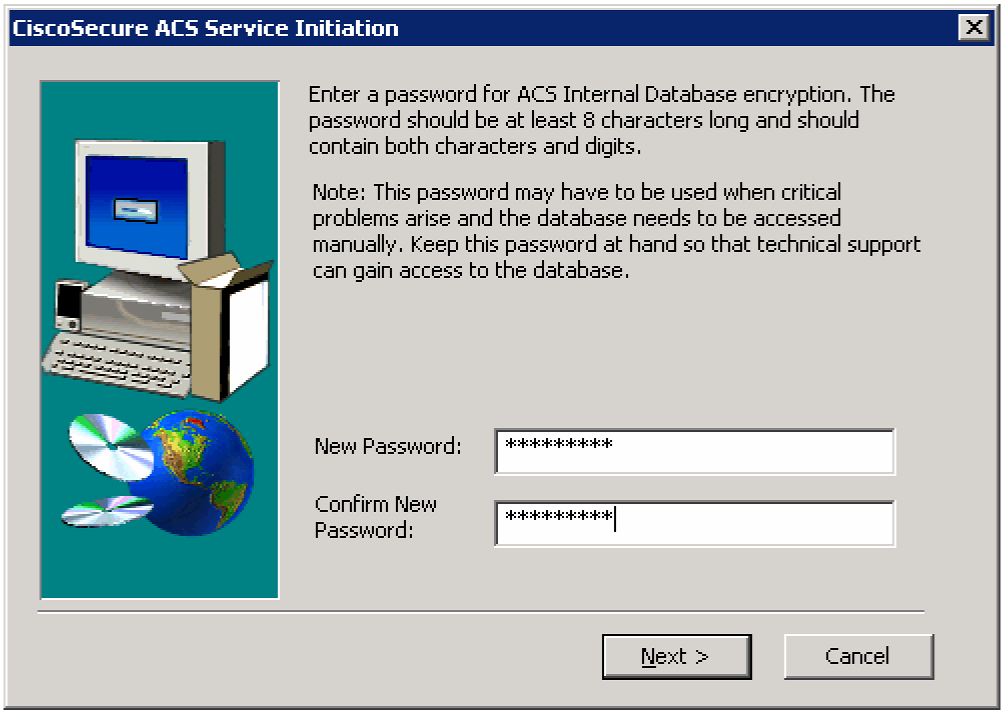
□ Enable Mail Notifications

SMTP mail server:

Mail account to notify:

Figure 2-9: CiscoSecure ACS Log-In

You must create a password for ACS internal database encryption. It must be at least eight characters and contain both letters and numbers. In the example below, “ciscoacs4” is used as the password. After configuring the password, click **Next**.



CiscoSecure ACS Service Initiation

Enter a password for ACS Internal Database encryption. The password should be at least 8 characters long and should contain both characters and digits.

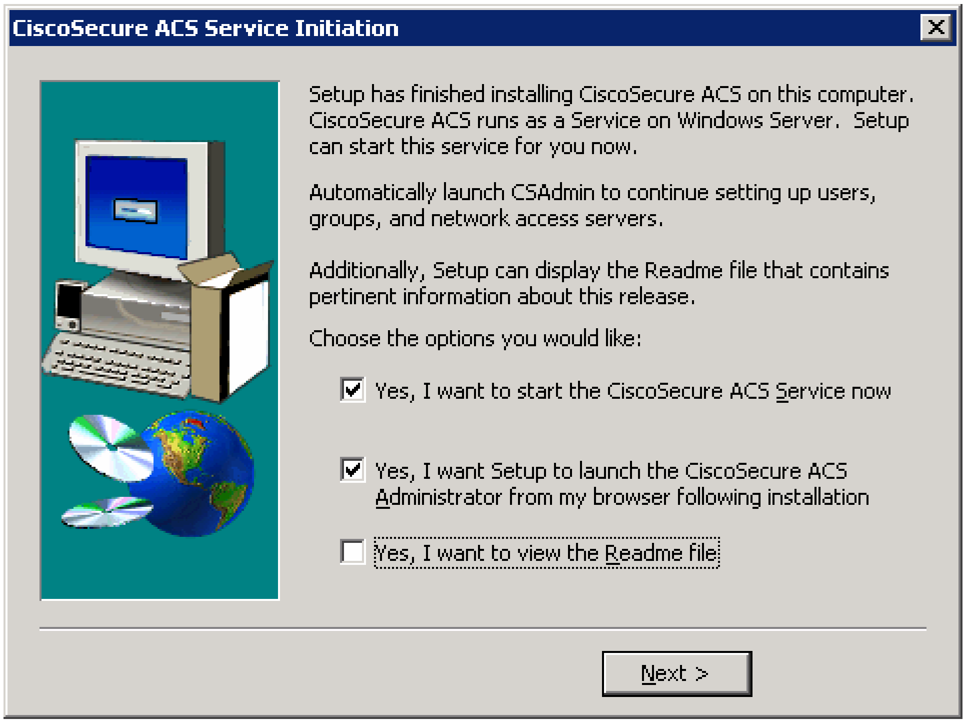
Note: This password may have to be used when critical problems arise and the database needs to be accessed manually. Keep this password at hand so that technical support can gain access to the database.

New Password: *********

Confirm New Password: *********

Figure 2-9: CiscoSecure ACS Password Configuration

Choose to start the ACS service on the host now. You should also select the option to start the administration window after the installer ends to verify the installation. Click **Next** after selecting the options.


CiscoSecure ACS Service Initiation

Setup has finished installing CiscoSecure ACS on this computer. CiscoSecure ACS runs as a Service on Windows Server. Setup can start this service for you now.

Automatically launch CSAdmin to continue setting up users, groups, and network access servers.

Additionally, Setup can display the Readme file that contains pertinent information about this release.

Choose the options you would like:

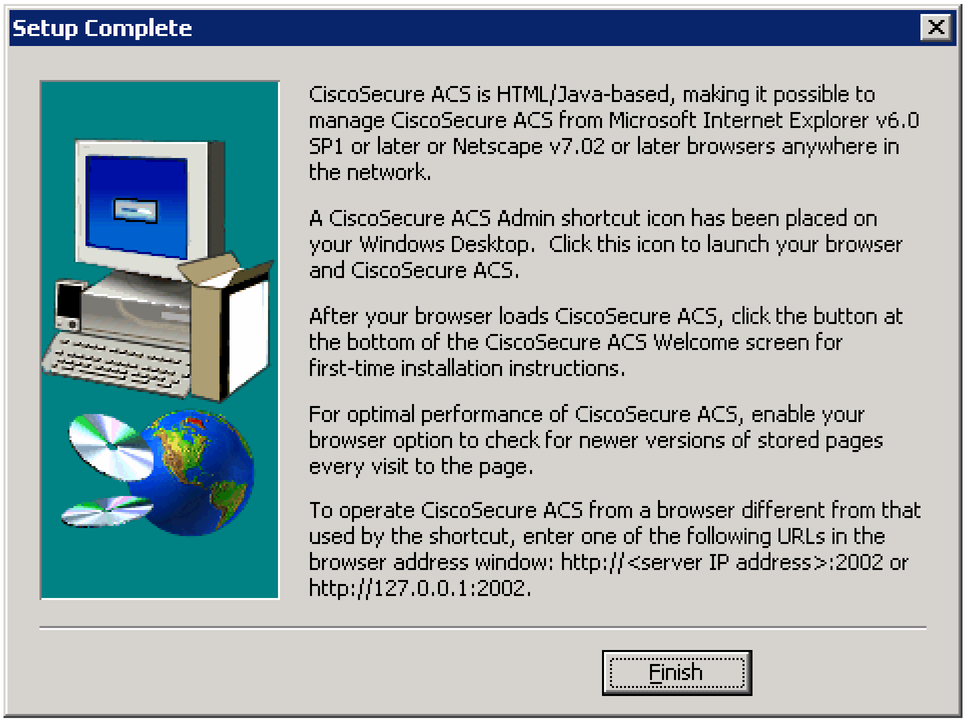
☑ Yes, I want to start the CiscoSecure ACS Service now

☑ Yes, I want Setup to launch the CiscoSecure ACS Administrator from my browser following installation

□ Yes, I want to view the Readme file

Figure 2-10: CiscoSecure ACS Service Configuration

Read the instructions and click **Finish**. You should also make sure your computer is compliant with all ACS access requirements, complying with the supported versions of Internet Explorer and the Java Runtime Environment.


Setup Complete

CiscoSecure ACS is HTML/Java-based, making it possible to manage CiscoSecure ACS from Microsoft Internet Explorer v6.0 SP1 or later or Netscape v7.02 or later browsers anywhere in the network.

A CiscoSecure ACS Admin shortcut icon has been placed on your Windows Desktop. Click this icon to launch your browser and CiscoSecure ACS.

After your browser loads CiscoSecure ACS, click the button at the bottom of the CiscoSecure ACS Welcome screen for first-time installation instructions.

For optimal performance of CiscoSecure ACS, enable your browser option to check for newer versions of stored pages every visit to the page.

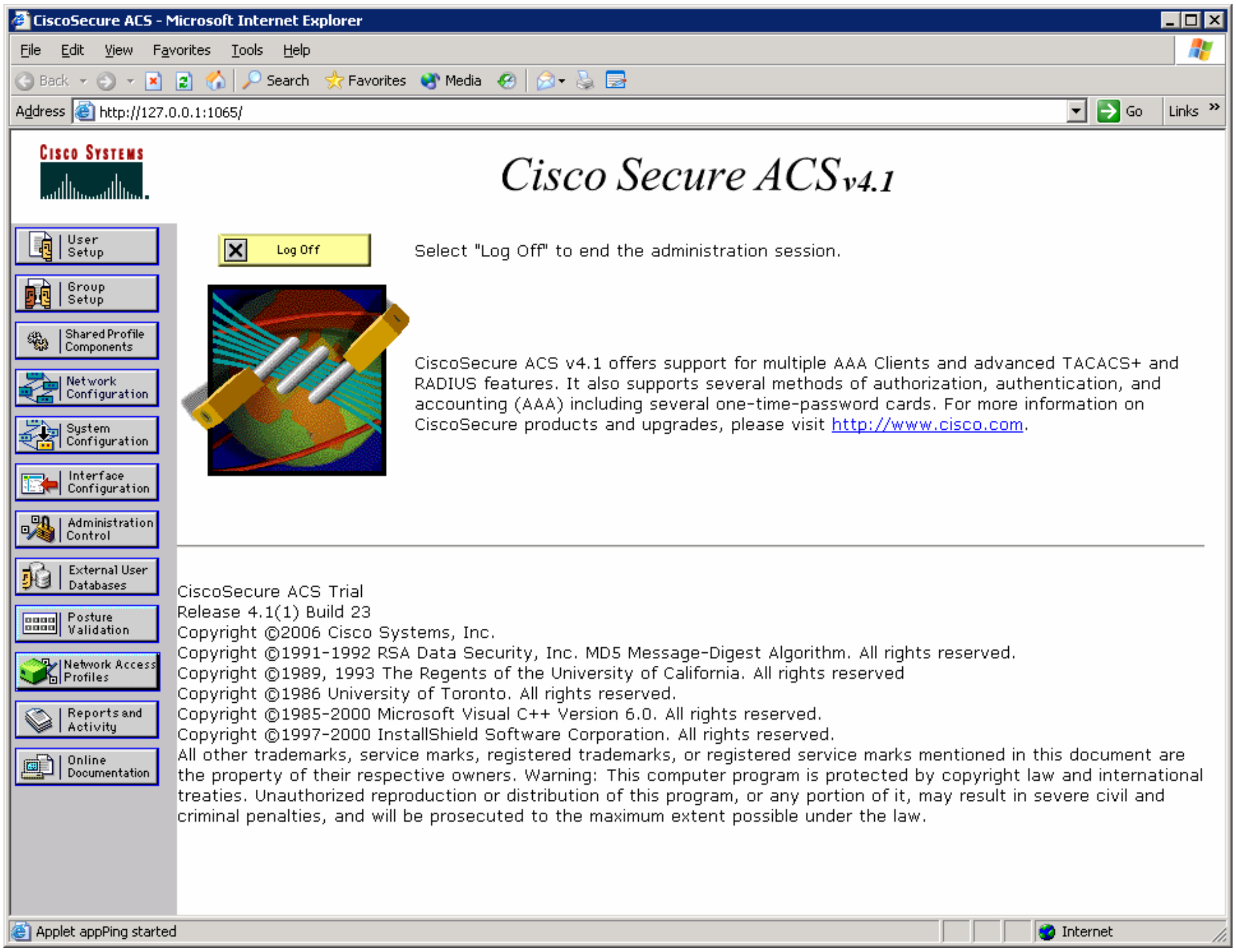
To operate CiscoSecure ACS from a browser different from that used by the shortcut, enter one of the following URLs in the browser address window: http://<server IP address>:2002 or http://127.0.0.1:2002.

Figure 2-11: CiscoSecure ACS Installation Complete Window

If the CiscoSecure ACS administrative screen comes up when the installer ends, it was successfully installed.

Step 3: Configure Users in CiscoSecure ACS

If CiscoSecure ACS application is not open, start it by clicking the **Start** button and choosing **Programs > CiscoSecure ACS v4.1 Trial > ACS Admin**.



Select "Log Off" to end the administration session.

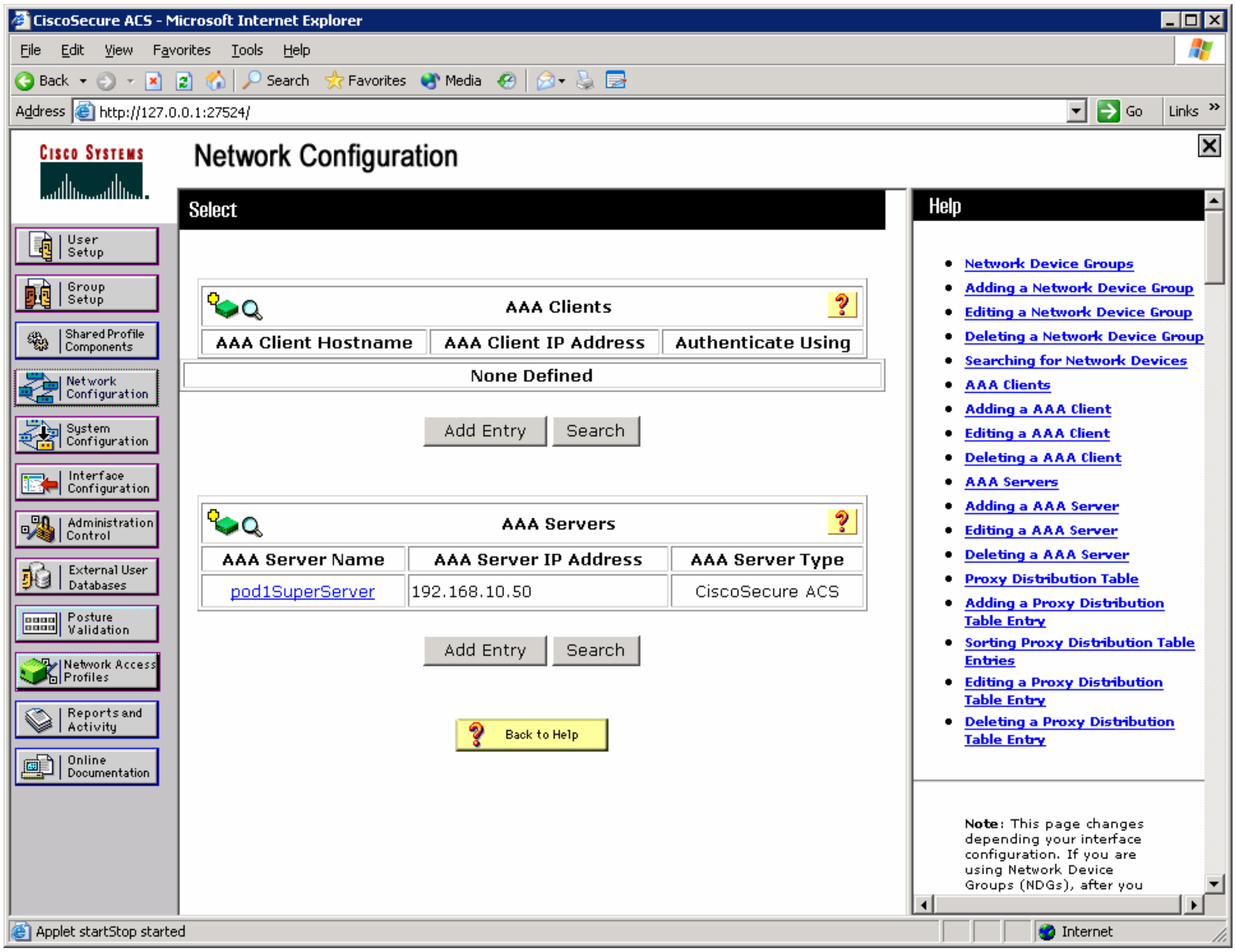
CiscoSecure ACS v4.1 offers support for multiple AAA Clients and advanced TACACS+ and RADIUS features. It also supports several methods of authorization, authentication, and accounting (AAA) including several one-time-password cards. For more information on CiscoSecure products and upgrades, please visit http://www.cisco.com.

CiscoSecure ACS Trial
Release 4.1(1) Build 23
Copyright ©2006 Cisco Systems, Inc.
Copyright ©1991-1992 RSA Data Security, Inc. MD5 Message-Digest Algorithm. All rights reserved.
Copyright ©1989, 1993 The Regents of the University of California. All rights reserved
Copyright ©1986 University of Toronto. All rights reserved.
Copyright ©1985-2000 Microsoft Visual C++ Version 6.0. All rights reserved.
Copyright ©1997-2000 InstallShield Software Corporation. All rights reserved.
All other trademarks, service marks, registered trademarks, or registered service marks mentioned in this document are the property of their respective owners. Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Applet appPing started

Figure 3-1: ACS Home Page

Click the **Network Configuration** button on the left side. On this screen, you can configure AAA clients directly. Click **Add Entry** under the heading AAA Clients.



Network Configuration

Select

AAA Clients

AAA Client Hostname	AAA Client IP Address	Authenticate Using
None Defined		

Add Entry Search

AAA Servers

AAA Server Name	AAA Server IP Address	AAA Server Type
pod1SuperServer	192.168.10.50	CiscoSecure ACS

Add Entry Search

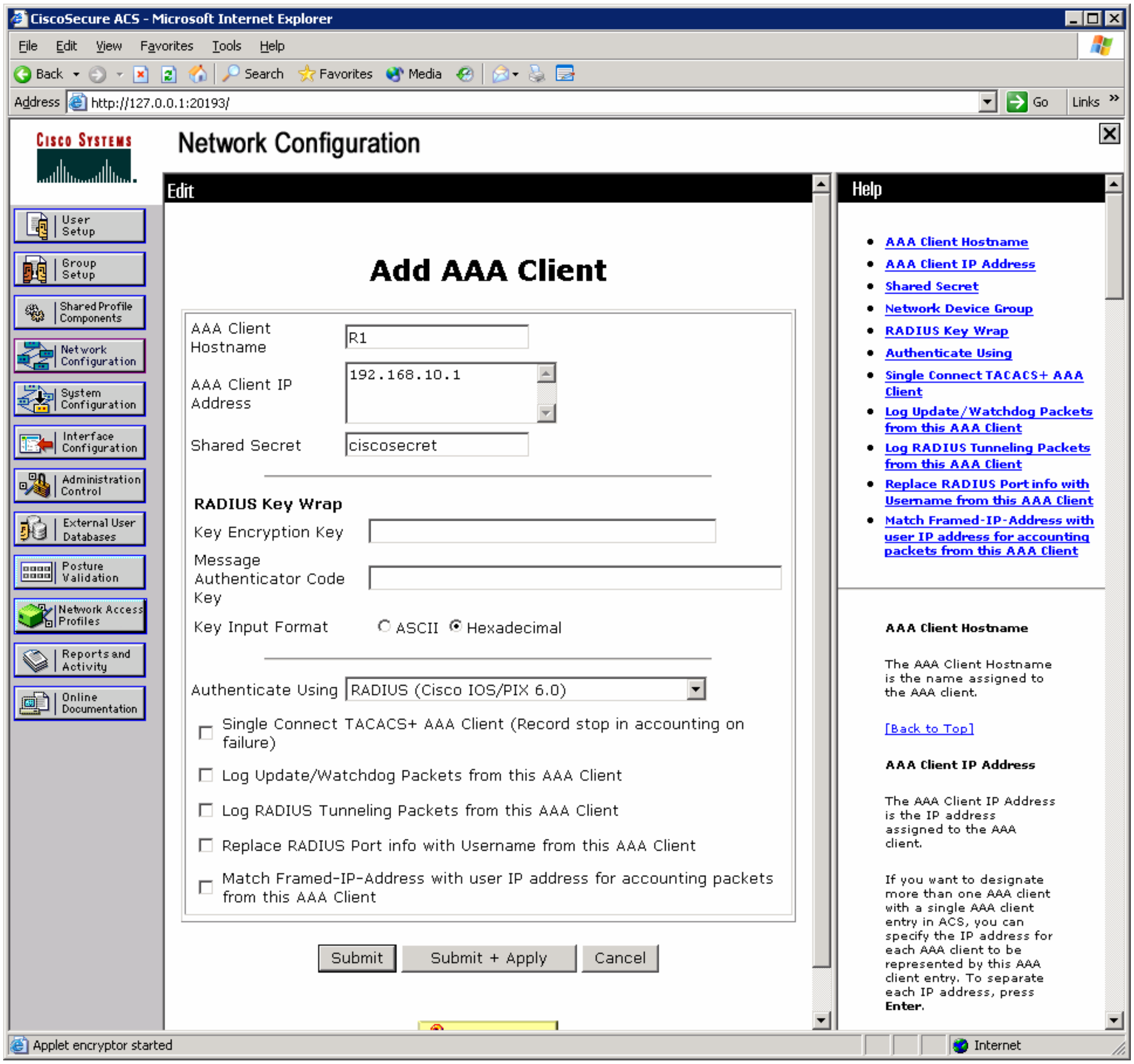
Back to Help

Help

Note: This page changes depending your interface configuration. If you are using Network Device Groups (NDGs), after you

Figure 3-2: ACS Network Configuration Page

Configure R1 as a RADIUS client as shown below, and then click **Submit + Apply**.



Network Configuration

Edit

Add AAA Client

AAA Client Hostname	R1
AAA Client IP Address	192.168.10.1
Shared Secret	ciscosecret

RADIUS Key Wrap

Key Encryption Key	
Message Authenticator Code Key	
Key Input Format	○ ASCII ● Hexadecimal

Authenticate Using RADIUS (Cisco IOS/PIX 6.0)

- ☐ Single Connect TACACS+ AAA Client (Record stop in accounting on failure)
- ☐ Log Update/Watchdog Packets from this AAA Client
- ☐ Log RADIUS Tunneling Packets from this AAA Client
- ☐ Replace RADIUS Port info with Username from this AAA Client
- ☐ Match Framed-IP-Address with user IP address for accounting packets from this AAA Client

Help

- AAA Client Hostname
- AAA Client IP Address
- Shared Secret
- Network Device Group
- RADIUS Key Wrap
- Authenticate Using
- Single Connect TACACS+ AAA Client
- Log Update/Watchdog Packets from this AAA Client
- Log RADIUS Tunneling Packets from this AAA Client
- Replace RADIUS Port info with Username from this AAA Client
- Match Framed-IP-Address with user IP address for accounting packets from this AAA Client

AAA Client Hostname

The AAA Client Hostname is the name assigned to the AAA client.

[Back to Top]

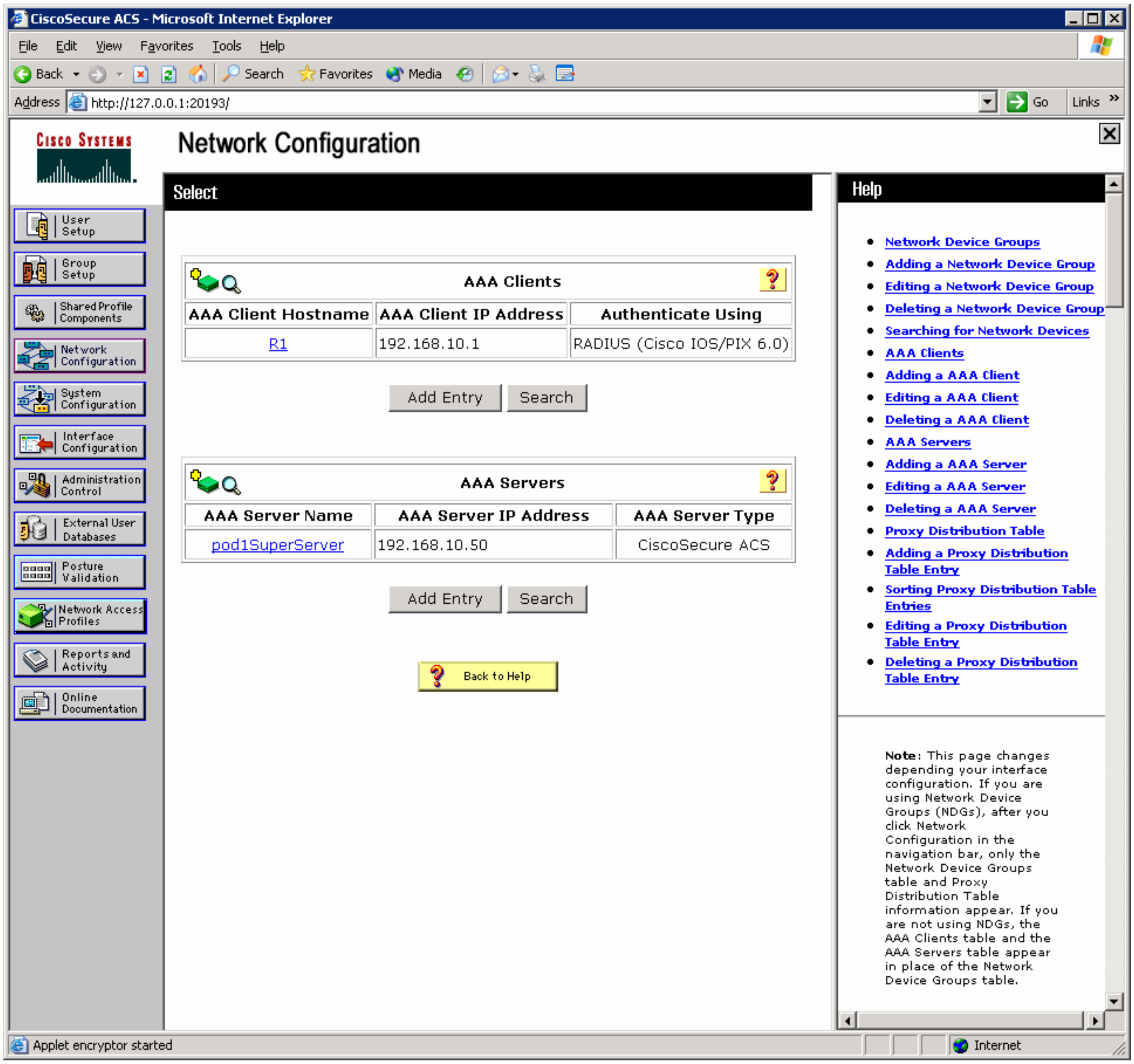
AAA Client IP Address

The AAA Client IP Address is the IP address assigned to the AAA client.

If you want to designate more than one AAA client with a single AAA client entry in ACS, you can specify the IP address for each AAA client to be represented by this AAA client entry. To separate each IP address, press **Enter.**

Figure 3-3: ACS AAA Client Configuration

You should now be able to see R1 listed as a AAA client on the network configuration screen.



Network Configuration

AAA Clients		
AAA Client Hostname	AAA Client IP Address	Authenticate Using
R1	192.168.10.1	RADIUS (Cisco IOS/PIX 6.0)

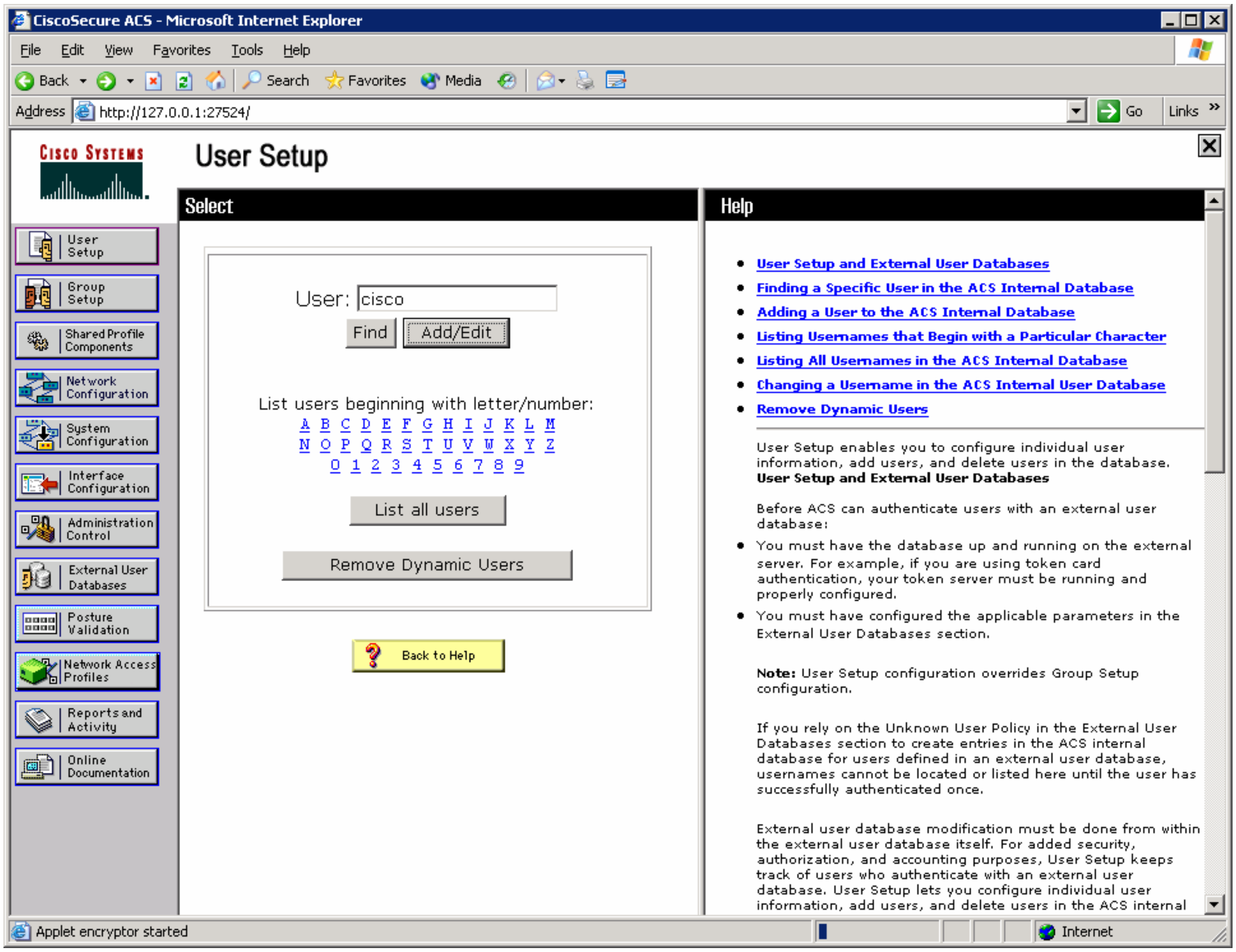
AAA Servers		
AAA Server Name	AAA Server IP Address	AAA Server Type
pod1SuperServer	192.168.10.50	CiscoSecure ACS

- Network Device Groups
- Adding a Network Device Group
- Editing a Network Device Group
- Deleting a Network Device Group
- Searching for Network Devices
- AAA Clients
- Adding a AAA Client
- Editing a AAA Client
- Deleting a AAA Client
- AAA Servers
- Adding a AAA Server
- Editing a AAA Server
- Deleting a AAA Server
- Proxy Distribution Table
- Adding a Proxy Distribution Table Entry
- Sorting Proxy Distribution Table Entries
- Editing a Proxy Distribution Table Entry
- Deleting a Proxy Distribution Table Entry

Note: This page changes depending your interface configuration. If you are using Network Device Groups (NDGs), after you click Network Configuration in the navigation bar, only the Network Device Groups table and Proxy Distribution Table information appear. If you are not using NDGs, the AAA Clients table and the AAA Servers table appear in place of the Network Device Groups table.

Figure 3-4: ACS Network Configuration Page, with Changes Applied

Click the **User Setup** button on the left side. Add a user named “cisco,” and then click **Add/Edit**.



User Setup

User: cisco

List users beginning with letter/number:
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9

- User Setup and External User Databases
- Finding a Specific User in the ACS Internal Database
- Adding a User to the ACS Internal Database
- Listing Usernames that Begin with a Particular Character
- Listing All Usernames in the ACS Internal Database
- Changing a Username in the ACS Internal User Database
- Remove Dynamic Users

User Setup enables you to configure individual user information, add users, and delete users in the database.
User Setup and External User Databases

Before ACS can authenticate users with an external user database:

- You must have the database up and running on the external server. For example, if you are using token card authentication, your token server must be running and properly configured.
- You must have configured the applicable parameters in the External User Databases section.

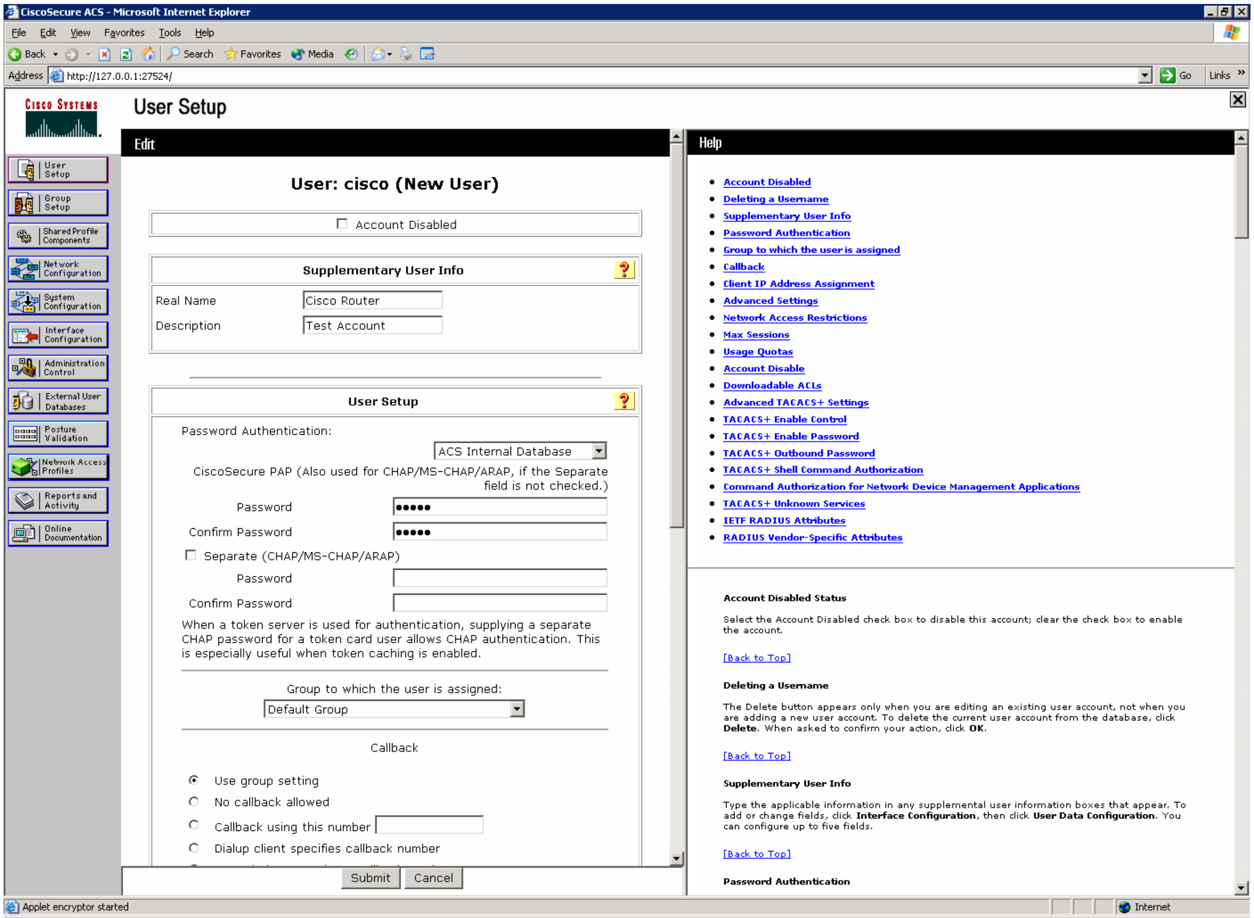
Note: User Setup configuration overrides Group Setup configuration.

If you rely on the Unknown User Policy in the External User Databases section to create entries in the ACS internal database for users defined in an external user database, usernames cannot be located or listed here until the user has successfully authenticated once.

External user database modification must be done from within the external user database itself. For added security, authorization, and accounting purposes, User Setup keeps track of users who authenticate with an external user database. User Setup lets you configure individual user information, add users, and delete users in the ACS internal

Figure 3-5: ACS User Configuration Page

Assign the real name to be your own name, and set the password to “cisco.” Click **Submit**.



User Setup

User: cisco (New User)

□ Account Disabled

Supplementary User Info

Real Name: Cisco Router
Description: Test Account

User Setup

Password Authentication:
ACS Internal Database

CiscoSecure PAP (Also used for CHAP/MS-CHAP/ARAP, if the Separate field is not checked.)

Password: •••••
Confirm Password: •••••

□ Separate (CHAP/MS-CHAP/ARAP)

Password:
Confirm Password:

When a token server is used for authentication, supplying a separate CHAP password for a token card user allows CHAP authentication. This is especially useful when token caching is enabled.

Group to which the user is assigned:
Default Group

Callback

● Use group setting
○ No callback allowed
○ Callback using this number
○ Dialup client specifies callback number

- Account Disabled
- Deleting a Username
- Supplementary User Info
- Password Authentication
- Group to which the user is assigned
- Callback
- Client IP Address Assignment
- Advanced Settings
- Network Access Restrictions
- Max Sessions
- Usage Quotas
- Account Disable
- Downloadable ACLs
- Advanced TACACS+ Settings
- TACACS+ Enable Control
- TACACS+ Enable Password
- TACACS+ Outbound Password
- TACACS+ Shell Command Authorization
- Command Authorization for Network Device Management Applications
- TACACS+ Unknown Services
- IETF RADIUS Attributes
- RADIUS Vendor-Specific Attributes

Account Disabled Status

Select the Account Disabled check box to disable this account; clear the check box to enable the account.

[Back to Top]

Deleting a Username

The Delete button appears only when you are editing an existing user account, not when you are adding a new user account. To delete the current user account from the database, click **Delete**. When asked to confirm your action, click **OK**.

[Back to Top]

Supplementary User Info

Type the applicable information in any supplemental user information boxes that appear. To add or change fields, click **Interface Configuration**, then click **User Data Configuration**. You can configure up to five fields.

[Back to Top]

Password Authentication

Figure 3-6: ACS Add New User Page

Why would a company want to use a centralized authentication server rather than configuring users and passwords on each individual router?

Step 4: Configure AAA Services on R1

On R1, enable AAA with the **aaa new-model** command in global configuration mode. Then set up the default login authentication list with the **aaa authentication login default** *method1* [*method2*] [*method3*] command. You may create a list of authentication methods. Configure the list to first use RADIUS for the authentication service, and then enter the **none** keyword. If no RADIUS server can be reached and authentication cannot be performed, the router globally allows access without authentication. This is a safeguard measure in case the router starts up without connectivity to an active RADIUS server. You could alternatively configure local authentication as the backup authentication method instead.

```
R1(config)# aaa new-model
R1(config)# aaa authentication login default group radius none
```

Note: If you do not set up a default login authentication list, you could get locked out of the router and need to use the password recovery procedure for your specific router.

Specify a RADIUS server using the **radius-server host** *hostname* **key** *key* command. The *hostname* parameter accepts either a hostname or an IP address. The *key* is a secret password shared between the RADIUS server and the AAA client and used to authenticate the connection between the router and the server before the user authentication process takes place.

```
R1(config)# radius-server host 192.168.10.50 key ciscosecret
```

Next, create a unique authentication list for Telnet access to the router. This does not have the fallback of no authentication, so if there is no access to the RADIUS server, Telnet access is disabled. To create an authentication list that is not the default list, use the global configuration command **aaa authentication login** *name method1* [*method2*] [*method3*]. Name the authentication method list "telnet_lines." To apply the list to vtys on the router, issue the **login authentication** *name* command in line configuration mode.

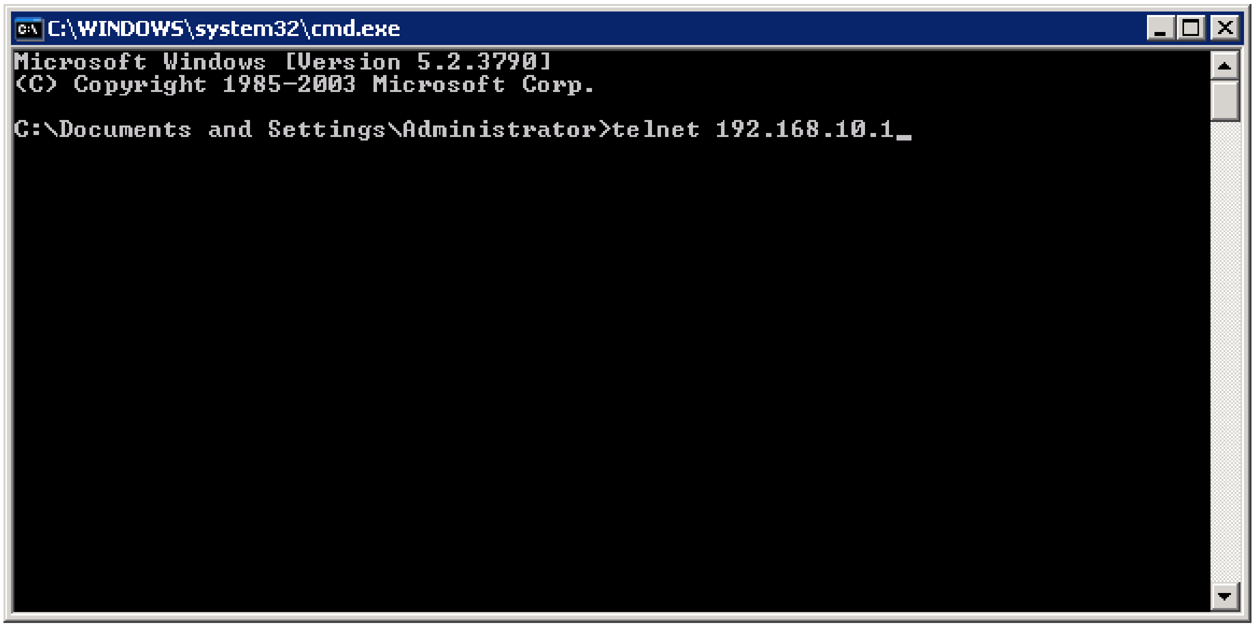
```
R1(config)# aaa authentication login telnet_lines group radius
R1(config)# line vty 0 4
R1(config-line)# login authentication telnet_lines
```

Given the configuration described above, if you enter a username and password pair stored in the ACS authentication database, and the router can reach and use the authentication methods available through RADIUS, would the user be permitted to access the router?

If you enter a username and password pair not stored in the ACS authentication database and the router can reach and use the authentication methods available through RADIUS, would the user be permitted to access the router?

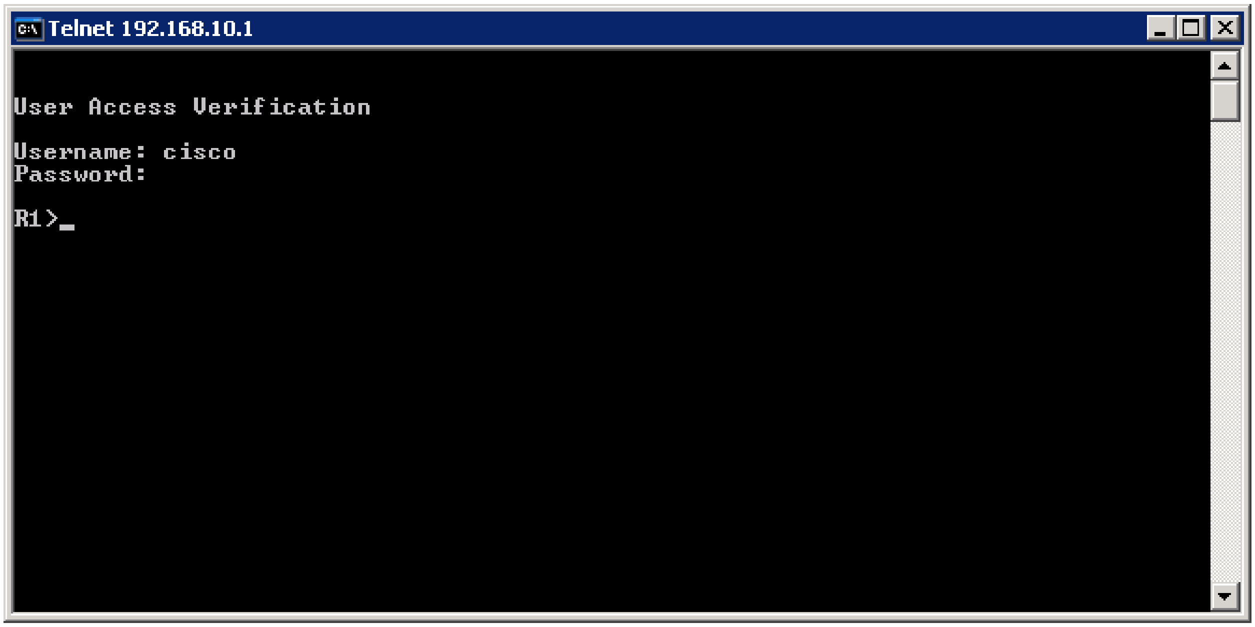
If you entered a username and password pair stored in the ACS authentication database, but the router could not reach a RADIUS server, would the user be permitted to access the router?

You can test your configuration by opening a Telnet session from the host to R1. Click the **Start** button and choose **Run**. Enter the **cmd** command in the Run dialog box, and click **OK**. At the command prompt, issue the **telnet** *host* command. At the login prompt, use the login credentials created earlier: the username and password are both “cisco.”



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>telnet 192.168.10.1_
```

Figure 4-1: Host Telnets to R1



```
User Access Verification

Username: cisco
Password:

R1>_
```

Figure 4-2: Test AAA Authentication Using Telnet

If your session with the router console port times out, you may have to log in using the default authentication list.

Which authentication database does the current default authentication list query?

Why is it advisable to assign redundant authentication methods when using AAA?

Final Configuration

```
R1# show run
hostname R1
!
aaa new-model
!
aaa authentication login default group radius none
aaa authentication login telnet_lines group radius
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
```

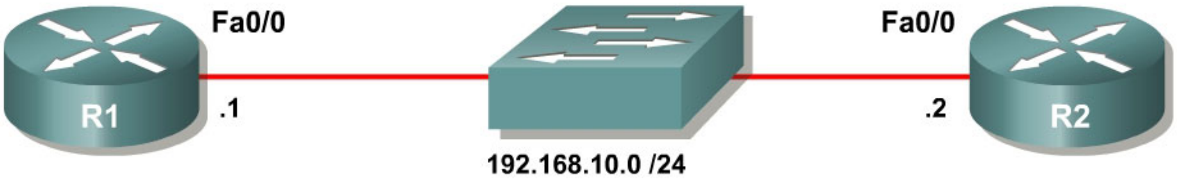
```
!
radius-server host 192.168.10.50 auth-port 1645 acct-port 1646 key ciscosecret
!
line vty 0 4
 login authentication telnet_lines
end
```

Lab 5.6c Configuring AAA Using Local Authentication

Learning Objectives

- Create a local user database on a router
- Configure AAA on a router

Topology Diagram



Scenario

In this lab, you will configure a local username database with AAA. R2's job will only be to access R1; most of the configuration will be done on R1.

Step 1: Configure the Interface

Configure the physical interface on R1 and R2 with the IP addresses shown in the topology diagram. Issue the **no shutdown** command to activate the interface.

```
R1(config)# interface fastethernet0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown

R2(config)# interface fastethernet0/0
R2(config-if)# ip address 192.168.10.2 255.255.255.0
R2(config-if)# no shutdown
```

Step 2: Configure the Local User Database

On R1, configure a user account in the local user database with the username and password of "cisco" using the **username** *username* **password** *password* command in global configuration mode.

```
R1(config)# username cisco password cisco
```

Step 3: Implement AAA Services

On R1, enable authentication, authorization, and accounting (AAA) services with the global configuration command **aaa new-model**. Since you are implementing local authentication, use local authentication as the first method, and no authentication as the secondary method. Create the default login authentication list by issuing the **aaa authentication login default** *methods* command with a method list using the **local** and **none** keywords.

If you were using an authentication method with a remote server, such as TACACS+ or RADIUS, you would configure a secondary authentication method for fallback if the server is unreachable. Normally, the secondary method would be the local database. In this case, if no usernames are configured in the local database, the router allows all users login access to the device.

```
R1(config)# aaa new-model
R1(config)# aaa authentication login default local none
```

Note: If you do not set up a default login authentication list, you could get locked out of the router and be forced to use the password recovery procedure for your specific router.

Next, create a unique authentication list for Telnet access to the router. This does not have the fallback of no authentication, so if there are no usernames in the local database, Telnet access is disabled. To create an authentication list that is not the default list, use the **aaa authentication login** *name method1* [*method2*] [*method3*] command. Name the authentication method list “telnet_lines.” To apply the authentication list to vtys, use the **login authentication** *name* command in line configuration mode.

```
R1(config)# aaa authentication login telnet_lines local
R1(config)# line vty 0 4
R1(config-line)# login authentication telnet_lines
```

Verify that this authentication list is used by opening a Telnet session from R2 to R1.

```
R2# telnet 192.168.10.1
Trying 192.168.10.1 ... Open


User Access Verification

Username: cisco
Password: cisco

R1> exit

[Connection to 192.168.10.1 closed by foreign host]
R2#
```

If you are able to log in to the router, your user account was verified against the local database on the router.

If your session with the console port of the router times out, you may have to log in using the default authentication list.

If no user accounts are configured in the local database, which users are permitted to access the device?

Final Configuration

```
R1# show run
hostname R1
!
aaa new-model
!
aaa authentication login default local none
aaa authentication login telnet_lines local
!
username cisco password 0 cisco
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
!
line vty 0 4
 login authentication telnet_lines
end

R2# show run
hostname R2
!
interface FastEthernet0/0
 ip address 192.168.10.2 255.255.255.0
 no shutdown
end
```

Lab 5.7 Configuring Role-Based CLI Views

Learning Objectives

- Configure prerequisites to role-based views
- Enable AAA on a router
- Change views on a router
- Create views and superviews

Topology Diagram

Scenario

In Lab 5.4, “Enhancing Router Security,” you assigned privilege levels to specific commands entered at the command-line interface (CLI) prompt. Users receive authorization for different command sets by authenticating with a password.

In this lab, you will configure role-based CLI views, a newer method of controlling which Cisco IOS commands a user can execute.

Step 1: Configure an Enable Secret Password

Set the R1 enable secret password to “cisco”.

```
R1(config)# enable secret cisco
```

Step 2: Enable AAA

One of the requirements for configuring role-based CLI views is enabling authentication, authorization, and accounting (AAA) services. To begin, create a user account in the local database with the username and password “cisco”. The local database should be the only login authentication method in use. If you do not set a default login method list when enabling AAA, you may get locked out of the router if your EXEC session on the console line terminates. For more information on configuring AAA with a local database, see Lab 5.8, “Configuring AAA Using Local Authentication.”

```
R1(config)# username cisco password cisco
R1(config)# aaa new-model
R1(config)# aaa authentication login default local
```

When are you prompted to enter a username and password?

If there are no user accounts configured in the local database, are users able to login?

Describe the concept of authentication in terms of networking and standard authentication types.

Describe the concept of authorization in terms of networking and common items in need of authorization.

.

Step 3: Change to the Root View

Role-based CLI views constitute a system of configuring individual roles on a router. Each role has access to a specific group of commands. Configuring roles to control command usage is much more granular than configuring privilege levels, because giving more commands to a single user does not necessarily mean that the user is authorized to access commands at a lower privilege commands. This method of configuring command usage is newer, introduced in the Cisco IOS 12.3T software train. As of the time of this writing, you may configure up to 15 views on a router, not including the root view.

To show the current view, use the **show parser view** command. Compare this output to that of the **show privilege** command.

```
R1# show privilege
Current privilege level is 15

R1# show parser view
No view is active ! Currently in Privilege Level Context
```

Available command sets are determined by either privilege level or by the view being used, but not both simultaneously.

In order to configure the views feature, you must first access the root view, which is not the same as being privilege level 15. Like the root user on a UNIX system, the root view has full authorization to all CLI commands. Issue the **enable view** *name* command using the **root** keyword in the *name* field.

Take special note that the root view password is the same as the enable password. You will notice that a message is logged when the view is changed. After entering the root view, display the privilege level and view.

```
R1# enable view root
Password: cisco

R1#

*Feb 12 05:09:06.442: %PARSER-6-VIEW_SWITCH: successfully set to view 'root'.

R1# show privilege
Currently in View Context with view 'root'
R1# show parser view
Current view is 'root'
```

Why must command authorization be managed by *either* views or privilege levels?

Step 4: Create Views

The role-based view feature is fairly simple to implement. To create a view, issue the **parser view** *name* command in global configuration mode. An informational message that a new view has been created is logged to the console.

Create a view named INTVIEW, which has monitoring capabilities for physical and logical interfaces. Before defining the view’s command set, you must set a password for the view using the view configuration **secret** *password* command. The password is stored as an MD5 hash value. Use “iv” as the password. Choose commands for the view using the **commands** *prompt* **include** *command-sequence* command. Assign this view access to two commands: **show interface** and **clear counters**.

```
R1(config)# parser view INTVIEW
```

```
R1(config-view)#
*Feb 12 05:12:32.954: %PARSER-6-VIEW_CREATED: view 'INTVIEW' successfully
created.
R1(config-view)# secret iv
R1(config-view)# commands exec include show interface
R1(config-view)# commands exec include clear counters
```

Before logging into the new view, display the commands that were just added.

```
R1# show run | section view
parser view INTVIEW
 secret 5 $1$CPI4$HIAH8aEqPztTPW0VLBYT60
 commands exec include show interfaces
 commands exec include show
 commands exec include clear counters
 commands exec include clear
```

When you assign a privilege level to a command sequence, each keyword in the sequence must have a corresponding **privilege** command in the configuration. Similarly, role-based view command sequences must also explicitly allow sequenced keywords in CLI commands, because of the manner in which the parser handles commands.

Log in to the INTVIEW view with the **enable view** *name* command using the “iv” password, and then enter **?** to view the available command set.

```
R1# enable view INTVIEW
Password: iv

R1#

*Feb 12 05:32:31.106: %PARSER-6-VIEW_SWITCH: successfully set to view
'INTVIEW'.

R1# ?
Exec commands:
  clear   Reset functions
  enable  Turn on privileged commands
  exit    Exit from the EXEC
  show    Show running system information

R1# show ?
  flash:      display information about flash: file system
  interfaces  Interface status and configuration
  parser      Display parser information

R1# show interfaces
FastEthernet0/0 is administratively down, line protocol is down
  Hardware is MV96340 Ethernet, address is 0019.0623.4380 (bia 0019.0623.4380)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
<OUTPUT OMITTED>

R1# clear ?
  counters  Clear counters on one or all interfaces

R1# clear counters
Clear "show interface" counters on all interfaces [confirm]
R1#
```

```
*Feb 12 05:32:55.318: %CLEAR-5-COUNTERS: Clear counter on all interfaces by
console
```

Log out of the INTVIEW view and log in to the root view before proceeding.

```
R1# enable view root
Password: cisco

R1#
*Feb 12 05:35:25.174: %PARSER-6-VIEW_SWITCH: successfully set to view 'root'.
```

Create another view named INTSHUT, and assign this view access to the **shutdown** and **no shutdown** commands for the Fast Ethernet interfaces and the menus necessary to configure these commands. Make the password for this view “is”. If your router has different ports, use any two existing ports on the router.

Which commands do you have to add to this view to allow the access defined above?

Enter these commands as follows:

```
R1(config)# parser view INTSHUT
R1(config-view)#

*Feb 12 05:36:37.738: %PARSER-6-VIEW_CREATED: view 'INTSHUT' successfully
created.

R1(config-view)# secret is
R1(config-view)# commands exec include configure terminal
R1(config-view)# commands configure include interface
R1(config-view)# commands configure include interface fastethernet0/0
R1(config-view)# commands configure include interface fastethernet0/1
R1(config-view)# commands interface include shutdown
R1(config-view)# commands interface include no shutdown
```

Enter this new view to test out its privileges. Again, use **?** to view the available command set.

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ?
Configure commands:
  do         To run exec commands in config mode
  exit       Exit from configure mode
  interface  Select an interface to configure

R1(config)# interface fastethernet0/0
R1(config-if)# ?
Interface configuration commands:
  exit      Exit from interface configuration mode
  no        Negate a command or set its defaults
  shutdown  Shutdown the selected interface

R1(config-if)# no shutdown
R1(config-if)#

*Feb 12 06:28:36.394: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Feb 12 06:28:37.394: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up

R1(config-if)#shutdown
```

Return to the root view.

```
R1# enable view root
Password: cisco

R1#
```

Step 5: Create a Superview

A superview is the union of one or more regular views. It is created like a regular view, but you use the superview keyword to define it. Name this superview INTADMIN with the password “ia”. Finally, add the two existing views to this superview using the **view** *name* command.

```
R1(config)# parser view INTADMIN superview
R1(config-view)#

*Feb 12 06:35:06.566: %PARSER-6-SUPER_VIEW_CREATED: super view 'INTADMIN'
successfully created.

R1(config-view)# secret ia
R1(config-view)# view INTVIEW

*Feb 12 06:35:21.086: %PARSER-6-SUPER_VIEW_EDIT_ADD: view INTVIEW added to
superview INTADMIN.

R1(config-view)# view INTSHUT

*Feb 12 06:35:29.594: %PARSER-6-SUPER_VIEW_EDIT_ADD: view INTSHUT added to
superview INTADMIN.
```

While still in the root view, display the available parser views and superviews with the **show parser view all** command.

```
R1# show parser view all
Views/SuperViews Present in System:
 INTVIEW
 INTSHUT
 INTADMIN *
-------(*) represent superview-------
R1#
```

Enter this view and see the available executable commands.

```
R1# enable view INTADMIN
Password:
```

```
R1#
*Feb 12 06:36:31.774: %PARSER-6-VIEW_SWITCH: successfully set to view
'INTADMIN'.
R1# ?
Exec commands:
  clear      Reset functions
  configure  Enter configuration mode
  enable     Turn on privileged commands
  exit       Exit from the EXEC
  show       Show running system information’
```

Final Configuration

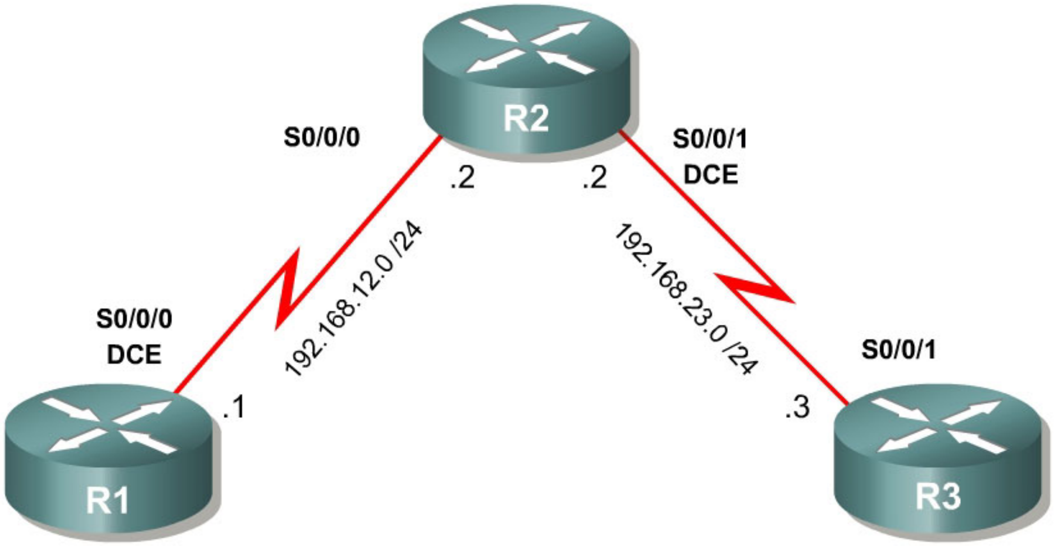
```
R1# show run
hostname R1
!
enable secret 5 $1$lETz$132w/UItPj25T6EFwChis1
!
aaa new-model
!
aaa authentication login default local
!
username cisco password 0 cisco
!
parser view INTVIEW
 secret 5 $1$CPI4$HIAH8aEqPztTPW0VLBYT60
 commands exec include show interfaces
 commands exec include show
 commands exec include clear counters
 commands exec include clear
!
parser view INTSHUT
 secret 5 $1$yeoh$asrBOTkwESSy.0lpCZgG.1
 commands interface include shutdown
 commands interface include no shutdown
 commands interface include no
 commands configure include interface
 commands exec include configure terminal
 commands exec include configure
 commands configure include interface FastEthernet0/0
 commands configure include interface FastEthernet0/1
!
parser view INTADMIN superview
 secret 5 $1$QFfM$GUXOqAtkwSTKt1aKuYKdN/
 view INTVIEW
 view INTSHUT
end
```

Lab 5.8 Configuring NTP

Learning Objectives

- Configure a router as an NTP master server
- Configure an NTP server on a router
- Configure an NTP peer
- Implement NTP authentication

Topology Diagram



Scenario

In this lab, you will configure network time protocol (NTP) in a small topology. NTP is essential in a large network, because it reduces administrative overhead as well as allows for consistent times throughout the network for logging and other time-related features, such as crypto certificate lifetimes.

Step 1: Configure the Physical Interfaces

Configure the loopback interfaces with the addresses shown in the topology diagram. Also configure the serial interfaces shown in the diagram. Set the clock rate on the appropriate interface, and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command.

```
R1(config)# interface serial0/0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# no shutdown
```

```
R2(config)# interface serial0/0/0
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/0/1
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3(config)# interface serial0/0/1
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Set Up the NTP Master

R1 is the master NTP server in this lab. All other routers learn their time from it, either directly or indirectly. For this reason, you must first ensure that R1 has the correct Coordinated Universal Time set.

Display the current time set on the router using the **show clock** command. To set the time on the router, use the **clock set** *time* command.

```
R1# show clock
*07:20:19.267 UTC Mon Feb 12 2007

R1# clock set 07:20:30 feb 12 2007
R1#
*Feb 12 07:20:30.000: %SYS-6-CLOCKUPDATE: System clock has been updated from
07:20:39 UTC Mon Feb 12 2007 to 07:20:30 UTC Mon Feb 12 2007, configured from
console by console.
```

Configure R1 as the NTP master using the **ntp master** *stratum* command in global configuration mode. The stratum number indicates the distance from the original source. For this lab, use a stratum number of 5 on R1.

When a device learns the time from an NTP source, its stratum number becomes one greater than its source's stratum number.

```
R1(config)# ntp master 5
```

Step 3: Configure an NTP Client

R2 will become an NTP client of R1. To configure R2, use the global configuration command **ntp server** *hostname*. Hostname can also be an IP address.

```
R2(config)# ntp server 192.168.12.1
```

After a while, verify that R2 has made an association with R1 with the **show ntp association** command. You can also use the more verbose version of the command by adding the **detail** argument. It may take some time for the NTP association to form.

```
R2# show ntp associations

    address         ref clock     st  when  poll reach  delay  offset    disp
```

```
*~192.168.12.1   127.127.7.1       5    24    64  377    23.1    0.72     0.5
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

R2# show ntp associations detail
192.168.12.1 configured, our_master, sane, valid, stratum 5
ref ID 127.127.7.1, time C97A9634.A5E51ED1 (07:31:00.648 UTC Mon Feb 12 2007)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.03, reach 377, sync dist 12.039
delay 23.09 msec, offset 0.7242 msec, dispersion 0.47
precision 2**18, version 3
org time C97A9643.CF0A3D1F (07:31:15.808 UTC Mon Feb 12 2007)
rcv time C97A9643.D1CFC661 (07:31:15.819 UTC Mon Feb 12 2007)
xmt time C97A9643.CBE4198B (07:31:15.796 UTC Mon Feb 12 2007)
filtdelay =    23.09   23.28   23.13   23.24   23.16   23.22   23.35   23.28
filtoffset =    0.72    0.44    0.07    0.06    0.04    0.06    0.05   -0.01
filterror =     0.02    0.99    1.97    1.98    2.00    2.01    2.03    2.04
```

Step 4: Configure NTP Peers with MD5 Authentication

In addition to the client-server model, NTP can also function with routers in a peer relationship in which each router synchronizes against its peers. For this scenario, R2 and R3 maintain a peering relationship.

Which security risks can either of these relationships pose?

To avoid a spoofing problem, configure MD5 authentication between the two NTP peers, R2 and R3. Usually, when NTP authentication is configured in a client-server model, the client authenticates the server, but not vice versa. Thus, NTP authentication is *source* authentication; clients do not need to be authenticated because they cannot manipulate the clock on the server.

However, because there is a peering relationship in which each peer may act as a corrector to the other device, each device must be configured as an authenticated NTP source. First, enable NTP authentication with the **ntp authenticate** command in global configuration mode. Next, add an NTP authentication key to the router with the **ntp authentication-key** *number* **md5** *key-string* command. Apply a key number of 1 for the key “cisco”. Finally, apply the authentication configuration by specifying NTP key number 1 as a trusted NTP source key with the **ntp trusted-key** *number* command.

```
R2(config)# ntp authenticate
R2(config)# ntp authentication-key 1 md5 cisco
R2(config)# ntp trusted-key 1

R3(config)# ntp authenticate
R3(config)# ntp authentication-key 1 md5 cisco
R3(config)# ntp trusted-key 1
```

Configure the NTP peer on R3. NTP peers have a passive side and an active side. You only have to configure the active side, in this case R3. R2 is listening on the NTP port and will form a peer relationship through this. Do not configure peers on both sides of the peer relationship, or it will not work. One of the devices in the peer relationship must be in active mode and the other device must be in passive mode for proper peer synchronization to occur.

```
R3(config)# ntp peer 192.168.23.2
```

It may take a few moments for the relationship to establish. On each of the three routers, verify NTP status and associations using the **show ntp status**, **show ntp associations**, and **show ntp associations detail** commands. Notice how the stratum level increases at each hop. Verify that their clocks are indeed synchronized with the **show clock** command.

```
R1# show ntp status
Clock is synchronized, stratum 5, reference is 127.127.7.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
reference time is C97A9B74.A5DF14AD (07:53:24.647 UTC Mon Feb 12 2007)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 0.02 msec, peer dispersion is 0.02 msec

R1# show ntp associations
     address         ref clock     st  when  poll reach  delay  offset    disp
*~127.127.7.1     127.127.7.1       4    55    64  377     0.0    0.00     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

R1# show ntp associations detail
127.127.7.1 configured, our_master, sane, valid, stratum 4
ref ID 127.127.7.1, time C97A9B74.A5DF14AD (07:53:24.647 UTC Mon Feb 12 2007)
our mode active, peer mode passive, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.00, reach 377, sync dist 0.015
delay 0.00 msec, offset 0.0000 msec, dispersion 0.02
precision 2**18, version 3
org time C97A9B74.A5DF14AD (07:53:24.647 UTC Mon Feb 12 2007)
rcv time C97A9B74.A5DF14AD (07:53:24.647 UTC Mon Feb 12 2007)
xmt time C97A9B74.A5DE90AF (07:53:24.647 UTC Mon Feb 12 2007)
filtdelay =     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtoffset =    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filterror =     0.02    0.99    1.97    2.94    3.92    4.90    5.87    6.85
Reference clock status:  Running normally
Timecode:


R2# show ntp status
Clock is synchronized, stratum 6, reference is 192.168.12.1
nominal freq is 250.0000 Hz, actual freq is 249.9998 Hz, precision is 2**18
reference time is C97A9BC3.D3820015 (07:54:43.826 UTC Mon Feb 12 2007)
clock offset is 1.9937 msec, root delay is 23.32 msec
root dispersion is 2.04 msec, peer dispersion is 0.03 msec

R2# show ntp associations

      address        ref clock     st  when  poll reach  delay  offset    disp
*~192.168.12.1     127.127.7.1      5    14    64  377    23.3    1.99     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

R2# show ntp associations detail
192.168.12.1 configured, our_master, sane, valid, stratum 5
```

```
ref ID 127.127.7.1, time C97A9BB4.A5DEE42C (07:54:28.647 UTC Mon Feb 12 2007)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.03, reach 377, sync dist 11.902
delay 23.32 msec, offset 1.9937 msec, dispersion 0.03
precision 2**18, version 3
org time C97A9BC3.D1082A4F (07:54:43.816 UTC Mon Feb 12 2007)
rcv time C97A9BC3.D3820015 (07:54:43.826 UTC Mon Feb 12 2007)
xmt time C97A9BC3.CD87599E (07:54:43.802 UTC Mon Feb 12 2007)
filtdelay =    23.32   23.38   23.21   23.25   23.07   23.18   23.25   23.22
filtoffset =    1.99    1.95    1.99    1.98    1.93    1.98    1.96    1.94
filterror =     0.02    0.99    1.97    2.94    3.92    4.90    5.87    6.85

R3# show ntp status
Clock is synchronized, stratum 7, reference is 192.168.23.2
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
reference time is C97A9BCF.B82D5269 (07:54:55.719 UTC Mon Feb 12 2007)
clock offset is -1.3696 msec, root delay is 25.59 msec
root dispersion is 3.92 msec, peer dispersion is 0.49 msec

R3# show ntp associations

      address        ref clock     st  when  poll reach  delay  offset    disp
*~192.168.23.2    192.168.12.1     6    27    64  377     2.3   -1.37     0.5
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

R3# show ntp associations detail
192.168.23.2 configured, our_master, sane, valid, stratum 6
ref ID 192.168.12.1, time C97A9BC3.D3820015 (07:54:43.826 UTC Mon Feb 12 2007)
our mode active, peer mode passive, our poll intvl 64, peer poll intvl 64
root delay 23.32 msec, root disp 2.06, reach 377, sync dist 15.335
delay 2.27 msec, offset -1.3696 msec, dispersion 0.49
precision 2**18, version 3
org time C97A9BCF.B788986A (07:54:55.716 UTC Mon Feb 12 2007)
rcv time C97A9BCF.B82D5269 (07:54:55.719 UTC Mon Feb 12 2007)
xmt time C97A9BCF.B7903BF8 (07:54:55.717 UTC Mon Feb 12 2007)
filtdelay =     2.27    2.26    2.29    2.30    2.29    2.27    2.29    2.26
filtoffset =   -1.37   -1.16   -0.90   -0.49   -0.10   -0.10   -0.09   -0.08
filterror =     0.02    0.99    1.97    2.94    3.92    3.94    3.95    3.97
```

Why would it be good to have routers peering equally rather than a client-server relationship?

Final Configuration

```
R1# show run
hostname R1
!
interface Serial0/0/0
 ip address 192.168.12.1 255.255.255.0
 clock rate 64000
 no shutdown
!
ntp master 5
```

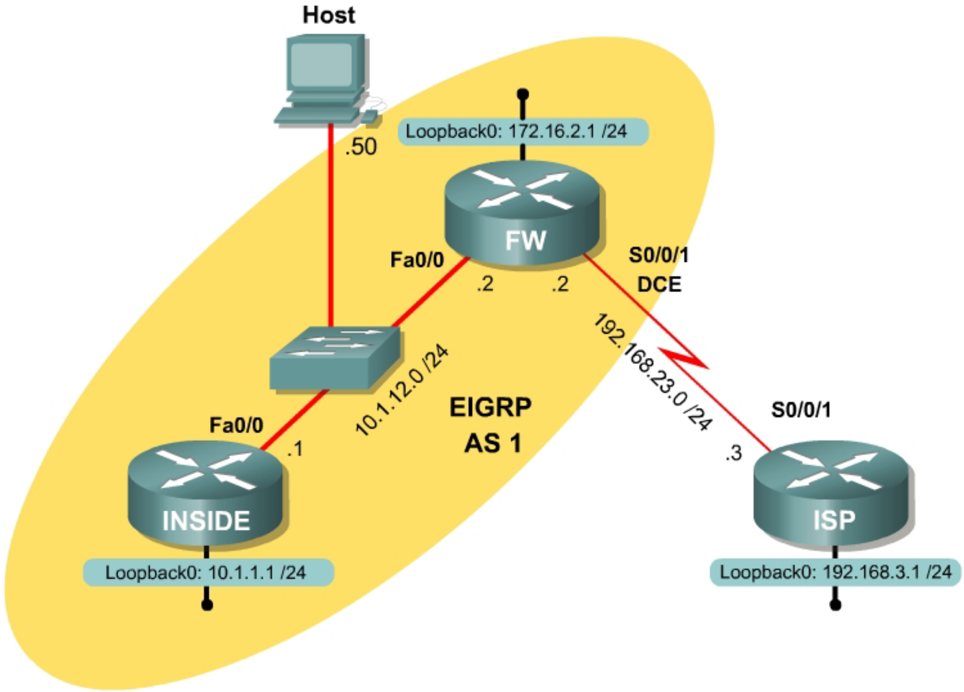
```
end

R2# show run
hostname R2
!
interface Serial0/0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 clockrate 64000
 no shutdown
!
ntp authentication-key 1 md5 01100F175804 7
ntp authenticate
ntp trusted-key 1
ntp server 192.168.12.1
end

R3# show run
hostname R3
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 no shutdown
!
ntp authentication-key 1 md5 00071A150754 7
ntp authenticate
ntp trusted-key 1
ntp peer 192.168.23.2
end
```

Lab 6.1 Configuring a Cisco IOS Firewall Using SDM

Learning Objectives

- Use SDM to configure a router as a firewall
- Understand basic firewall operation
- Configure basic routing through a firewall
- Verify firewall parameters using SDM

Topology Diagram



Scenario

In this lab, you will configure a perimeter router for International Travel Agency (ITA) using Security Device Manager (SDM). The SDM firewall wizard allows you to create a relatively strong firewall configuration in a few simple configuration steps.

Each router represents a router in a large corporate network scenario. The ISP router is an ISP edge router connected to the ITA corporate firewall, and the ISP's loopback is a remote internet network. FW is a corporate firewall router, and its loopback interface is a demilitarized zone (DMZ) where Internet-facing servers reside. INSIDE is an internal corporate router, and its loopback is a subnet within the internal corporate domain.

Step 1: Configure Loopbacks and Physical Interfaces

Configure the loopback interfaces with the addresses shown in the topology diagram. Also configure the serial interfaces. Set the clock rate on the appropriate interface, and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command.

```
INSIDE(config)# interface loopback0
INSIDE(config-if)# ip address 10.1.1.1 255.255.255.0
INSIDE(config-if)# interface fastethernet0/0
INSIDE(config-if)# ip address 10.1.12.1 255.255.255.0
INSIDE(config-if)# no shutdown

FW(config)# interface loopback0
FW(config-if)# ip address 172.16.2.1 255.255.255.0
FW(config-if)# interface fastethernet0/0
FW(config-if)# ip address 10.l.12.2 255.255.255.0
FW(config-if)# no shutdown
FW(config-if)# interface serial0/0/1
FW(config-if)# ip address 192.168.23.2 255.255.255.0
FW(config-if)# clockrate 64000
FW(config-if)# no shutdown

ISP(config)# interface loopback0
ISP(config-if)# ip address 192.168.3.1 255.255.255.0
ISP(config-if)# interface serial0/0/1
ISP(config-if)# ip address 192.168.23.3 255.255.255.0
ISP(config-if)# no shutdown
```

Step 2: Configure Routing Protocols

Because this scenario is limited to three routers, be sure that you clearly understand which each major network in the diagram represents.

The 10.0.0.0/8 address space represents the internal network owned and administered by your corporation. Per your security policy, no traffic should enter the internal network from the ISP unless the connection was initiated from inside the firewall.

The connection between FW and ISP is a globally routable subnet in the public address space. Although RFC 1918 defines the 192.168.0.0/16 network as private, consider it for the purposes of this lab as a global address space.

The loopback interface on the FW router is an interface in the DMZ. A DMZ is often used to host web servers and other servers for which users from outside your network can access certain content without authentication or authorization.

Configure EIGRP for AS1 on INSIDE and FW. Advertise the entire 10.0.0.0/8 major network and disable automatic summarization. Also add the 172.16.0.0 network on FW to advertise it, but categorize the loopback interface as a passive interface, because this is simply a DMZ for servers but not routers. ISP does not participate in this routing process.

```
INSIDE(config)# router eigrp 1
INSIDE(config-router)# no auto-summary
INSIDE(config-router)# network 10.0.0.0

FW(config)# router eigrp 1
FW(config-router)# no auto-summary
FW(config-router)# network 10.0.0.0
FW(config-router)# network 172.16.0.0
FW(config-router)# passive-interface loopback0
```

Define the purpose and operation of a firewall.

Define some basic firewall rules about how traffic should be permitted through the network.

Step 3: Configure Static Routes to Reach the Internet

Since ISP connects to the Internet, send all traffic to networks not in the corporate routing tables to the ISP via a default route. This route can be statically created on FW, but must be redistributed into EIGRP so that routers on the inside of the firewall can learn the default route.

```
FW(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.3
FW(config)# router eigrp 1
FW(config-router)# redistribute static
```

On ISP, create static routes for the major corporate destination networks. Remember that the 172.16.0.0 network is the company’s DMZ, and 10.0.0.0 is its internal network.

```
ISP(config)# ip route 10.0.0.0 255.0.0.0 192.168.23.2
ISP(config)# ip route 172.16.0.0 255.255.0.0 192.168.23.2
```

At this point, you have full IP connectivity between all networks in the diagram. Once the FW router is fully configured, there will only be partial IP connectivity as a result of implementing security policies.

```
ISP# 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:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/112/112 ms
```

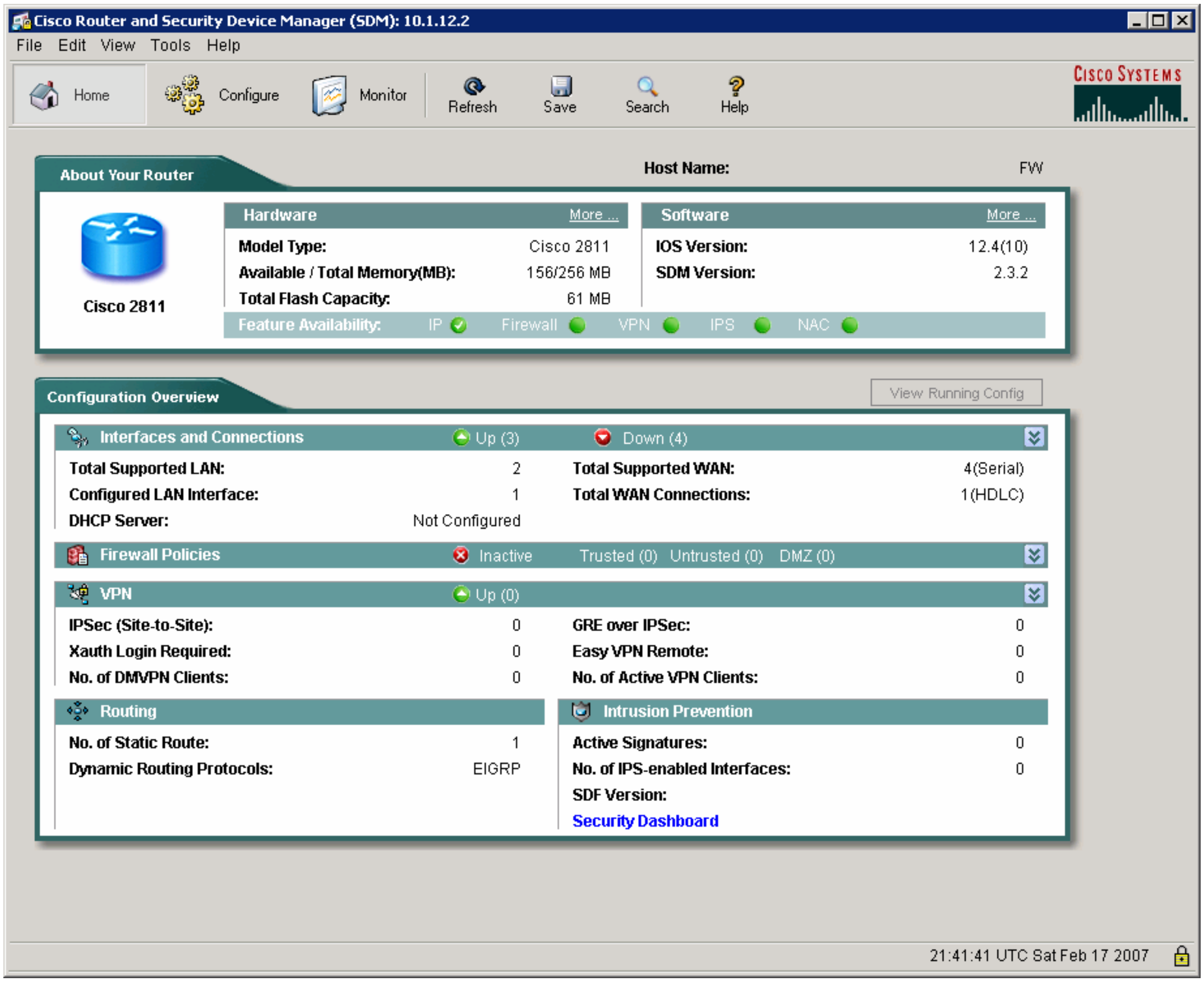
Step 4: Connect to FW Using SDM

Configure the 10.1.12.50/24 IP address on the host, using the FW Fast Ethernet interface as its default gateway. If you do not know how to set the IP address on a host or connect to a router using SDM, consult Lab 2-1: Configuring SDM on a Router.

Prepare FW for access via SDM. Remember that SDM utilizes secure shell (SSH) and secure HTTP (HTTPS) technology and must authenticate with the router before configuration can begin.

```
FW(config)# username ciscosdm privilege 15 password ciscosdm
FW(config)# ip http secure-server
FW(config)# ip http authentication local
FW(config)# line vty 0 4
FW(config-line)# transport input telnet ssh
```

Connect to FW using SDM.



About Your Router

Host Name: FW

Cisco 2811

Hardware More ...

Model Type: Cisco 2811
Available / Total Memory(MB): 156/256 MB
Total Flash Capacity: 61 MB

Software More ...

IOS Version: 12.4(10)
SDM Version: 2.3.2

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Configuration Overview

Interfaces and Connections Up (3) Down (4)

Total Supported LAN: 2
Configured LAN Interface: 1
DHCP Server: Not Configured
Total Supported WAN: 4(Serial)
Total WAN Connections: 1(HDLC)

Firewall Policies ☒ Inactive Trusted (0) Untrusted (0) DMZ (0)

VPN Up (0)

IPSec (Site-to-Site): 0
Xauth Login Required: 0
No. of DMVPN Clients: 0
GRE over IPSec: 0
Easy VPN Remote: 0
No. of Active VPN Clients: 0

Routing

No. of Static Route: 1
Dynamic Routing Protocols: EIGRP

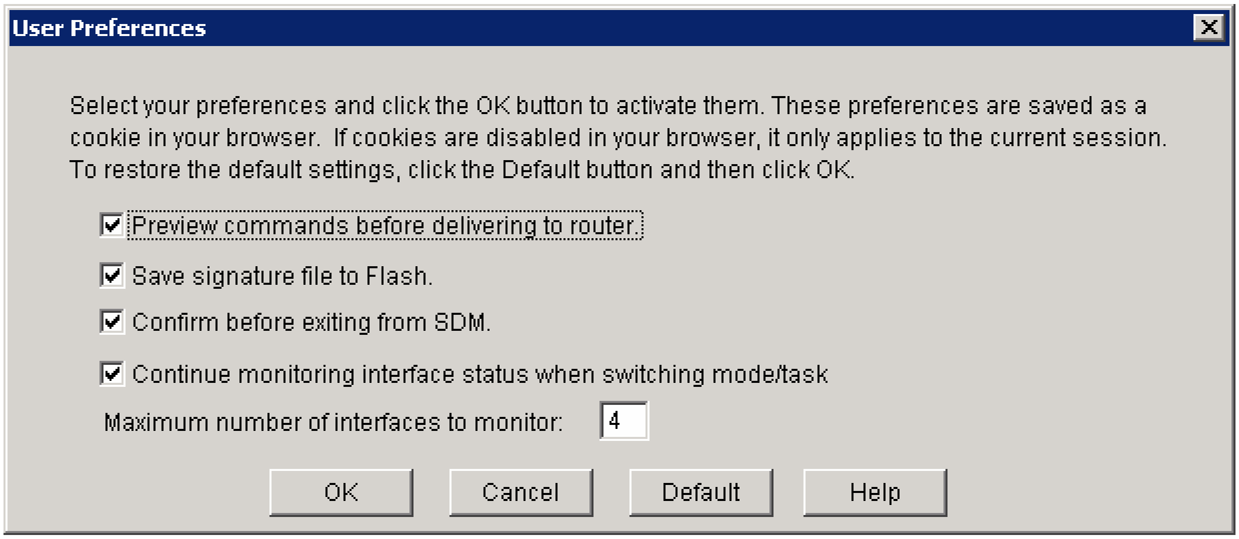
Intrusion Prevention

Active Signatures: 0
No. of IPS-enabled Interfaces: 0
SDF Version:
Security Dashboard

21:41:41 UTC Sat Feb 17 2007

Figure 4-1: Cisco Security Device Manager Home Screen

Select **Edit** > **Preferences**. Make sure that **Preview commands before delivering to router** is checked before continuing. Click **OK** when you are done configuring the preferences.


User Preferences

Select your preferences and click the OK button to activate them. These preferences are saved as a cookie in your browser. If cookies are disabled in your browser, it only applies to the current session. To restore the default settings, click the Default button and then click OK.

☑ Preview commands before delivering to router.

☑ Save signature file to Flash.

☑ Confirm before exiting from SDM.

☑ Continue monitoring interface status when switching mode/task

Maximum number of interfaces to monitor: 4

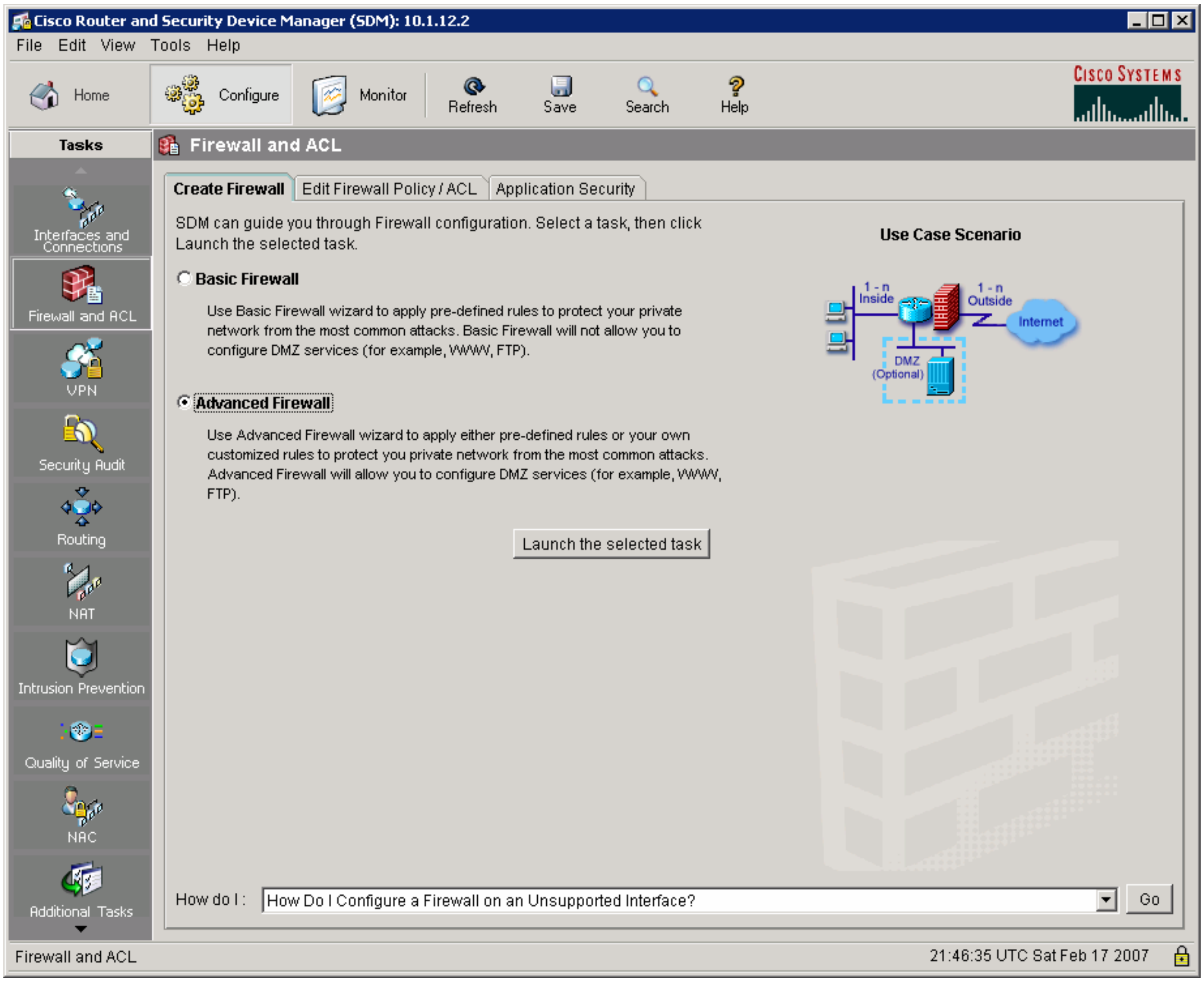
Figure 4-2: SDM User Preferences

Step 5: Use the SDM Advanced Firewall Wizard

In SDM, start the Advanced Firewall Wizard by clicking **Configure** on the top toolbar and then clicking **Firewall and ACL** on the left-hand toolbar.

On the **Create Firewall** tab, choose **Advanced Firewall**, and then click **Launch the selected task**.

You will not configure the basic firewall in this lab because it is a less robust version of the advanced firewall. If you can successfully configure the advanced firewall, you will have no problem configuring a basic firewall. The difference is that you have less control with a basic firewall and there is no DMZ support.



SDM can guide you through Firewall configuration. Select a task, then click Launch the selected task.

○ **Basic Firewall**

Use Basic Firewall wizard to apply pre-defined rules to protect your private network from the most common attacks. Basic Firewall will not allow you to configure DMZ services (for example, WWW, FTP).

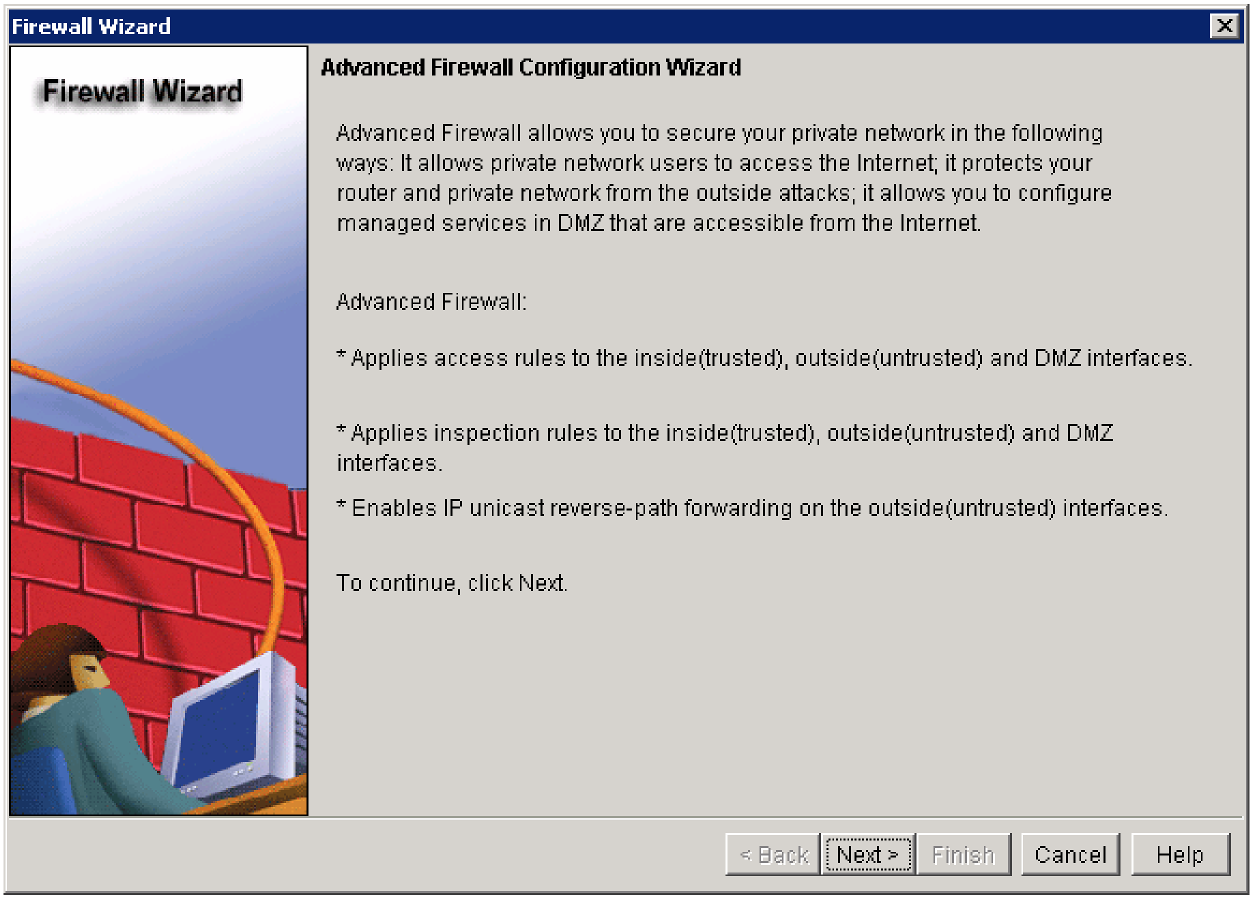
● **Advanced Firewall**

Use Advanced Firewall wizard to apply either pre-defined rules or your own customized rules to protect you private network from the most common attacks. Advanced Firewall will allow you to configure DMZ services (for example, WWW, FTP).

How do I : How Do I Configure a Firewall on an Unsupported Interface?

Figure 5-1: SDM Create Firewall Tab

Read the information for the Advanced Firewall Configuration Wizard, and click **Next**.



Advanced Firewall Configuration Wizard

Advanced Firewall allows you to secure your private network in the following ways: It allows private network users to access the Internet; it protects your router and private network from the outside attacks; it allows you to configure managed services in DMZ that are accessible from the Internet.

Advanced Firewall:

* Applies access rules to the inside(trusted), outside(untrusted) and DMZ interfaces.

* Applies inspection rules to the inside(trusted), outside(untrusted) and DMZ interfaces.

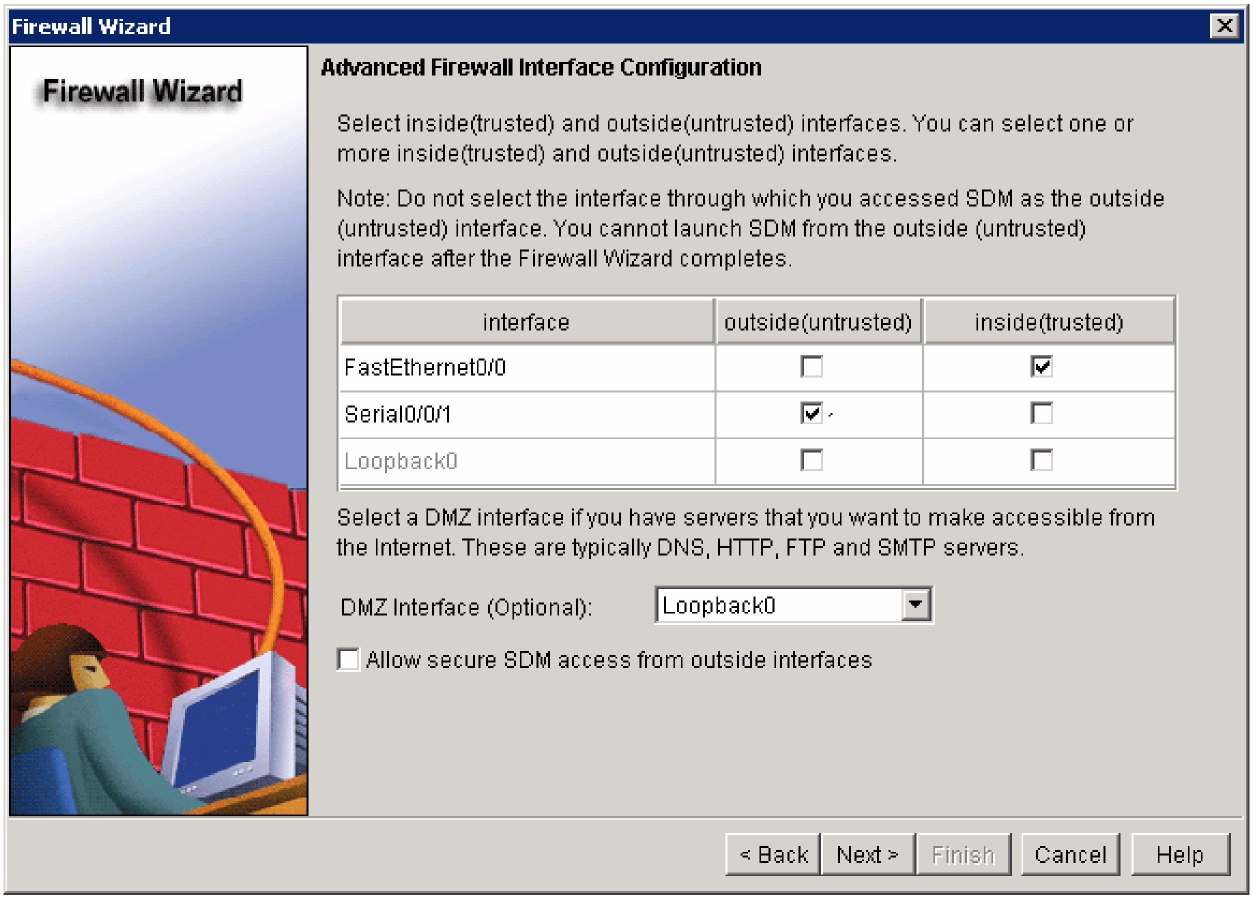
* Enables IP unicast reverse-path forwarding on the outside(untrusted) interfaces.

To continue, click Next.

Figure 5-2: SDM Advanced Firewall Configuration Wizard

Choose the ISP-facing interface to be the outside untrusted interface, and the interface that faces the INSIDE router as the inside trusted interface.

Do not check either the inside or outside boxes for the loopback interface. However, select the loopback interface as the DMZ interface. You do not need to allow secure SDM access from the outside interfaces. Once you have configured all interfaces correctly, click **Next**.



Advanced Firewall Interface Configuration

Select inside(trusted) and outside(untrusted) interfaces. You can select one or more inside(trusted) and outside(untrusted) interfaces.

Note: Do not select the interface through which you accessed SDM as the outside (untrusted) interface. You cannot launch SDM from the outside (untrusted) interface after the Firewall Wizard completes.

interface	outside(untrusted)	inside(trusted)
FastEthernet0/0	□	☑
Serial0/0/1	☑	□
Loopback0	□	□

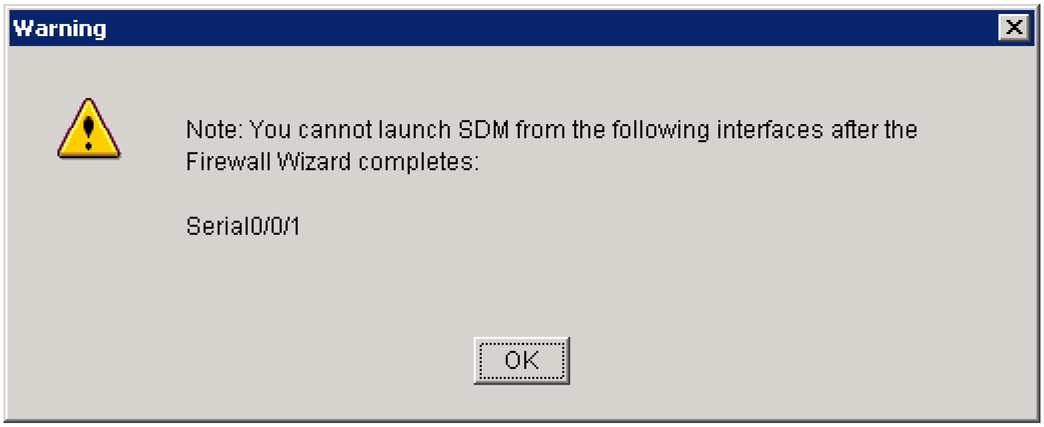
Select a DMZ interface if you have servers that you want to make accessible from the Internet. These are typically DNS, HTTP, FTP and SMTP servers.

DMZ Interface (Optional): Loopback0

□ Allow secure SDM access from outside interfaces

Figure 5-3: SDM Firewall Wizard Interface Selection

Since you did not check the option at the bottom, SDM warns you that you cannot configure the router using SDM from the outside interface. Because you are using SDM through the inside interface of FW, this is acceptable. Click **OK** to continue.

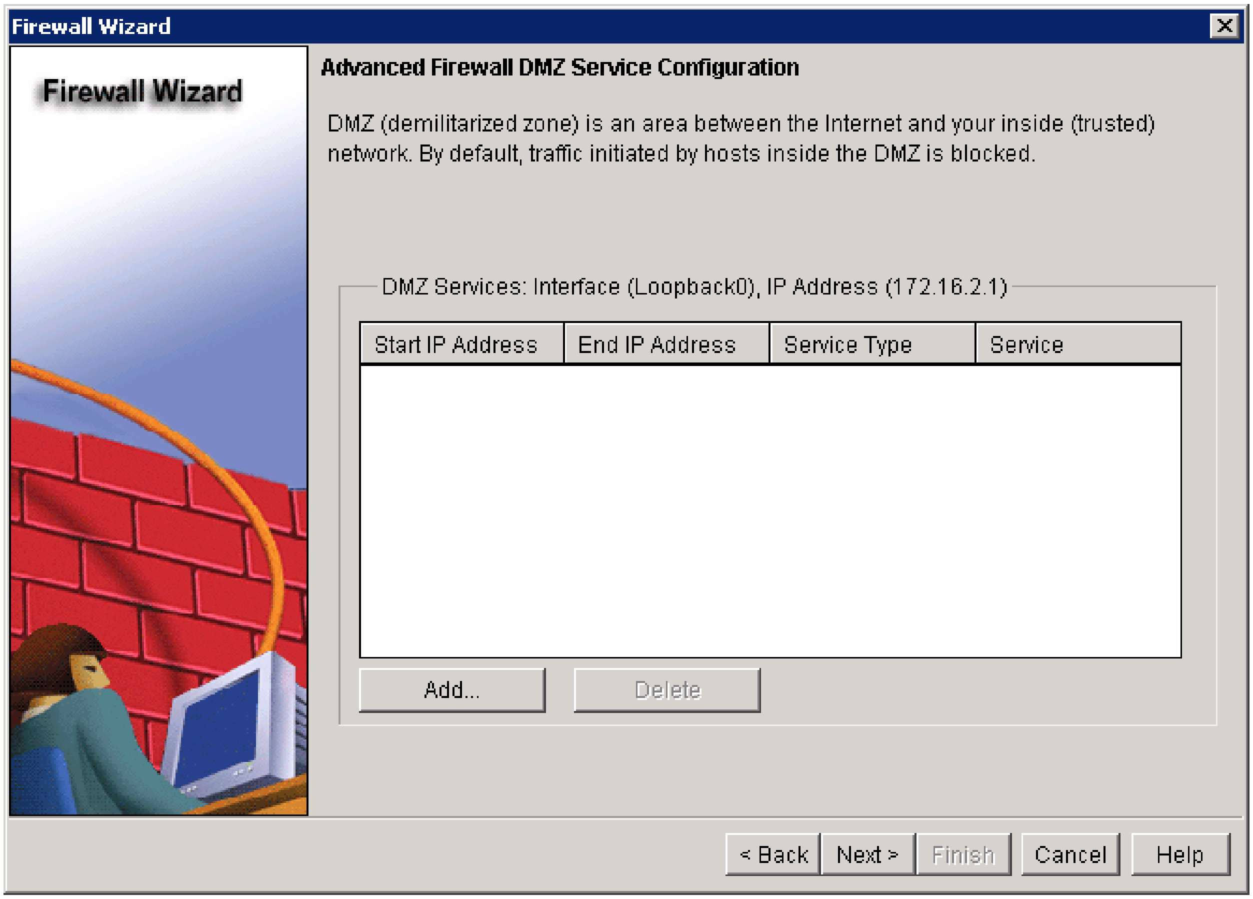


Note: You cannot launch SDM from the following interfaces after the Firewall Wizard completes:

Serial0/0/1

Figure 5-4: SDM Connectivity Warning

Now add DMZ services, by clicking the **Add** button.



Advanced Firewall DMZ Service Configuration

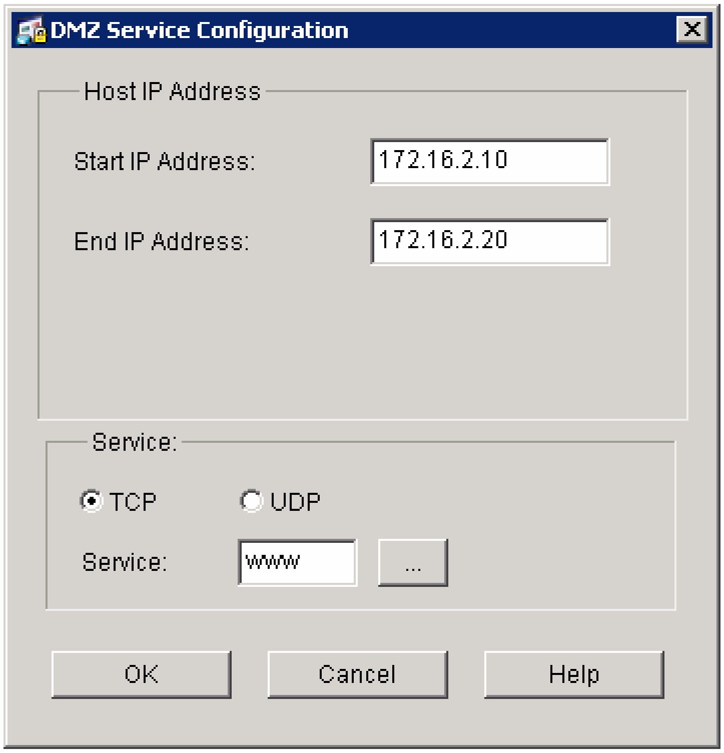
DMZ (demilitarized zone) is an area between the Internet and your inside (trusted) network. By default, traffic initiated by hosts inside the DMZ is blocked.

DMZ Services: Interface (Loopback0), IP Address (172.16.2.1)

Start IP Address	End IP Address	Service Type	Service

Figure 5-5: SDM Firewall Demilitarized Zone Configuration

Add a range of web servers with addresses from 172.16.2.10 to 172.16.2.20. Configure the service as “www”, a keyword that the router identifies with TCP port 80, the standard HTTP port. Click **OK** when your configuration is complete.



Host IP Address

Start IP Address: 172.16.2.10

End IP Address: 172.16.2.20

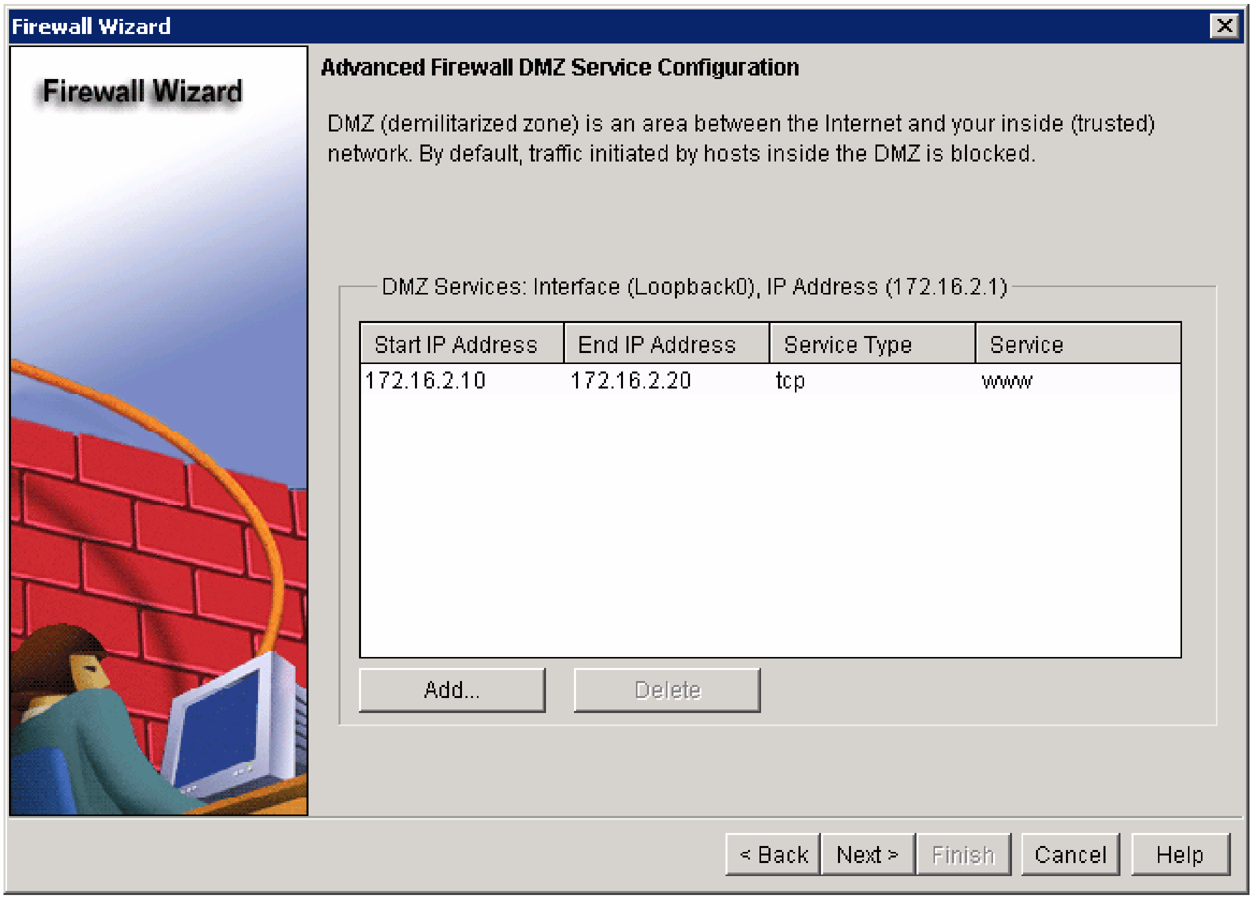
Service:

● TCP ○ UDP

Service: www

Figure 5-6: SDM DMZ Service Configuration

Review the added servers in the list of DMZ services, and then click **Next**.



Advanced Firewall DMZ Service Configuration

DMZ (demilitarized zone) is an area between the Internet and your inside (trusted) network. By default, traffic initiated by hosts inside the DMZ is blocked.

DMZ Services: Interface (Loopback0), IP Address (172.16.2.1)

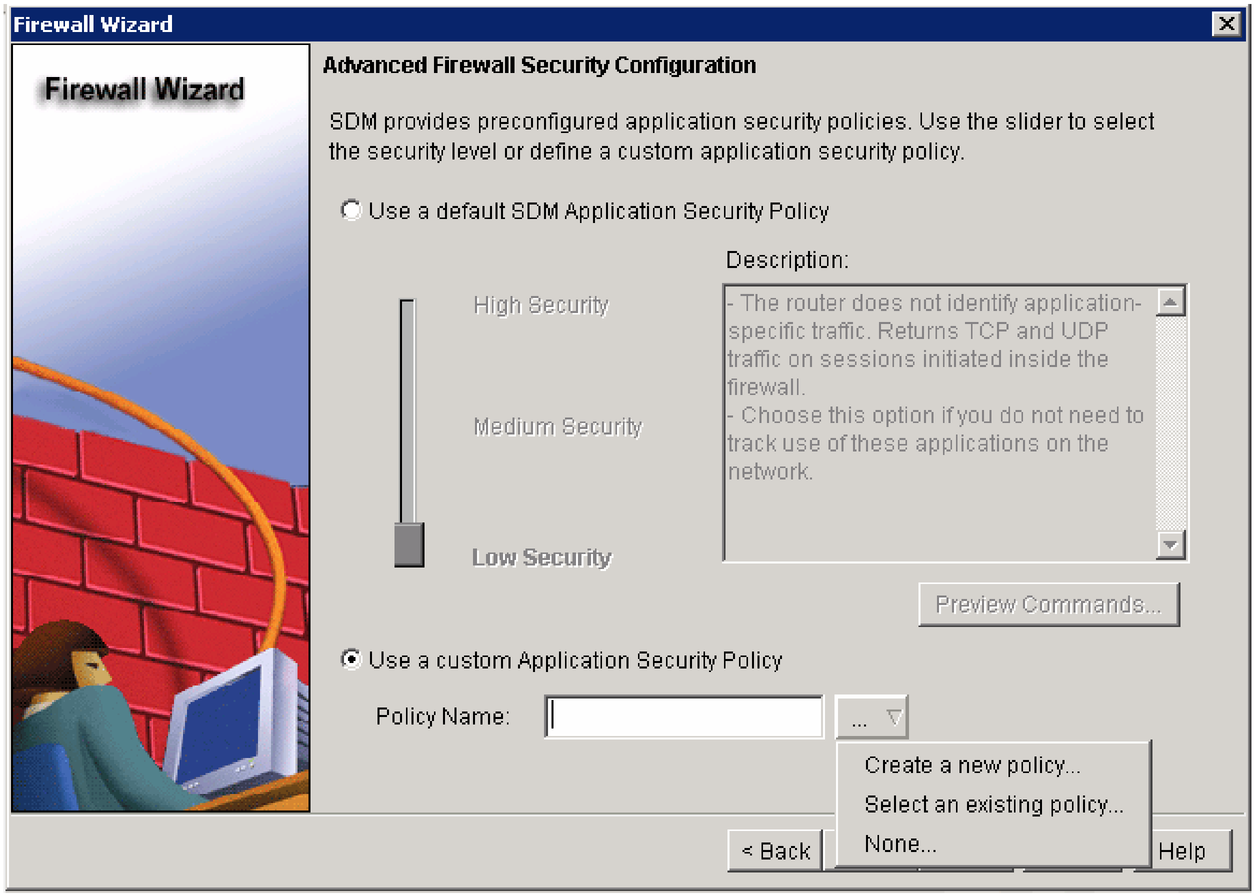
Start IP Address	End IP Address	Service Type	Service
172.16.2.10	172.16.2.20	tcp	www

Figure 5-7: SDM Firewall Demilitarized Zone Configuration with Changes Applied

The configuration of interfaces as inside, outside, and DMZ interfaces applies access control lists to interfaces, creating firewall traffic boundaries.

Therefore you may want to create permissions through the firewall for specific applications. Context-based access control (CBAC) lists can examine the TCP and UDP layers of packets passing through the firewall to create per-flow holes in the firewall. CBAC watches generic transport protocols, such as TCP and UDP, and tracks connections, permitting incoming untrusted traffic only if it is return traffic for a TCP or UDP connection initiated from the inside. For some well-known protocols, CBAC also examines certain application-layer data to make sure that the packets are following those application protocols correctly. For more information on CBAC, see Lab 6.2: Configuring CBAC.

For this lab, you will not use a default application security policy, but instead create your own. Click **Use a custom Application Security Policy** and then click **Create a new policy** in the drop-down box.



Advanced Firewall Security Configuration

SDM provides preconfigured application security policies. Use the slider to select the security level or define a custom application security policy.

○ Use a default SDM Application Security Policy

Description:

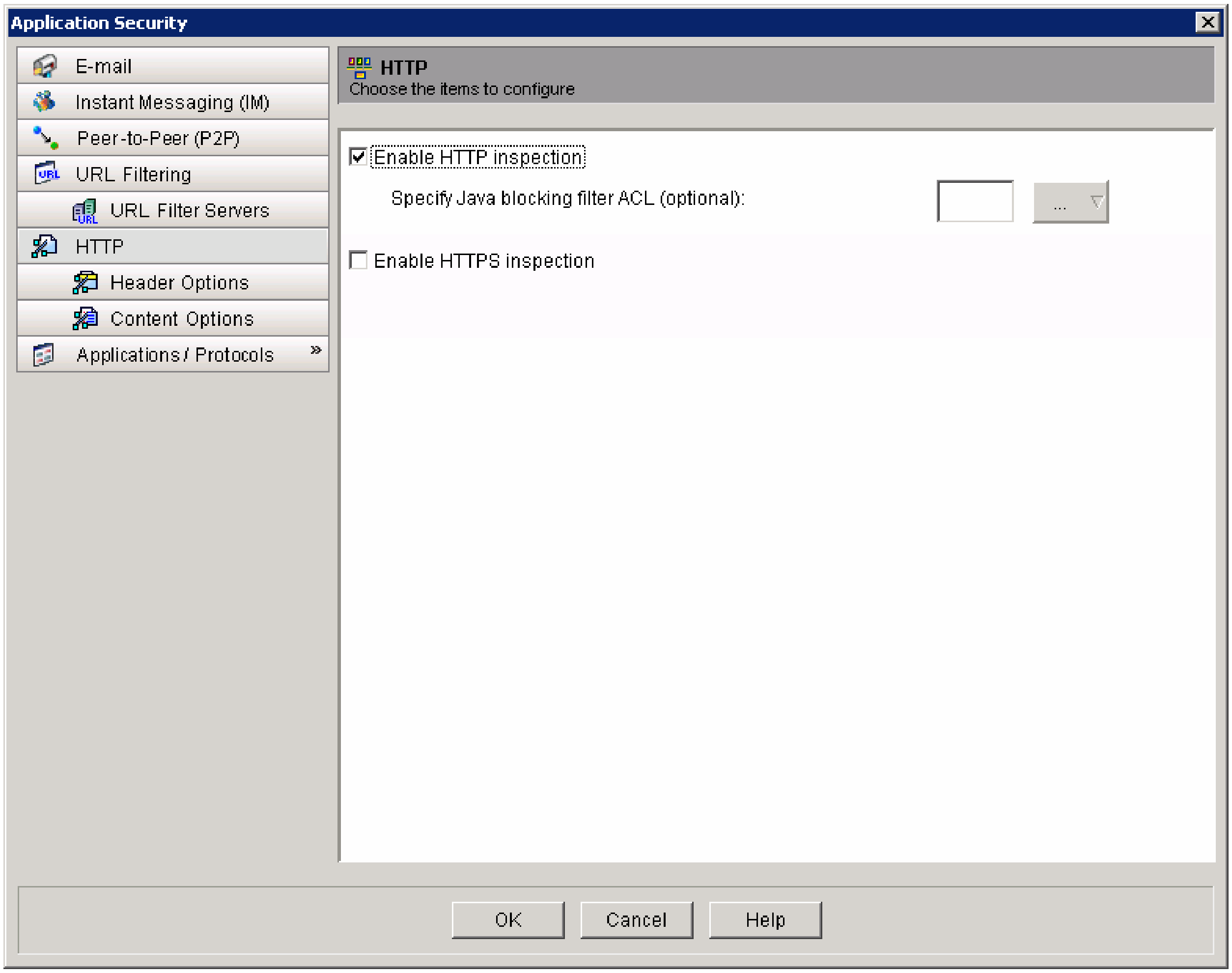
- The router does not identify application-specific traffic. Returns TCP and UDP traffic on sessions initiated inside the firewall.
- Choose this option if you do not need to track use of these applications on the network.

● Use a custom Application Security Policy

Policy Name:

Figure 5-8: SDM Firewall Security Policy Configuration

On the left side, click **HTTP**, and then check **Enable HTTP inspection** on the right to have CBAC make sure that the HTTP packets received are valid.



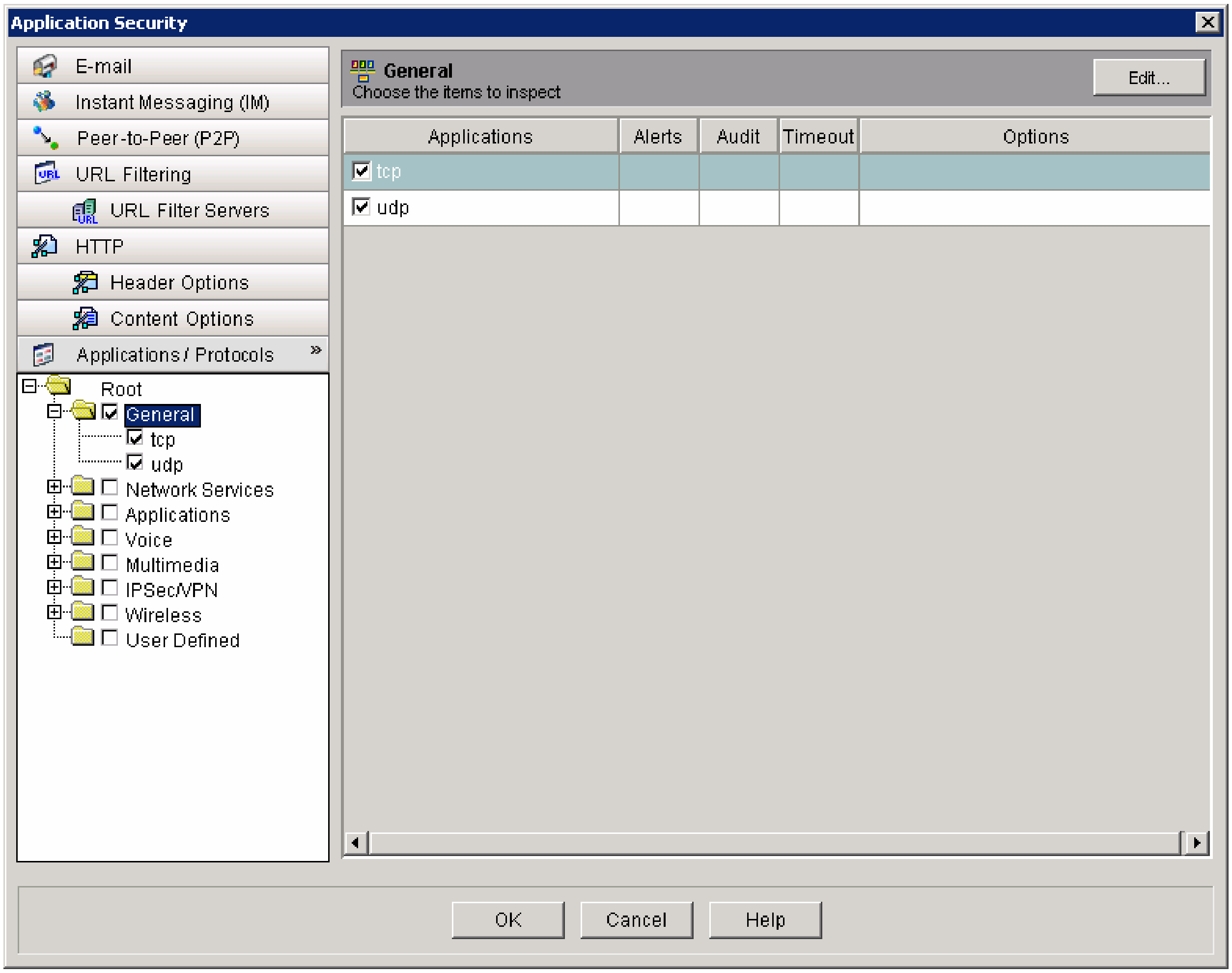
☑ Enable HTTP inspection

Specify Java blocking filter ACL (optional):

☐ Enable HTTPS inspection

Figure 5-9: SDM CBAC Configuration Window, HTTP Options

Click **Applications / Protocols** on the left to expand it. In the tree, select **General** and the **tcp** and **udp** subentries to allow generic TCP and UDP inspection. For TCP connections, it will monitor the TCP state of each connection initiated from the inside, and will only allow return traffic matching that state. For UDP traffic, which is connectionless—and therefore stateless—it will only allow return traffic matching source and destination addresses and UDP ports within a timeout period. Click **OK** to save this new policy.



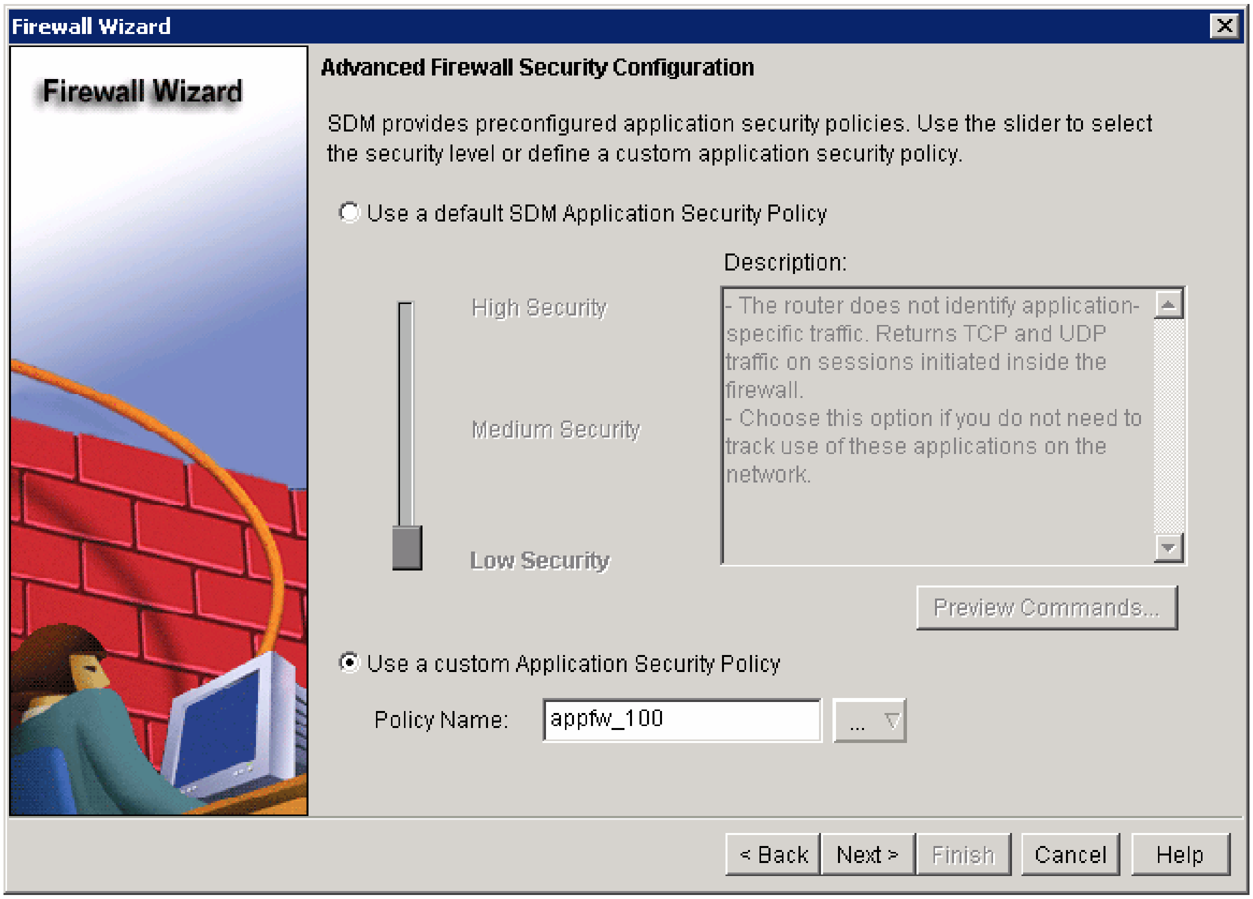
- Root
  - ☑ General
    - ☑ tcp
    - ☑ udp
  - □ Network Services
  - □ Applications
  - □ Voice
  - □ Multimedia
  - □ IPSec/VPN
  - □ Wireless
  - □ User Defined

Applications	Alerts	Audit	Timeout	Options
☑ tcp				
☑ udp				

Figure 5-10: SDM CBAC Configuration Window, General Options

Why is it important to keep track of connection states, especially with TCP connections?

The new policy appears in the text box. Click **Next** to continue.



Firewall Wizard

Advanced Firewall Security Configuration

SDM provides preconfigured application security policies. Use the slider to select the security level or define a custom application security policy.

○ Use a default SDM Application Security Policy

Description:

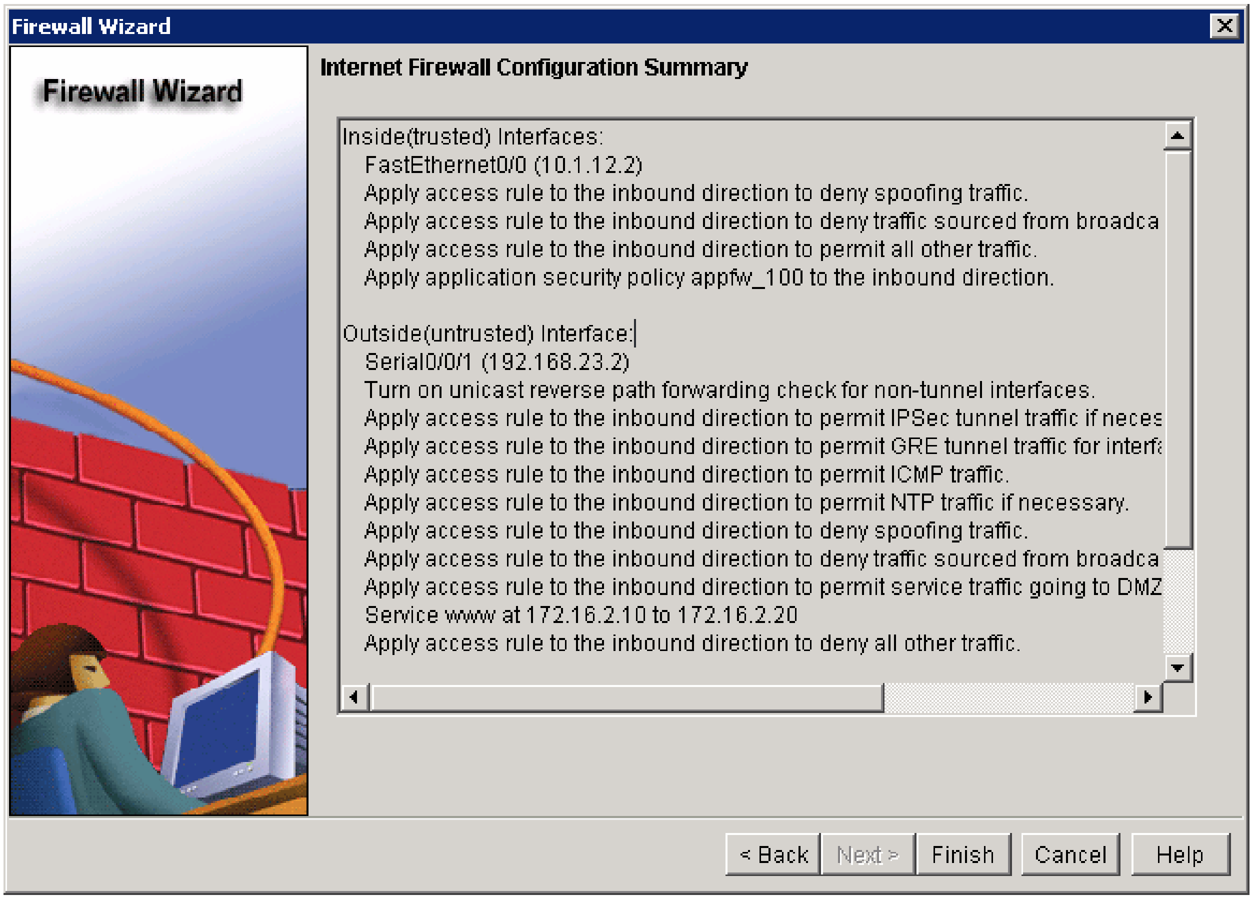
- The router does not identify application-specific traffic. Returns TCP and UDP traffic on sessions initiated inside the firewall.
- Choose this option if you do not need to track use of these applications on the network.

● Use a custom Application Security Policy

Policy Name: appfw_100

Figure 5-11: Applying New Custom Security Policy with SDM

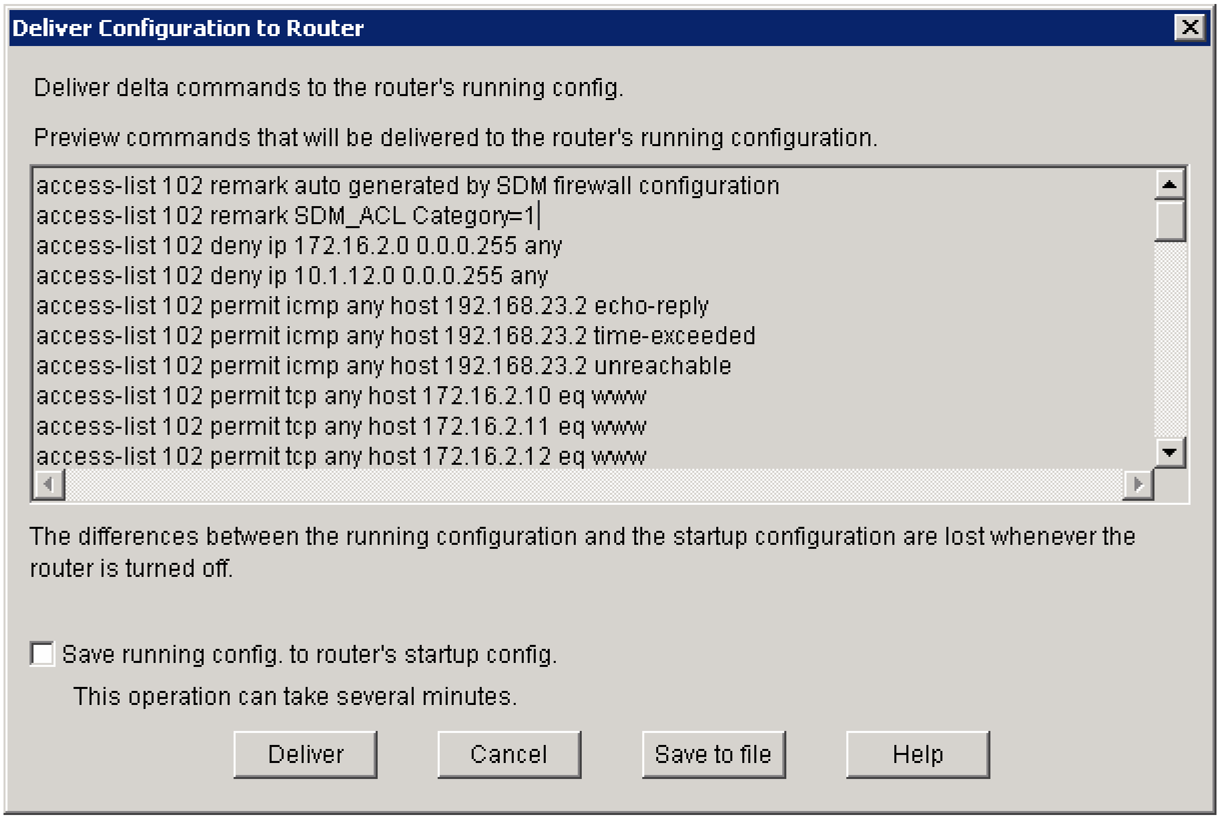
Verify the configuration summary and then click **Finish**.



```
Inside(trusted) Interfaces:
   FastEthernet0/0 (10.1.12.2)
   Apply access rule to the inbound direction to deny spoofing traffic.
   Apply access rule to the inbound direction to deny traffic sourced from broadca
   Apply access rule to the inbound direction to permit all other traffic.
   Apply application security policy appfw_100 to the inbound direction.

Outside(untrusted) Interface:
   Serial0/0/1 (192.168.23.2)
   Turn on unicast reverse path forwarding check for non-tunnel interfaces.
   Apply access rule to the inbound direction to permit IPSec tunnel traffic if neces
   Apply access rule to the inbound direction to permit GRE tunnel traffic for interf
   Apply access rule to the inbound direction to permit ICMP traffic.
   Apply access rule to the inbound direction to permit NTP traffic if necessary.
   Apply access rule to the inbound direction to deny spoofing traffic.
   Apply access rule to the inbound direction to deny traffic sourced from broadca
   Apply access rule to the inbound direction to permit service traffic going to DMZ
   Service www at 172.16.2.10 to 172.16.2.20
   Apply access rule to the inbound direction to deny all other traffic.
```

Figure 5-12: SDM Firewall Configuration Summary

Preview the configuration commands that SDM will apply on the router, and then click **Deliver** to deliver the commands to the router.



Deliver delta commands to the router's running config.

Preview commands that will be delivered to the router's running configuration.

```
access-list 102 remark auto generated by SDM firewall configuration
access-list 102 remark SDM_ACL Category=1
access-list 102 deny ip 172.16.2.0 0.0.0.255 any
access-list 102 deny ip 10.1.12.0 0.0.0.255 any
access-list 102 permit icmp any host 192.168.23.2 echo-reply
access-list 102 permit icmp any host 192.168.23.2 time-exceeded
access-list 102 permit icmp any host 192.168.23.2 unreachable
access-list 102 permit tcp any host 172.16.2.10 eq www
access-list 102 permit tcp any host 172.16.2.11 eq www
access-list 102 permit tcp any host 172.16.2.12 eq www
```

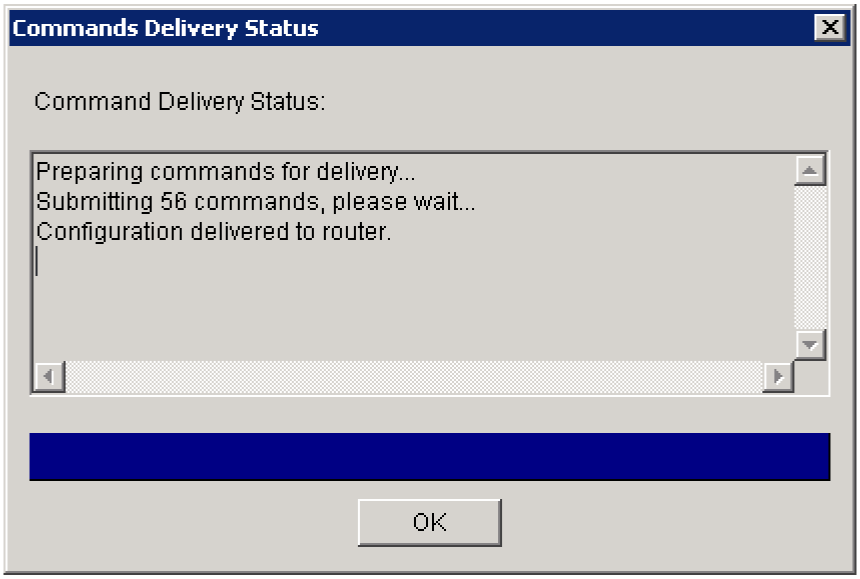
The differences between the running configuration and the startup configuration are lost whenever the router is turned off.

☐ Save running config. to router's startup config.

This operation can take several minutes.

Figure 5-13: SDM Firewall Configuration Delivery Notification

Click **OK** after the commands are delivered to the router.

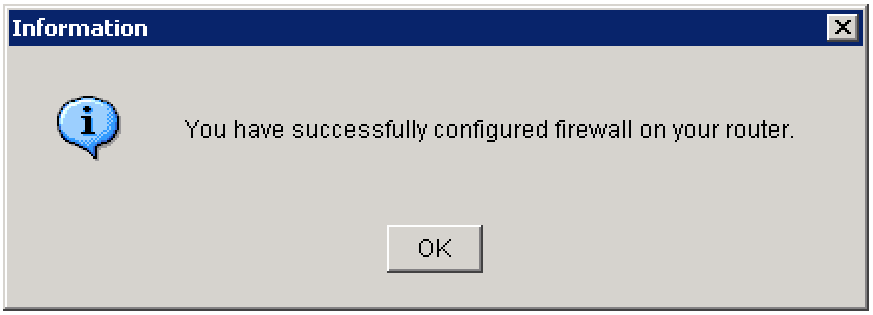


Command Delivery Status:

```
Preparing commands for delivery...
Submitting 56 commands, please wait...
Configuration delivered to router.
```

Figure 5-14: SDM Firewall Configuration Delivery Progress Indicator

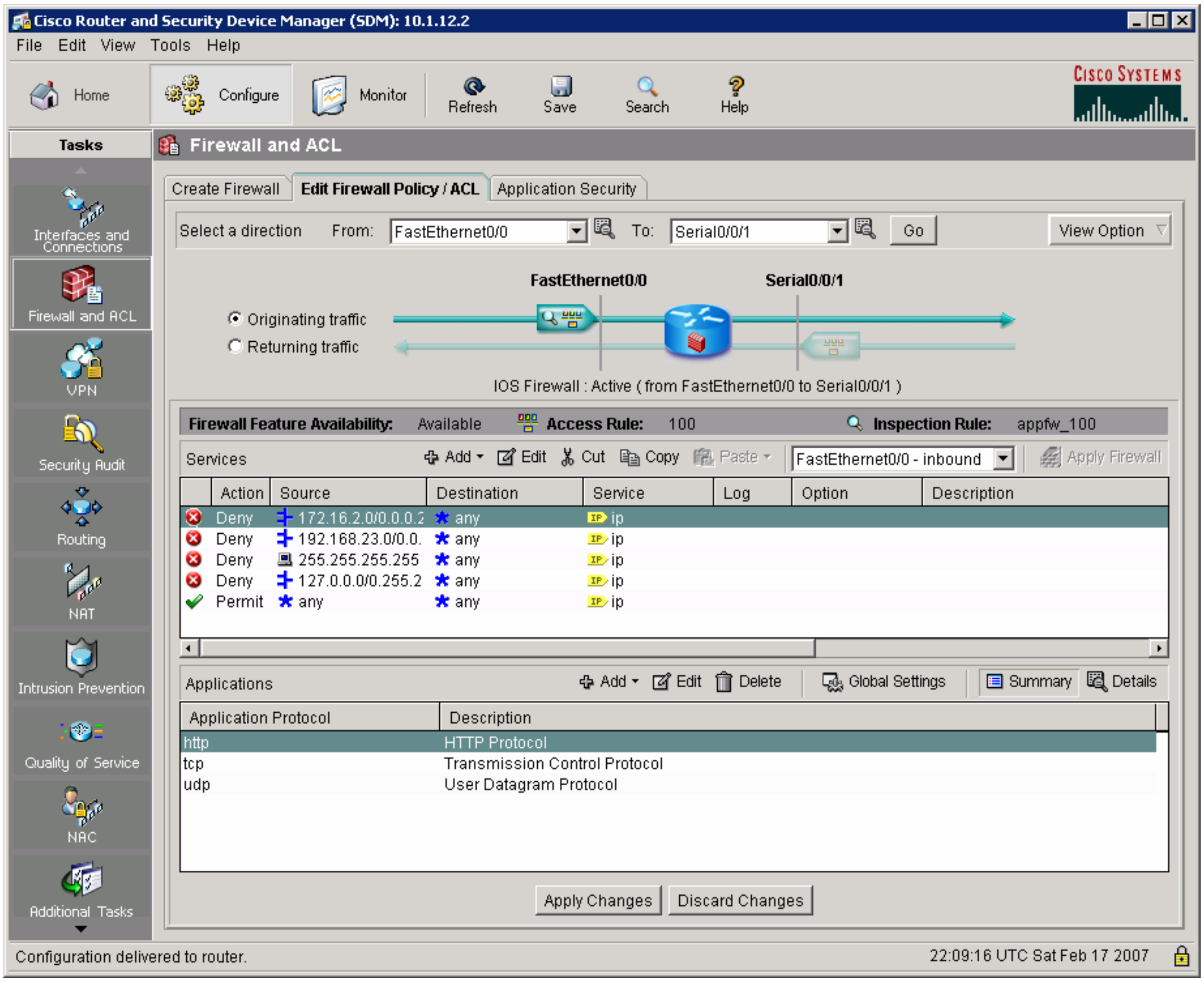
A dialog box informs you that the firewall configuration was successful. Click **OK**. The SDM success message displays.



You have successfully configured firewall on your router.

Figure 5-15: SDM Successful Configuration Notification

Click **OK** and you will be placed back at the **Edit Firewall Policy / ACL** tab in SDM.



Select a direction From: FastEthernet0/0 To: Serial0/0/1

● Originating traffic
○ Returning traffic

IOS Firewall : Active (from FastEthernet0/0 to Serial0/0/1)

Firewall Feature Availability: Available Access Rule: 100 Inspection Rule: appfw_100

Services FastEthernet0/0 - inbound

Action	Source	Destination	Service	Log	Option	Description
Deny	172.16.2.0/0.0.0.2	any	ip			
Deny	192.168.23.0/0.0.	any	ip			
Deny	255.255.255.255	any	ip			
Deny	127.0.0.0/0.255.2	any	ip			
Permit	any	any	ip			

Applications

Application Protocol	Description
http	HTTP Protocol
tcp	Transmission Control Protocol
udp	User Datagram Protocol

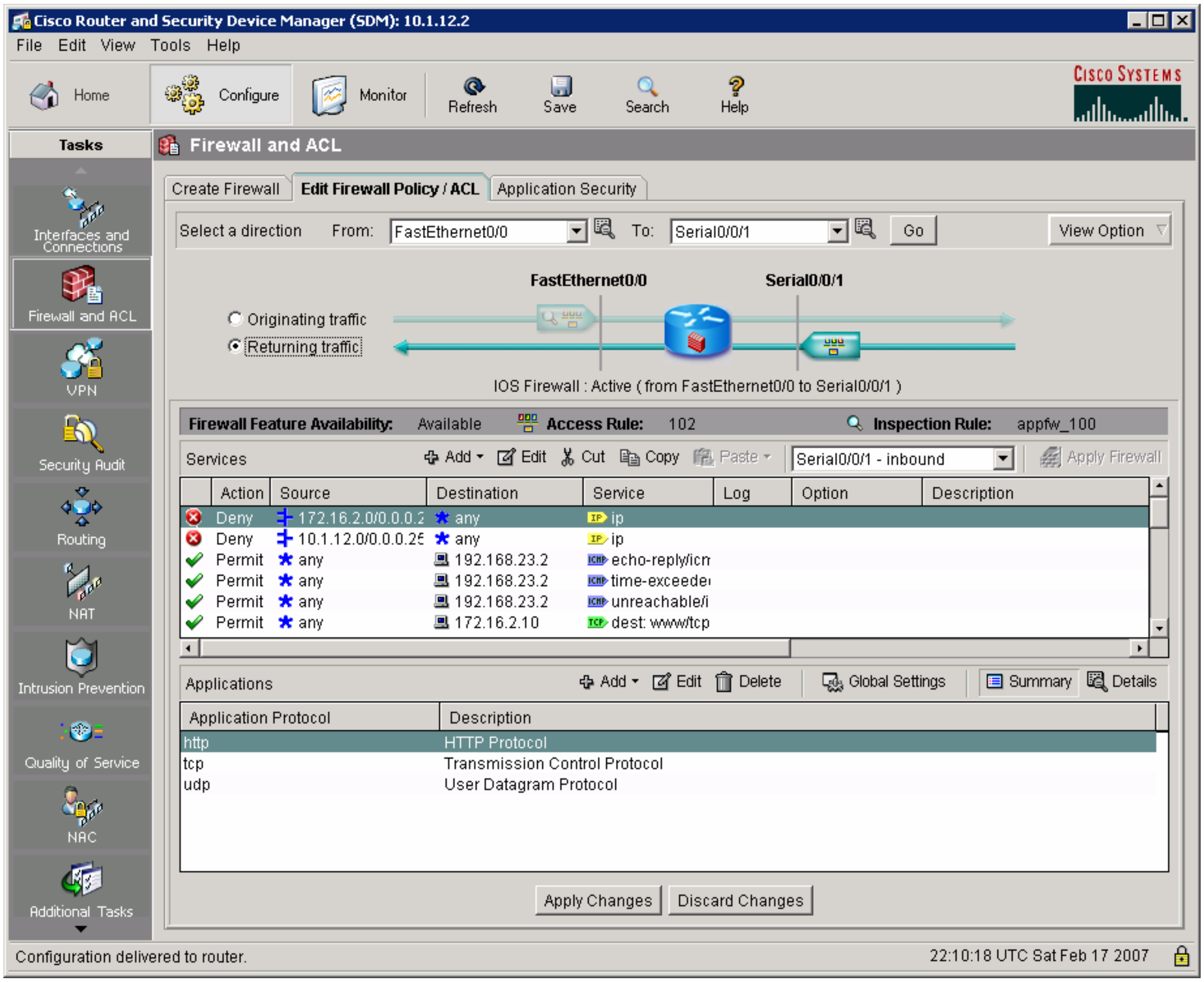
Configuration delivered to router. 22:09:16 UTC Sat Feb 17 2007

Figure 5-16: SDM Firewall Policy Summary for Originating Traffic

Step 6: Modify the Firewall Configuration

In which direction and to which interface were most of the access rules applied in the previous step?

Select **Returning Traffic** in the upper part of the tab to view the access rules configured for returning traffic passing through the router from the outside interface to the inside interface.



Select a direction From: FastEthernet0/0 To: Serial0/0/1

IOS Firewall : Active (from FastEthernet0/0 to Serial0/0/1)

Firewall Feature Availability: Available Access Rule: 102 Inspection Rule: appfw_100

Services: Serial0/0/1 - inbound

Action	Source	Destination	Service	Log	Option	Description
Deny	172.16.2.0/0.0.0.2	any	ip			
Deny	10.1.12.0/0.0.0.25	any	ip			
Permit	any	192.168.23.2	echo-reply/icn			
Permit	any	192.168.23.2	time-exceede			
Permit	any	192.168.23.2	unreachable/i			
Permit	any	172.16.2.10	dest: www/tcp			

Application Protocol	Description
http	HTTP Protocol
tcp	Transmission Control Protocol
udp	User Datagram Protocol

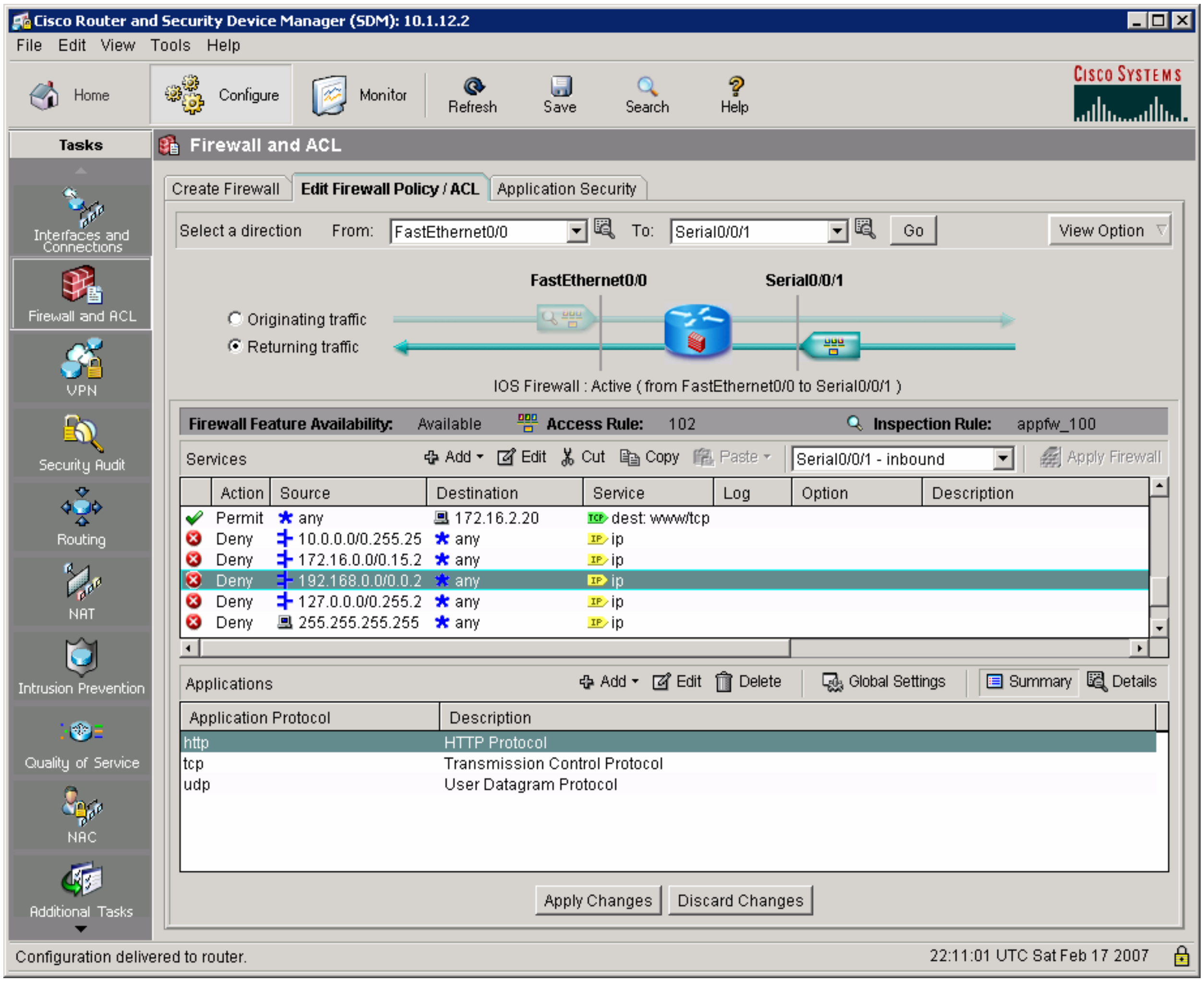
Configuration delivered to router. 22:10:18 UTC Sat Feb 17 2007

Figure 6-1: SDM Firewall Policy Summary for Returning Traffic

Scroll down the access list to the deny statement for the 192.168.0.0/16 networks.

Why is this statement incompatible for this scenario?

Based on other elements in the access list, why do you think that this statement was added?



Cisco Router and Security Device Manager (SDM): 10.1.12.2

File Edit View Tools Help

Home | Configure | Monitor | Refresh | Save | Search | Help

Tasks: Interfaces and Connections, Firewall and ACL, VPN, Security Audit, Routing, NAT, Intrusion Prevention, Quality of Service, NAC, Additional Tasks

Firewall and ACL

Create Firewall | Edit Firewall Policy / ACL | Application Security

Select a direction From: FastEthernet0/0 To: Serial0/0/1 Go | View Option

FastEthernet0/0 Serial0/0/1

○ Originating traffic
● Returning traffic

IOS Firewall : Active (from FastEthernet0/0 to Serial0/0/1)

Firewall Feature Availability: Available Access Rule: 102 Inspection Rule: appfw_100

Services: Add | Edit | Cut | Copy | Paste | Serial0/0/1 - inbound | Apply Firewall

	Action	Source	Destination	Service	Log	Option	Description
✓	Permit	any	172.16.2.20	dest: www/tcp			
☒	Deny	10.0.0.0/0.255.25	any	ip			
☒	Deny	172.16.0.0/0.15.2	any	ip			
☒	Deny	192.168.0.0/0.0.2	any	ip			
☒	Deny	127.0.0.0/0.255.2	any	ip			
☒	Deny	255.255.255.255	any	ip			

Applications: Add | Edit | Delete | Global Settings | Summary | Details

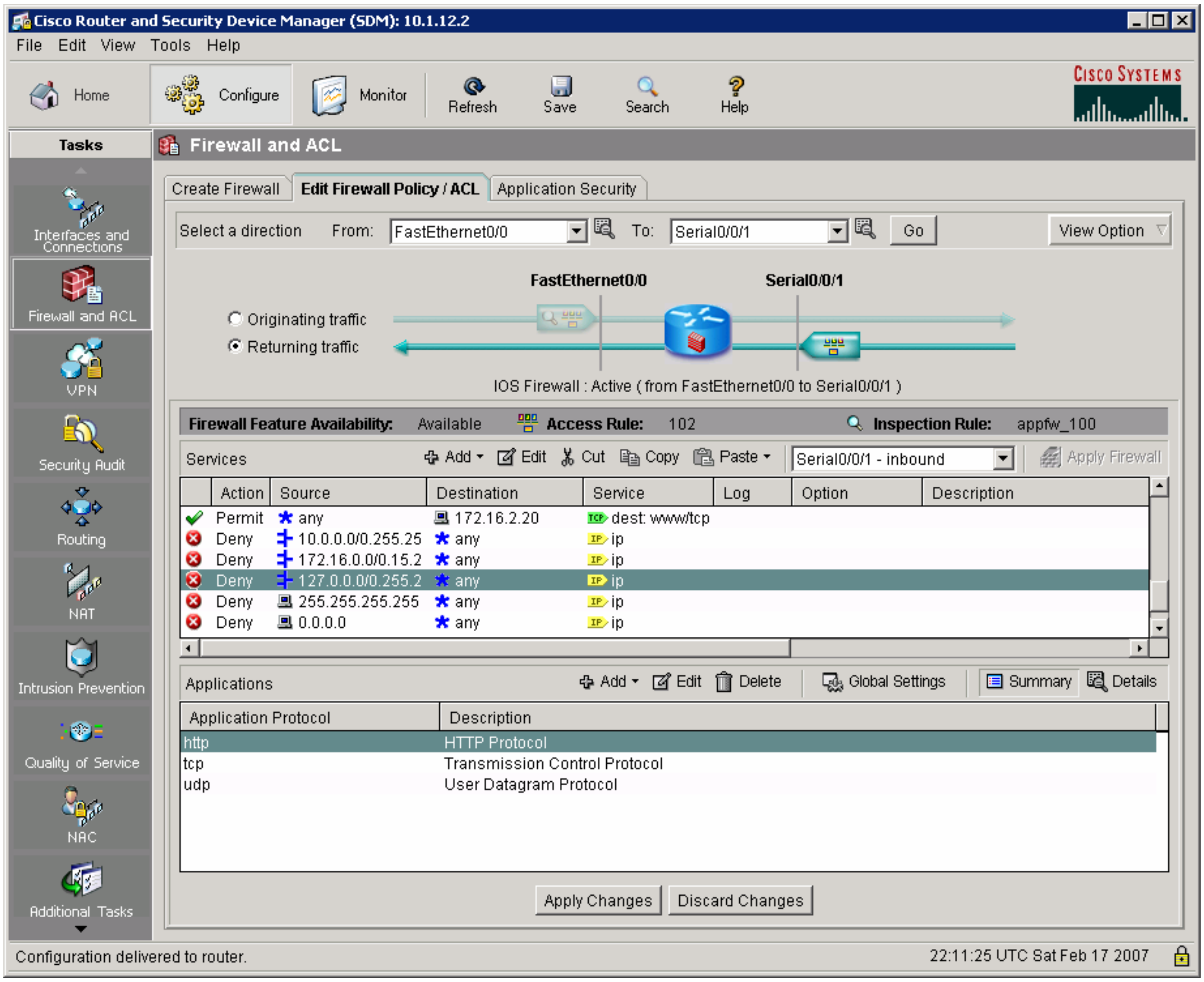
Application Protocol	Description
http	HTTP Protocol
tcp	Transmission Control Protocol
udp	User Datagram Protocol

Apply Changes | Discard Changes

Configuration delivered to router. 22:11:01 UTC Sat Feb 17 2007

Figure 6-2: Deny Statement in ACL 102 for 192.168.0.0/16

Select the 192.168.0.0/16 access rule statement and then click **Cut** to remove the statement. Verify that the statement is removed, and then click **Apply Changes**.



Select a direction From: FastEthernet0/0 To: Serial0/0/1

○ Originating traffic
● Returning traffic

IOS Firewall : Active (from FastEthernet0/0 to Serial0/0/1)

Firewall Feature Availability: Available Access Rule: 102 Inspection Rule: appfw_100

Services Serial0/0/1 - inbound

	Action	Source	Destination	Service	Log	Option	Description
✓	Permit	any	172.16.2.20	dest: www/tcp			
☒	Deny	10.0.0.0/0.255.25	any	ip			
☒	Deny	172.16.0.0/0.15.2	any	ip			
☒	Deny	127.0.0.0/0.255.2	any	ip			
☒	Deny	255.255.255.255	any	ip			
☒	Deny	0.0.0.0	any	ip			

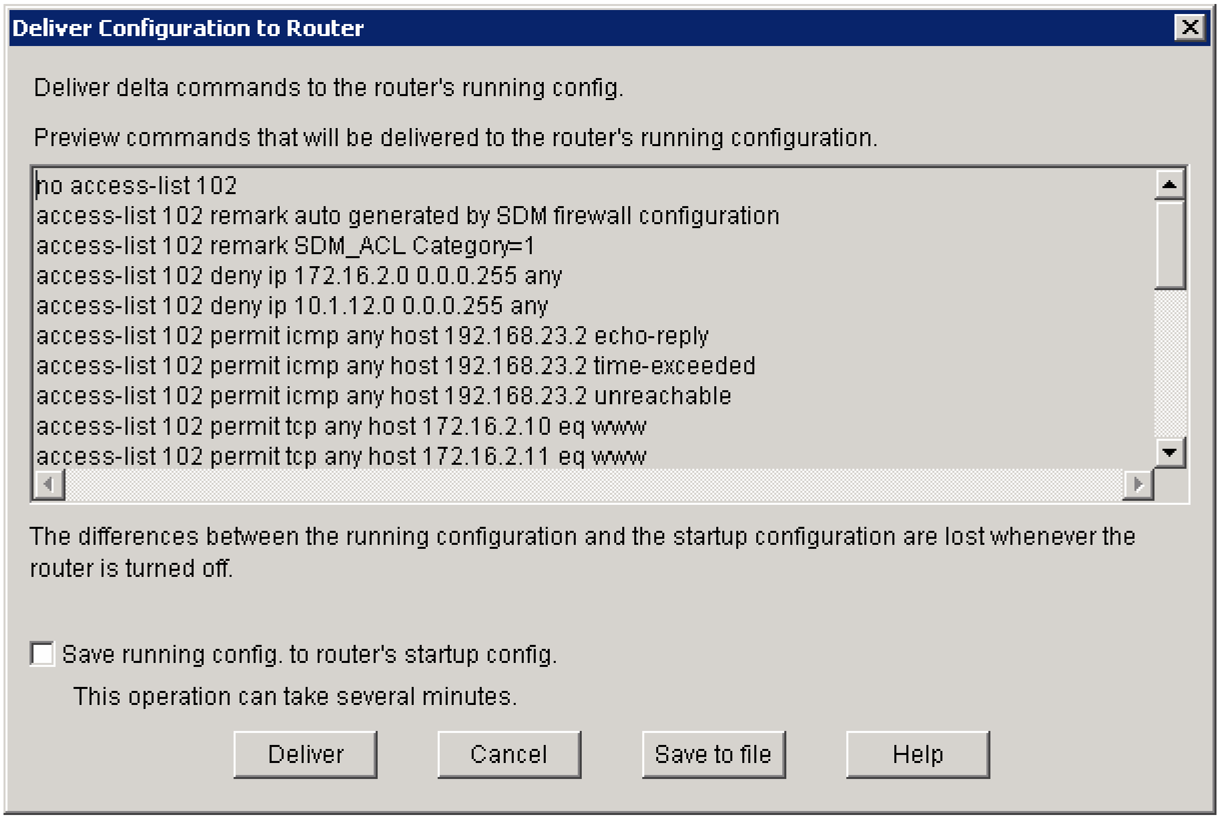
Applications

Application Protocol	Description
http	HTTP Protocol
tcp	Transmission Control Protocol
udp	User Datagram Protocol

Configuration delivered to router. 22:11:25 UTC Sat Feb 17 2007

Figure 6-3: ACL 102 After Deleting the Deny Statement for 192.168.0.0/16

Preview the commands that SDM will send to the router, and then click **Deliver**.



Deliver delta commands to the router's running config.

Preview commands that will be delivered to the router's running configuration.

```
no access-list 102
access-list 102 remark auto generated by SDM firewall configuration
access-list 102 remark SDM_ACL Category=1
access-list 102 deny ip 172.16.2.0 0.0.0.255 any
access-list 102 deny ip 10.1.12.0 0.0.0.255 any
access-list 102 permit icmp any host 192.168.23.2 echo-reply
access-list 102 permit icmp any host 192.168.23.2 time-exceeded
access-list 102 permit icmp any host 192.168.23.2 unreachable
access-list 102 permit tcp any host 172.16.2.10 eq www
access-list 102 permit tcp any host 172.16.2.11 eq www
```

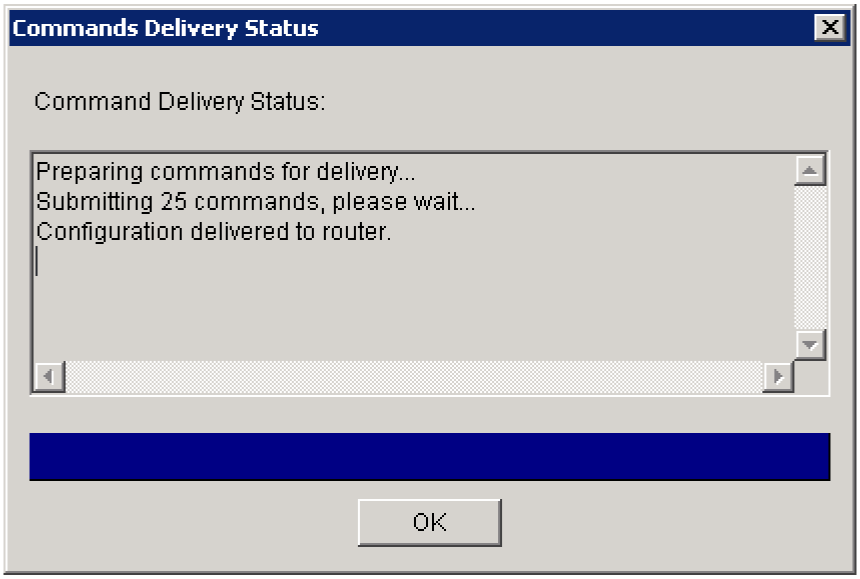
The differences between the running configuration and the startup configuration are lost whenever the router is turned off.

☐ Save running config. to router's startup config.

This operation can take several minutes.

Figure 6-4: SDM Firewall Configuration Delivery Notification

Click **OK** after the commands have been delivered.



Command Delivery Status:

```
Preparing commands for delivery...
Submitting 25 commands, please wait...
Configuration delivered to router.
```

Figure 6-5: Figure 5-14: SDM Firewall Configuration Delivery Progress Indicator

Step 7: Monitor Firewall Activity

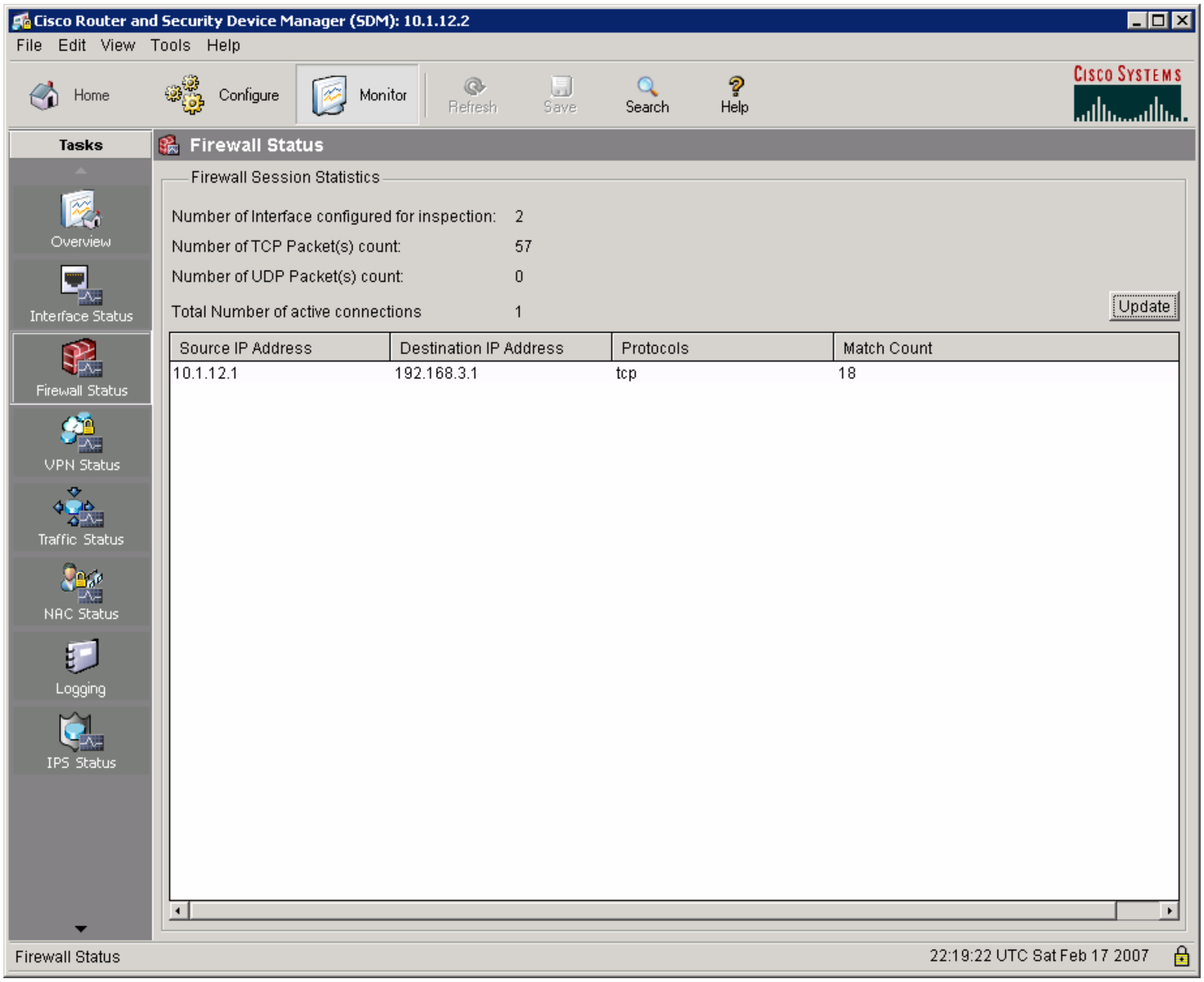
On the ISP router, add the password “cisco” to the virtual terminal lines (VTYs) to allow an authenticated Telnet session to the router.

```
ISP(config)# line vty 0 4
ISP(config-line)# password cisco
ISP(config-line)# login
```

On the INSIDE router, establish a Telnet session to the ISPs loopback interface.

```
INSIDE# telnet 192.168.3.1
Trying 192.168.3.1 ... Open


User Access Verification

Password:
ISP>
```

While the Telnet session is active, click the SDM **Monitor** tab and then click **Firewall Status** on the left side. Click **Update** to get the latest firewall statistics. You should see the active Telnet session in the list.



Firewall Session Statistics

Number of Interface configured for inspection: 2
Number of TCP Packet(s) count: 57
Number of UDP Packet(s) count: 0
Total Number of active connections 1

Source IP Address	Destination IP Address	Protocols	Match Count
10.1.12.1	192.168.3.1	tcp	18

Figure 7-1: SDM Firewall Monitor

After displaying the parameters of the Telnet session through the SDM monitor, close the session by issuing the **exit** command.

```
ISP> exit

[Connection to 192.168.3.1 closed by foreign host]
INSIDE#
```

From the ISP router, initiate a Telnet session with the INSIDE loopback interface. Do not set up a line password. We want the Telnet session to fail.

```
ISP# telnet 10.1.1.1
Trying 10.1.1.1 ...
% Destination unreachable; gateway or host down

ISP#
```

Why does this Telnet session fail?

On the FW router, a log message will appear indicating that a Telnet session was denied. The log message appears because the **log** keyword was used in the access list applied to the outside interface.

```
FW#
*Feb 17 22:21:01.085: %SEC-6-IPACCESSLOGP: list 102 denied tcp
192.168.23.3(20650) -> 10.1.1.1(23), 1 packet
```

View the configuration on the FW outside interface using the **show run** command.

```
FW# show run interface serial0/0/1
Building configuration...

Current configuration : 168 bytes
!
interface Serial0/0/1
 description $FW_OUTSIDE$
 ip address 192.168.23.2 255.255.255.0
 ip access-group 102 in
 ip verify unicast reverse-path
 clock rate 64000
end
```

From the **show access-lists** *number* command, you can see that the **log** part of the access list is applied inbound to the outside interface.

```
FW# show access-lists 102
Extended IP access list 102
    10 deny ip 172.16.2.0 0.0.0.255 any
    20 deny ip 10.1.12.0 0.0.0.255 any
    30 permit icmp any host 192.168.23.2 echo-reply
    40 permit icmp any host 192.168.23.2 time-exceeded
    50 permit icmp any host 192.168.23.2 unreachable
    60 permit tcp any host 172.16.2.10 eq www
    70 permit tcp any host 172.16.2.11 eq www
    80 permit tcp any host 172.16.2.12 eq www
    90 permit tcp any host 172.16.2.13 eq www
    100 permit tcp any host 172.16.2.14 eq www
    110 permit tcp any host 172.16.2.15 eq www
    120 permit tcp any host 172.16.2.16 eq www
    130 permit tcp any host 172.16.2.17 eq www
    140 permit tcp any host 172.16.2.18 eq www
    150 permit tcp any host 172.16.2.19 eq www
    160 permit tcp any host 172.16.2.20 eq www
    170 deny ip 10.0.0.0 0.255.255.255 any
    180 deny ip 172.16.0.0 0.15.255.255 any
    190 deny ip 127.0.0.0 0.255.255.255 any
    200 deny ip host 255.255.255.255 any
    210 deny ip host 0.0.0.0 any
    220 deny ip any any log (1 match)
```

What are access control entries with sequence numbers 170–190 intended to prevent?

What other command applied in the SDM firewall configuration prevents the same type of attack?

Conclusion

Given the above ACL configuration, predict what will happen in the following scenarios and explain the reason for each.

- ISP pings 10.1.1.1

- ISP pings 172.16.23.2

- ISP pings 172.16.2.1

- Imagine that ISP pings an existing web server at 172.16.2.10

- Imagine that a network connected to ISP initiates an HTTP session with a web server at 172.16.2.10

Final Configuration

```
INSIDE# show run
hostname INSIDE
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 10.1.12.1 255.255.255.0
 no shutdown
!
router eigrp 1
 network 10.0.0.0
 no auto-summary
!
end

FW# show run
hostname FW
```

```
!
ip inspect name appfw_100 http
ip inspect name appfw_100 tcp
ip inspect name appfw_100 udp
ip inspect name dmzinspect tcp
ip inspect name dmzinspect udp
!
crypto pki trustpoint TP-self-signed-3043721146
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-3043721146
 revocation-check none
 rsakeypair TP-self-signed-3043721146
!
crypto pki certificate chain TP-self-signed-3043721146
 certificate self-signed 01
  3082023A 308201A3 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303433 37323131 3436301E 170D3037 30323137 32313336
  33375A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D33 30343337
  32313134 3630819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100EAE7 7D716529 0B5A56C3 C519EE38 AF5BB46C 20C2D045 8CF5E575 F31CB907
  AD9EC6D2 25EBE142 6994982A EFF4565D B5C8DD3C 3FD64334 6F447CD9 3763FBB8
  6D3A8583 C91C8F9E 9F69716F CD773448 52CA343E FE4CC690 F8D74D1C A8EAC14E
  AE30BFCF 9248BD02 908ADF7D 02F855BD 71269846 6CAFA205 9EDDD7C0 2F8A240D
  D6D50203 010001A3 62306030 0F060355 1D130101 FF040530 030101FF 300D0603
  551D1104 06300482 02465730 1F060355 1D230418 30168014 1C974D6A 53CA1D26
  96919DA4 9F55F630 6986CBAD 301D0603 551D0E04 1604141C 974D6A53 CA1D2696
  919DA49F 55F63069 86CBAD30 0D06092A 864886F7 0D010104 05000381 81000566
  95264590 A35B0A19 89F68CEC DEA11FF0 B27FD67A 577404A3 C8EEDD99 BDFCFF33
  B05ACFE5 9EF70D3F 13B35864 35CA069F 9D4DB369 6E4262D6 E77683F2 CB3B36DC
  13A5208C F85CA99B 4BB7E81F BF89C43D 1EC91E03 322EC2FA 28763141 C63E9AA3
  45CE31B4 65B90AE6 D7EC8878 08CB30ED A4D24389 10B66C8E C7611D56 048D
  quit
username ciscosdm privilege 15 password 0 ciscosdm
!
interface Loopback0
 description $FW_DMZ$
 ip address 172.16.2.1 255.255.255.0
 ip access-group 101 in
 ip inspect dmzinspect out
!
interface FastEthernet0/0
 description $FW_INSIDE$
 ip address 10.1.12.2 255.255.255.0
 ip access-group 100 in
 ip inspect appfw_100 in
 no shutdown
!
interface Serial0/0/1
 description $FW_OUTSIDE$
 ip address 192.168.23.2 255.255.255.0
 ip access-group 102 in
 ip verify unicast reverse-path
 clock rate 64000
 no shutdown
!
router eigrp 1
 redistribute static
 passive-interface Loopback0
 network 10.0.0.0
 network 172.16.0.0
 no auto-summary
```

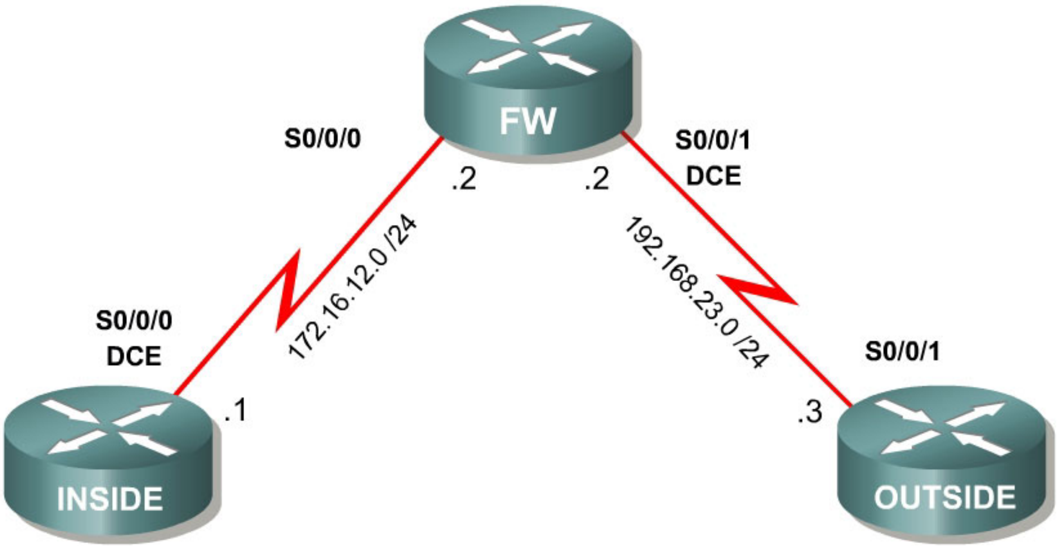
```
!
ip route 0.0.0.0 0.0.0.0 192.168.23.3
!
ip http server
ip http authentication local
ip http secure-server
!
access-list 100 remark auto generated by SDM firewall configuration
access-list 100 remark SDM_ACL Category=1
access-list 100 deny   ip 172.16.2.0 0.0.0.255 any
access-list 100 deny   ip 192.168.23.0 0.0.0.255 any
access-list 100 deny   ip host 255.255.255.255 any
access-list 100 deny   ip 127.0.0.0 0.255.255.255 any
access-list 100 permit ip any any
access-list 101 remark auto generated by SDM firewall configuration
access-list 101 remark SDM_ACL Category=1
access-list 101 deny   ip any any log
access-list 102 remark auto generated by SDM firewall configuration
access-list 102 remark SDM_ACL Category=1
access-list 102 deny   ip 172.16.2.0 0.0.0.255 any
access-list 102 deny   ip 10.1.12.0 0.0.0.255 any
access-list 102 permit icmp any host 192.168.23.2 echo-reply
access-list 102 permit icmp any host 192.168.23.2 time-exceeded
access-list 102 permit icmp any host 192.168.23.2 unreachable
access-list 102 permit tcp any host 172.16.2.10 eq www
access-list 102 permit tcp any host 172.16.2.11 eq www
access-list 102 permit tcp any host 172.16.2.12 eq www
access-list 102 permit tcp any host 172.16.2.13 eq www
access-list 102 permit tcp any host 172.16.2.14 eq www
access-list 102 permit tcp any host 172.16.2.15 eq www
access-list 102 permit tcp any host 172.16.2.16 eq www
access-list 102 permit tcp any host 172.16.2.17 eq www
access-list 102 permit tcp any host 172.16.2.18 eq www
access-list 102 permit tcp any host 172.16.2.19 eq www
access-list 102 permit tcp any host 172.16.2.20 eq www
access-list 102 deny   ip 10.0.0.0 0.255.255.255 any
access-list 102 deny   ip 172.16.0.0 0.15.255.255 any
access-list 102 deny   ip 127.0.0.0 0.255.255.255 any
access-list 102 deny   ip host 255.255.255.255 any
access-list 102 deny   ip host 0.0.0.0 any
access-list 102 deny   ip any any log
!
line vty 0 4
 login local
end

ISP# show run
hostname ISP
!
interface Loopback0
 ip address 192.168.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 no shutdown
!
ip route 10.0.0.0 255.0.0.0 192.168.23.2
ip route 172.16.0.0 255.255.0.0 192.168.23.2
!
line vty 0 4
 password cisco
 login
end
```

Lab 6.2 Configuring CBAC

Learning Objectives

- Configure CBAC rules on a router
- Apply CBAC rules on a router

Topology Diagram



Scenario

Context-based access control (CBAC) is a powerful tool in the Cisco IOS Firewall feature set. It allows stateful packet inspection of certain types of attacks. In this lab, INSIDE represents an inside corporate router, OUTSIDE represents an outside Internet or ISP router, and FW represents the corporate firewall. CBAC alone is not enough to make a router into a secure Internet firewall, but in addition to other security features it can be a very powerful defense.

Step 1: Configure the Physical Interfaces

Configure the loopback interfaces with the addresses shown in the topology diagram. Also configure the serial interfaces shown in the diagram. Set the clock rate on the appropriate interface and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command.

```
INSIDE(config)# interface serial0/0/0
INSIDE(config-if)# ip address 172.16.12.1 255.255.255.0
INSIDE(config-if)# clockrate 64000
INSIDE(config-if)# no shutdown

FW(config)# interface serial0/0/0
FW(config-if)# ip address 172.16.12.2 255.255.255.0
FW(config-if)# no shutdown
FW(config-if)# interface serial0/0/1
FW(config-if)# ip address 192.168.23.2 255.255.255.0
FW(config-if)# clockrate 64000
FW(config-if)# no shutdown

OUTSIDE(config)# interface serial0/0/1
OUTSIDE(config-if)# ip address 192.168.23.3 255.255.255.0
OUTSIDE(config-if)# no shutdown
```

Step 2: Configure Static Default Routes

On the INSIDE and OUTSIDE routers, configure static default routes directing traffic to unknown destinations to be forwarded to the FW router. FW will not need any routes because it has interfaces directly connected to both networks (as shown in the topology diagram.)

```
INSIDE(config)# ip route 0.0.0.0 0.0.0.0 172.16.12.2

OUTSIDE(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2
```

Your network should have full IP connectivity at this point. If it does not have full connectivity, troubleshoot.

Normally, a single-homed company might use Network Address Translation (NAT) at its corporate edge to protect its network and allow private addressing within the bounds of its network. In that case, the OUTSIDE router, normally a provider edge router would have a static route directing traffic to the address owned by the customer out one of its interfaces. In this scenario you will not configure NAT, and you will use a default route for simplicity.

Step 3: Enable Telnet Access

You will be using the Telnet protocol to test connectivity in this lab scenario. In order to enable Telnet access on a router beginning with its default configuration, simply apply the **password** *string* command on the virtual terminal lines.

Apply this configuration change on the INSIDE and OUTSIDE routers. Use “cisco” as the line password. This will be used later for verification purposes.

```
INSIDE(config)# line vty 0 4
INSIDE(config-line)# password cisco
INSIDE(config-line)# login

OUTSIDE(config)# line vty 0 4
OUTSIDE(config-line)# password cisco
OUTSIDE(config-line)# login
```

Step 4: Create IP Inspect Rules

CBAC operates by statefully inspecting some protocols and tracking TCP connections and UDP flows. CBAC examines the protocols to determine if incoming, untrusted (outside) traffic is return traffic for an inside-initiated connection, or the result of arbitrarily spoofed packets. For some well-known protocols, CBAC can also examine particular application-layer fields to make sure that the packets are following the protocols of those specific applications correctly. Any traffic that is not accepted by CBAC is treated appropriately according to the rules indicated by the access list on the interface. This is done by explicitly blocking untrusted traffic (which we will configure later) except when allowed by CBAC.

Why is it important to keep track of connection states, especially with TCP connections?

The critical part of configuring CBAC involves creating rules to track connections and flows. Create rules to track TCP and UDP flows using the **ip inspect name** *name protocol* command. Use the name “myrules” and apply the CBAC rule to the to Serial0/0/0 interface in the inbound direction. To see the protocols available (most of the protocols listed will be application layer protocols), enter the **ip inspect name** *name* command followed by the **?** character. Newer IOS versions will have more protocols listed.

```
FW(config)# ip inspect name myrules ?
  802-11-iapp       IEEE 802.11 WLANs WG IAPP
  ace-svr           ACE Server/Propagation
  aol               America-Online
  appfw             Application Firewall
  appleqtc          Apple QuickTime
  bgp               Border Gateway Protocol
  <OUTPUT OMITTED>

FW(config)# ip inspect name myrules tcp
FW(config)# ip inspect name myrules udp
```

You can also set CBAC timeouts for various protocols. To change the amount of time that should pass before a UDP flow times out, use the **ip inspect udp idle-time** *timeout* command in global configuration mode. The default UDP idle timeout is 30 seconds. Change the UDP timeout to 60 seconds.

```
FW(config)# ip inspect udp idle-time 60
```

Why is this particularly important for UDP protocols?

On a per-protocol basis, there are other adjustable settings. For instance, you can manipulate CBAC to trigger logging messages based on the matched protocol. This is important for security accounting as well as for debugging purposes. View the options available on a per-protocol basis, using the **?** character.

```
FW(config)# ip inspect name myrules tcp ?
  alert           Turn on/off alert
  audit-trail     Turn on/off audit trail
  router-traffic  Enable inspection of sessions to/from the router
  timeout         Specify the inactivity timeout time
  <cr>
```

In a secure network, you would likely set up a Syslog server to monitor security information including communication to external networks. Alert and audit trail messages allow holes in the firewall created by CBAC to be monitored and logged for later use. By default, CBAC logs alert messages to the console which can be configured on a per-protocol basis to override the global settings for the alert messages (as shown above). To change the global setting for alerts, use the command **ip inspect alert-off**. By default, alerts are on. To enable audit-trail messages, use the global command **ip inspect audit-trail**. By default, audit-trail messages are off. The **timeout** argument specifies a per-protocol connection timeout period. Add in Internet Control Message Protocol (ICMP) with a timeout time of 5 seconds, HTTP inspection without alerting, and FTP inspection with an audit-trail. ICMP inspection may not work on older IOS releases.

```
FW(config)# ip inspect name myrules icmp timeout 5
FW(config)# ip inspect name myrules http alert off
FW(config)# ip inspect name myrules ftp audit-trail on
```

To apply the rule set to an interface, use the interface level command **ip inspect** *name direction*. Apply “myrules” to the inside interface on FW with an inbound direction. This means that any traffic initiated from the inside interface going through the router will have IP inspection performed on it.

```
FW(config)# interface serial0/0/0
FW(config-if)# ip inspect myrules in
```

In this scenario, you could also apply it outbound on the outside interface to achieve the same effect. When would this not apply?

Step 5: Block Unwanted Outside Traffic

Configure an extended access list to deny any traffic coming in the outside interface. The access list must be extended because CBAC needs to open up temporary holes in it for return traffic and cannot do this with standard access lists. Also have the **deny** portion of the access-list log packets that are blocked. Apply this access list to be inbound on the outside interface on the firewall.

NOTE: If you are using an older IOS release that did not accept ICMP inspection earlier, you may want to add the statement **access-list 100 permit icmp any any** before the deny statement in this access list to allow all ICMP traffic to go through (since it will not be inspected by CBAC).

```
FW(config)# access-list 100 deny ip any any log
FW(config)# interface serial0/0/1
FW(config-if)# ip access-group 100 in
```

Step 6: Verify CBAC Operation

Telnet from OUTSIDE to INSIDE. This should fail.

```
OUTSIDE# telnet 172.16.12.1
Trying 172.16.12.1 ...
% Destination unreachable; gateway or host down

OUTSIDE#
```

In addition, you should see a log message appear on FW. This log message is not from CBAC but instead from the access list denying the packet.

```
FW#
*Feb 18 02:11:11.823: %SEC-6-IPACCESSLOGP: list 100 denied tcp 192.168.23.3(0)
-> 172.16.12.1(0), 1 packet
```

Now, attempt to telnet from INSIDE to OUTSIDE. Leave the telnet session open so you can verify the connection on FW.

```
INSIDE# telnet 192.168.23.3
Trying 192.168.23.3 ... Open


User Access Verification

Password:
OUTSIDE>
```

On FW, issue the **show ip inspect all** command to see the configuration and operation of CBAC. Notice the inspected TCP connection between INSIDE and OUTSIDE is listed at the end.

```
FW# show ip inspect all
Session audit trail is disabled
Session alert is enabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
```

```
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
 Inspection name myrules
    tcp alert is on audit-trail is off timeout 3600
    udp alert is on audit-trail is off timeout 30
    icmp alert is on audit-trail is off timeout 5
    http alert is off audit-trail is off timeout 3600
    ftp alert is on audit-trail is on timeout 3600

Interface Configuration
 Interface Serial0/0/0
  Inbound inspection rule is myrules
    tcp alert is on audit-trail is off timeout 3600
    udp alert is on audit-trail is off timeout 30
    icmp alert is on audit-trail is off timeout 5
    http alert is off audit-trail is off timeout 3600
    ftp alert is on audit-trail is on timeout 3600
  Outgoing inspection rule is not set
  Inbound access list is not set
  Outgoing access list is not set

Established Sessions
 Session 458348C4 (172.16.12.1:54736)=>(192.168.23.3:23) tcp SIS_OPEN
```

View detailed session information by issuing the **show ip inspect detail** command on FW.

```
FW# show ip inspect sessions detail
Established Sessions
 Session 458348C4 (172.16.12.1:54736)=>(192.168.23.3:23) tcp SIS_OPEN
  Created 00:03:25, Last heard 00:03:23
  Bytes sent (initiator:responder) [37:79]
  In  SID 192.168.23.3[23:23]=>172.16.12.1[54736:54736] on ACL 100  (11
matches)
```

Close the telnet connection when you are done verifying CBAC operation.

```
OUTSIDE> exit

[Connection to 192.168.23.3 closed by foreign host]
INSIDE#
```

Note: If your Cisco IOS release does not support ICMP inspection, skip the following verification step since ICMP traffic will not be inspected.

Enable debugging of IP inspection for ICMP traffic using the **debug ip inspect** *protocol* command. In a production environment, debugging CBAC is not recommended because of the high amounts of output it can generate.

```
FW# debug ip inspect icmp
INSPECT ICMP Inspection debugging is on
```

From the INSIDE router, **ping** OUTSIDE. Note that this would not work if you try to ping the other way because it would be denied by the access list. If attempted, the denied packets from OUTSIDE to INSIDE would be logged to FW's console line as well.

```
INSIDE# ping 192.168.23.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

FW#
*Feb 18 02:23:29.591: CBAC: ICMP Echo pkt 172.16.12.1 => 192.168.23.3
*Feb 18 02:23:29.591: CBAC: ICMP Echo pkt 172.16.12.1 => 192.168.23.3
*Feb 18 02:23:29.591: CBAC: ICMP Echo pkt 172.16.12.1 => 192.168.23.3
*Feb 18 02:23:29.619: CBAC: ICMP Echo Reply pkt 192.168.23.3 => 172.16.12.1
*Feb 18 02:23:29.647: CBAC: ICMP Echo pkt 172.16.12.1 => 192.168.23.3
*Feb 18 02:23:29.675: CBAC: ICMP Echo Reply pkt 192.168.23.3 => 172.16.12.1
*Feb 18 02:23:29.703: CBAC: ICMP Echo pkt 172.16.12.1 => 192.168.23.3
*Feb 18 02:23:29.735: CBAC: ICMP Echo Reply pkt 192.168.23.3 => 172.16.12.1
*Feb 18 02:23:29.763: CBAC: ICMP Echo pkt 172.16.12.1 => 192.168.23.3
*Feb 18 02:23:29.791: CBAC: ICMP Echo Reply pkt 192.168.23.3 => 172.16.12.1
*Feb 18 02:23:29.819: CBAC: ICMP Echo pkt 172.16.12.1 => 192.168.23.3
*Feb 18 02:23:29.847: CBAC: ICMP Echo Reply pkt 192.168.23.3 => 172.16.12.1
FW# undebug all
```

Final Configurations

```
INSIDE# show run
hostname INSIDE
!
interface Serial0/0/0
 ip address 172.16.12.1 255.255.255.0
 clock rate 64000
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 172.16.12.2
!
line vty 0 4
 password cisco
 login
end

FW# show run
hostname FW
!
ip inspect name myrules tcp
ip inspect name myrules udp
ip inspect name myrules icmp timeout 5
ip inspect name myrules http alert off
ip inspect name myrules ftp audit-trail on
ip inspect udp idle-time 60
!
interface Serial0/0/0
 ip address 172.16.12.2 255.255.255.0
 ip inspect myrules in
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 ip access-group 100 in
 clock rate 64000
 no shutdown
```

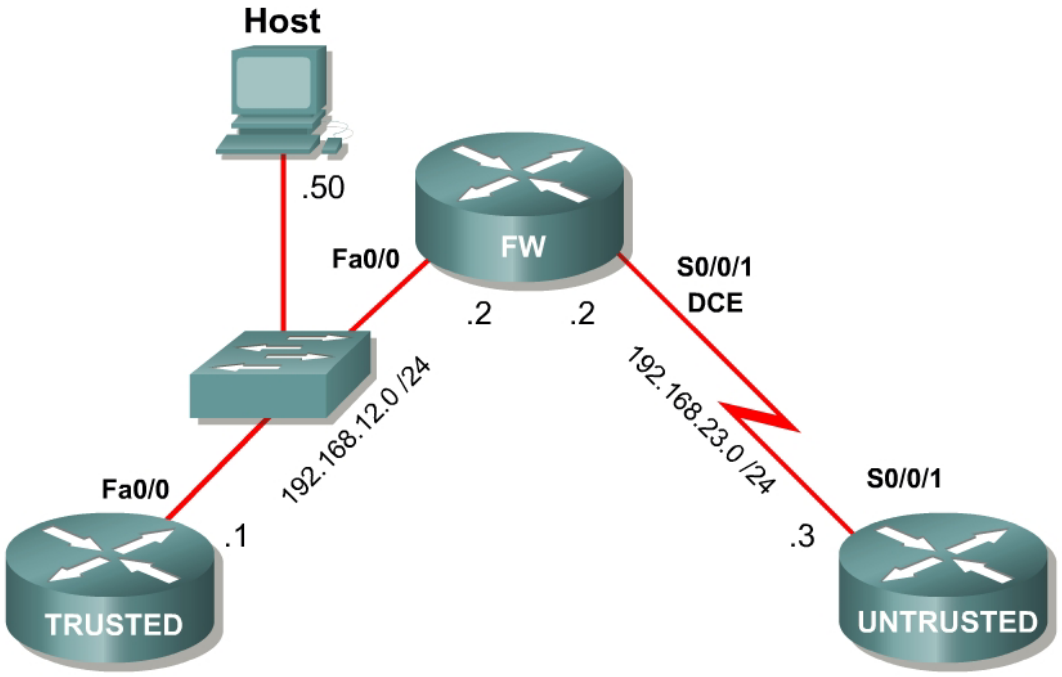
```
!
access-list 100 deny   ip any any log
end

OUTSIDE# show run
hostname OUTSIDE
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 192.168.23.2
!
line vty 0 4
 password cisco
 login
end
```

Lab 6.3 Configuring IPS with SDM

Learning Objectives

- Configure IPS using the Cisco Security Device Manager (SDM) IPS Wizard
- Modify default IPS settings
- Create an IPS signature

Topology Diagram



Scenario

In this lab, you will configure the Cisco IOS Intrusion Prevention System (IPS), which is part of the Cisco IOS Firewall feature set. IPS examines certain attack patterns and will alert and/or mitigate when those patterns occur.

In this scenario, TRUSTED represents a trusted inside router, FW represents a router serving as an intrusion prevention router, and UNTRUSTED represents an untrusted outside router. Since UNTRUSTED is outside, FW will examine packets inbound from it. IPS alone is not enough to make a router into a secure Internet firewall, but in addition to other security features, it can be a powerful defense.

Step 1: Configure the Physical Interfaces

Configure the interfaces with the addresses shown in the above topology diagram. Set the clock rate on the appropriate interface and issue the **no shutdown** command on all serial connections, as necessary. Verify that you have connectivity across the local subnet using the **ping** command.

```
TRUSTED(config)# interface fastethernet0/0
TRUSTED(config-if)# ip address 192.168.12.1 255.255.255.0
TRUSTED(config-if)# no shutdown

FW(config)# interface fastethernet0/0
FW(config-if)# ip address 192.168.12.2 255.255.255.0
FW(config-if)# no shutdown
FW(config-if)# interface serial0/0/1
FW(config-if)# ip address 192.168.23.2 255.255.255.0
FW(config-if)# clockrate 64000
FW(config-if)# no shutdown

UNTRUSTED(config)# interface serial0/0/1
UNTRUSTED(config-if)# ip address 192.168.23.3 255.255.255.0
UNTRUSTED(config-if)# no shutdown
```

Step 2: Configure Static Default Routes

On the TRUSTED and UNTRUSTED routers, configure static default routes directing traffic to unknown destinations to be forwarded to the FW router. FW will not need any routes because it has interfaces that are directly connected to both networks.

```
TRUSTED(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.2

UNTRUSTED(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2
```

Your network should have full IP connectivity at this point. If it does not have full connectivity, troubleshoot.

Normally a single-homed company might use Network Address Translation (NAT) at its corporate edge to protect its network and allow private addressing within the bounds of its network. In that case, the UNTRUSTED router, normally a provider edge router would have a static route directing traffic to the address owned by the customer out one of its interfaces. In this scenario you will not configure NAT, and you will use a default route for simplicity.

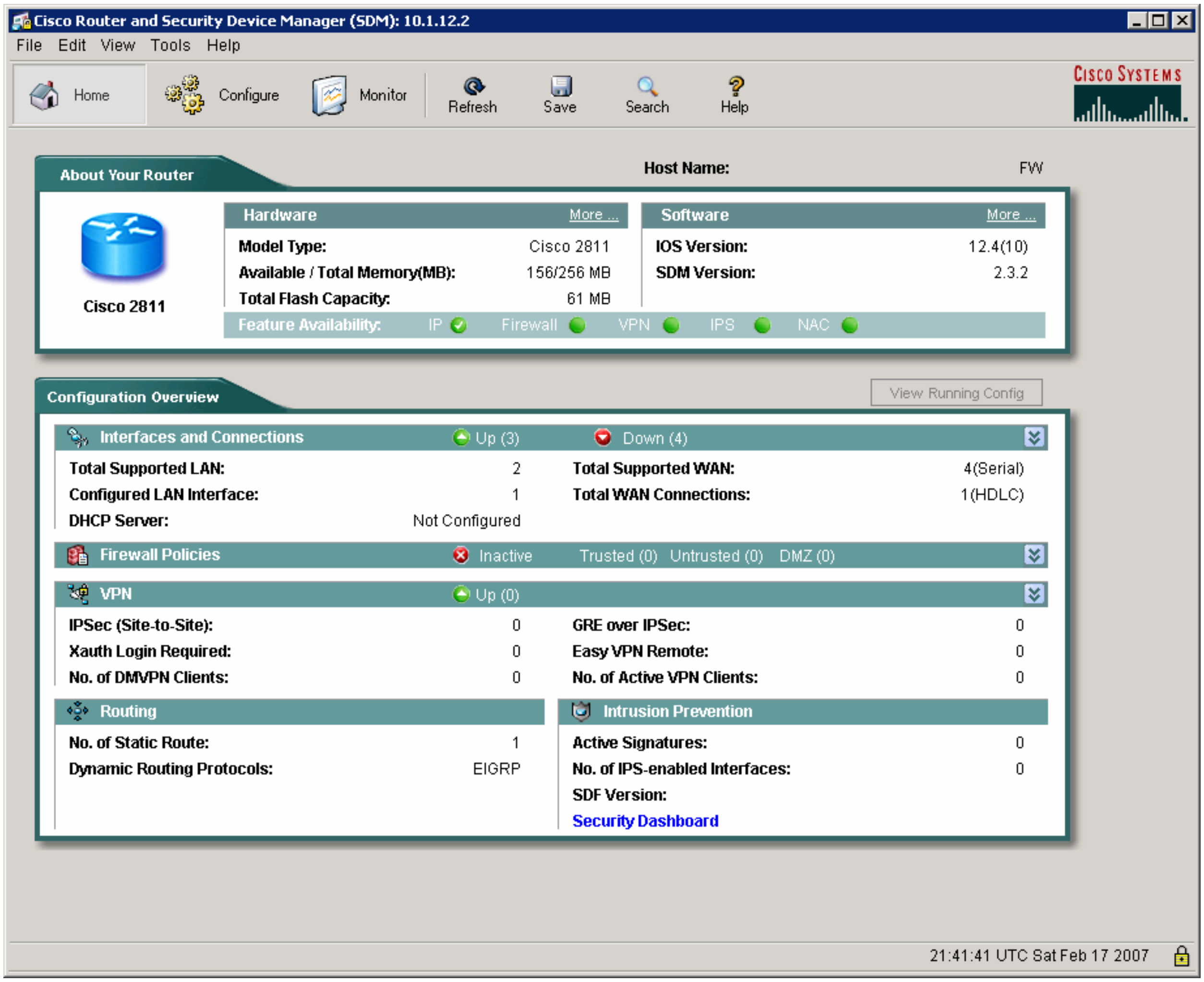
Step 3: Enable Telnet Access

On TRUSTED, enable Telnet access by setting a line password and enabling login on virtual terminal lines. Use “cisco” as a line password. These virtual terminal lines will be used later for testing the IPS.

```
TRUSTED(config)# line vty 0 4
TRUSTED(config-line)# password cisco
TRUSTED(config-line)# login
```

Step 4: Connect to FW using SDM

Set up the host with the IP addressing shown in the topology diagram. Set up FW for SDM access and connect to it using the host. If you do not know how to set the IP address on a host or connect to a router using SDM, consult the “Configuring SDM” lab. The wizard in the next step requires that you use HTTPS to connect to the router with SDM, so make sure you check this option.



About Your Router

Host Name: FW

Hardware	More ...
Model Type:	Cisco 2811
Available / Total Memory(MB):	156/256 MB
Total Flash Capacity:	61 MB

Software	More ...
IOS Version:	12.4(10)
SDM Version:	2.3.2

Feature Availability: IP ✓ Firewall ● VPN ● IPS ● NAC ●

Configuration Overview

Interfaces and Connections ● Up (3) ● Down (4)

Total Supported LAN:	2	Total Supported WAN:	4(Serial)
Configured LAN Interface:	1	Total WAN Connections:	1(HDLC)
DHCP Server:	Not Configured		

Firewall Policies ☒ Inactive Trusted (0) Untrusted (0) DMZ (0)

VPN ● Up (0)

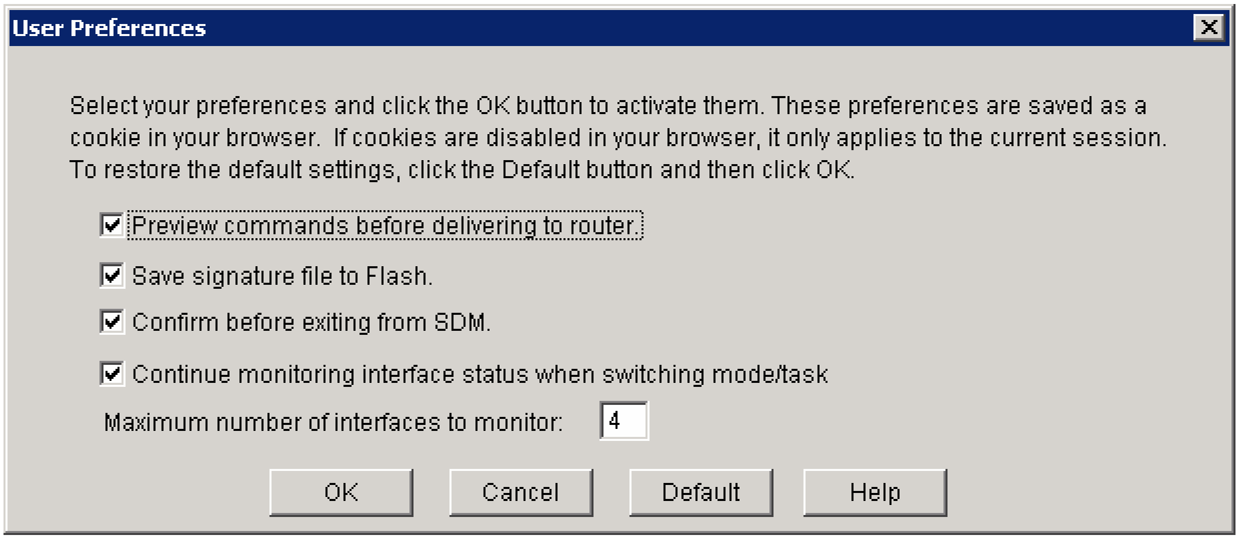
IPSec (Site-to-Site):	0	GRE over IPSec:	0
Xauth Login Required:	0	Easy VPN Remote:	0
No. of DMVPN Clients:	0	No. of Active VPN Clients:	0

Routing		Intrusion Prevention	
No. of Static Route:	1	Active Signatures:	0
Dynamic Routing Protocols:	EIGRP	No. of IPS-enabled Interfaces:	0
		SDF Version:	
		Security Dashboard	

21:41:41 UTC Sat Feb 17 2007

Figure 4-1: Cisco Security Device Manager Home Screen

On the **Edit** menu, choose **Preferences**. Verify that the **Preview commands before delivering to router** option is checked before continuing. Click **OK** when you are done configuring the preferences.


User Preferences

Select your preferences and click the OK button to activate them. These preferences are saved as a cookie in your browser. If cookies are disabled in your browser, it only applies to the current session. To restore the default settings, click the Default button and then click OK.

☑ Preview commands before delivering to router.

☑ Save signature file to Flash.

☑ Confirm before exiting from SDM.

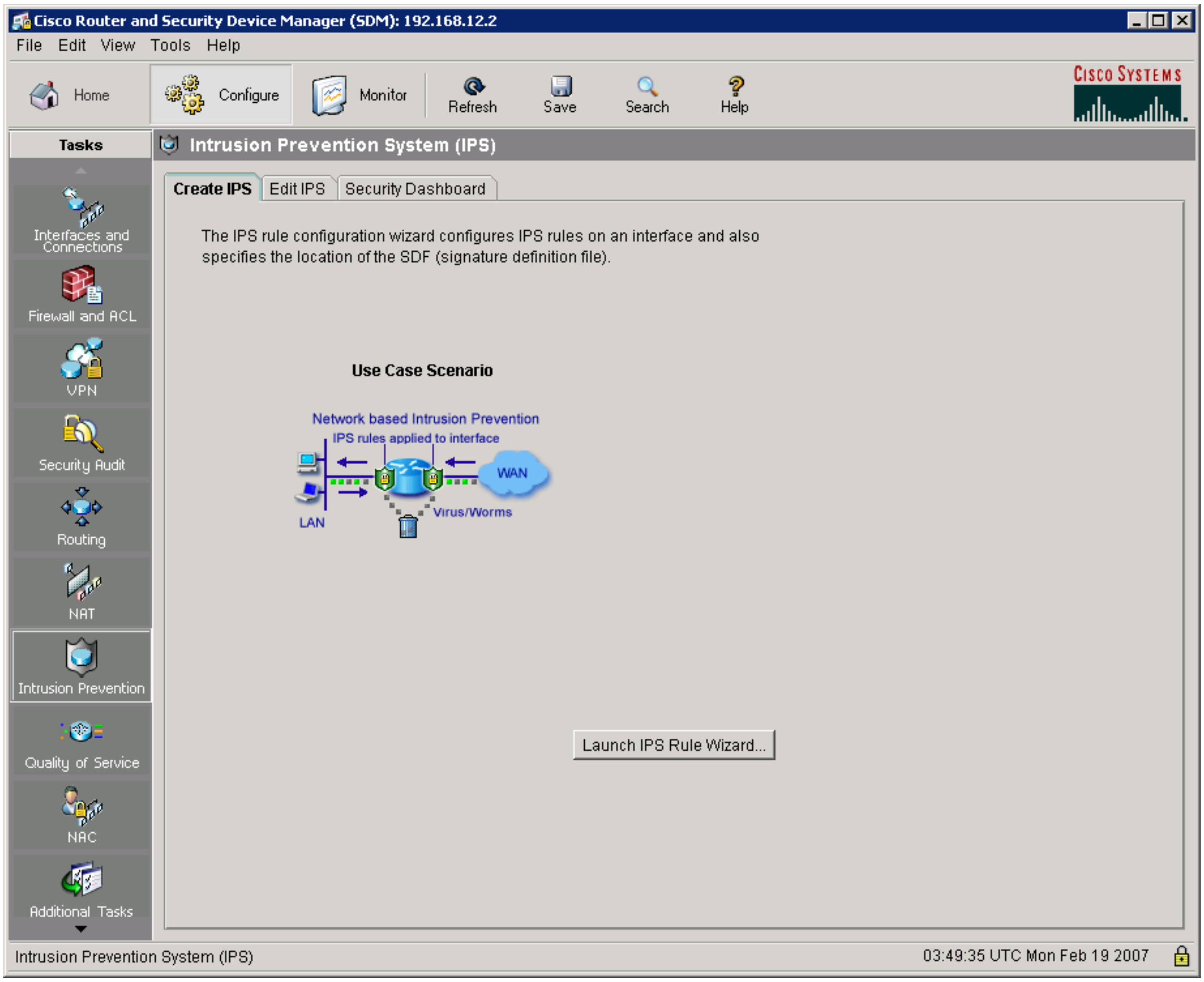
☑ Continue monitoring interface status when switching mode/task

Maximum number of interfaces to monitor: 4

Figure 4-2: SDM User Preferences

Step 5: Use the SDM IPS Rule Wizard

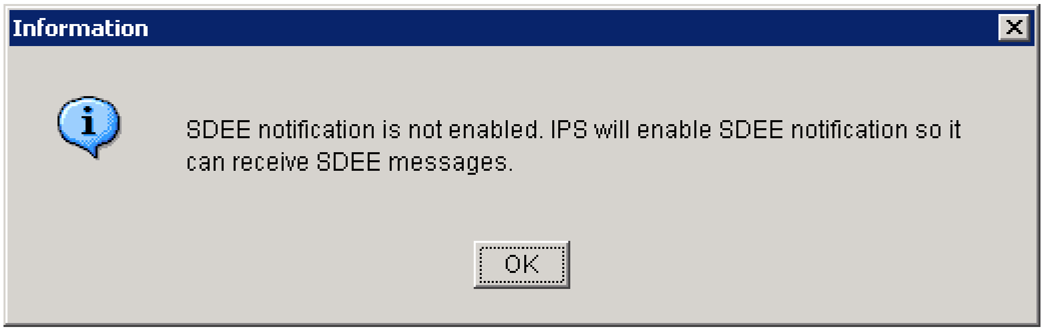
In SDM, to start the IPS Rule Wizard click to the **Configure** icon in the top menu bar, click **Intrusion Prevention** on the Tasks toolbar, and then click the **Launch IPS Rule Wizard** button.


Cisco Router and Security Device Manager (SDM): 192.168.12.2
File Edit View Tools Help
Home
Configure
Monitor
Refresh
Save
Search
Help
Cisco Systems
Tasks
Interfaces and Connections
Firewall and ACL
VPN
Security Audit
Routing
NAT
Intrusion Prevention
Quality of Service
NAC
Additional Tasks
Intrusion Prevention System (IPS)
Create IPS
Edit IPS
Security Dashboard
Use Case Scenario
Network based Intrusion Prevention
IPS rules applied to interface
WAN
LAN
Virus/Worms
Launch IPS Rule Wizard...
Intrusion Prevention System (IPS)
03:49:35 UTC Mon Feb 19 2007

The IPS rule configuration wizard configures IPS rules on an interface and also specifies the location of the SDF (signature definition file).

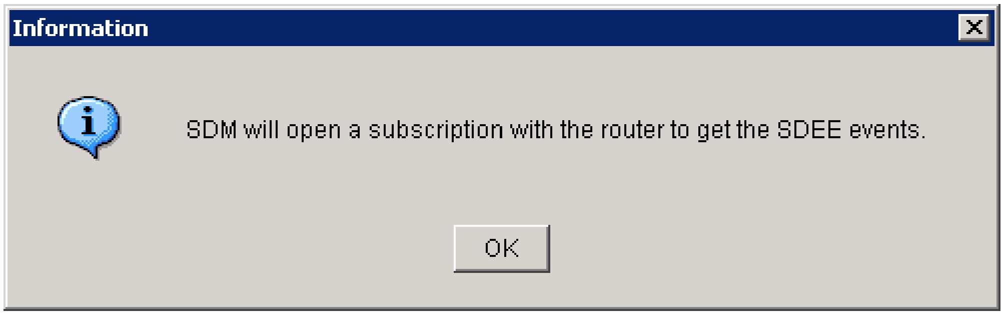
Figure 5-1: Launching the SDM IPS Wizard

Some dialog boxes regarding SDEE notification will appear. SDEE is a technology used to report security events and responses which is enabled when IPS is enabled on a router. Click **OK** on each dialog box.



SDEE notification is not enabled. IPS will enable SDEE notification so it can receive SDEE messages.

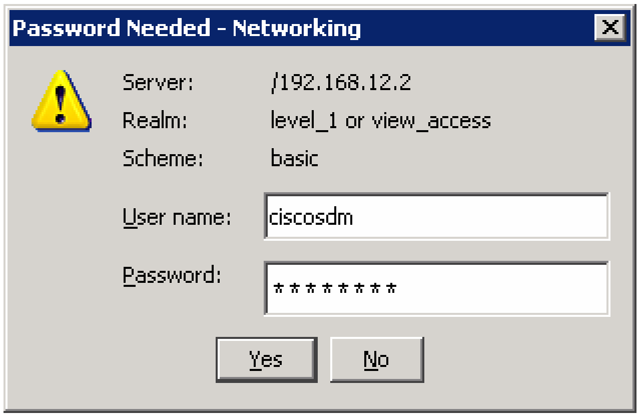
Figure 5-2: Notification of Enabling SDEE



SDM will open a subscription with the router to get the SDEE events.

Figure 5-3: Permission of Enabling SDEE

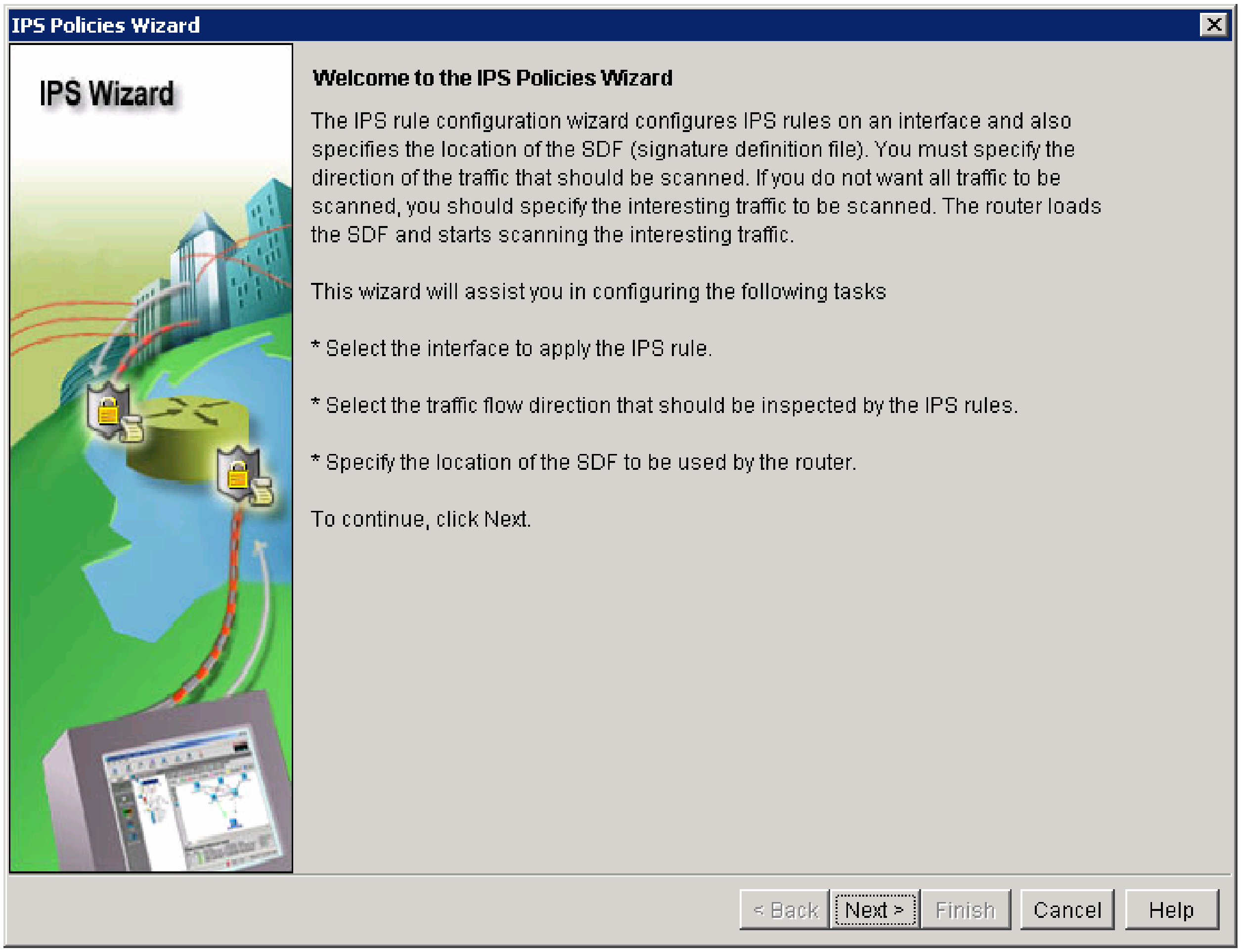
When prompted for a username and password, use the username and password you used to log in to SDM.



Server: /192.168.12.2
Realm: level_1 or view_access
Scheme: basic
User name: ciscosdm
Password: ********

Figure 5-4: SDM Login to FW Router

Read the welcome page of the wizard, and then click **Next**.



Welcome to the IPS Policies Wizard

The IPS rule configuration wizard configures IPS rules on an interface and also specifies the location of the SDF (signature definition file). You must specify the direction of the traffic that should be scanned. If you do not want all traffic to be scanned, you should specify the interesting traffic to be scanned. The router loads the SDF and starts scanning the interesting traffic.

This wizard will assist you in configuring the following tasks

* Select the interface to apply the IPS rule.

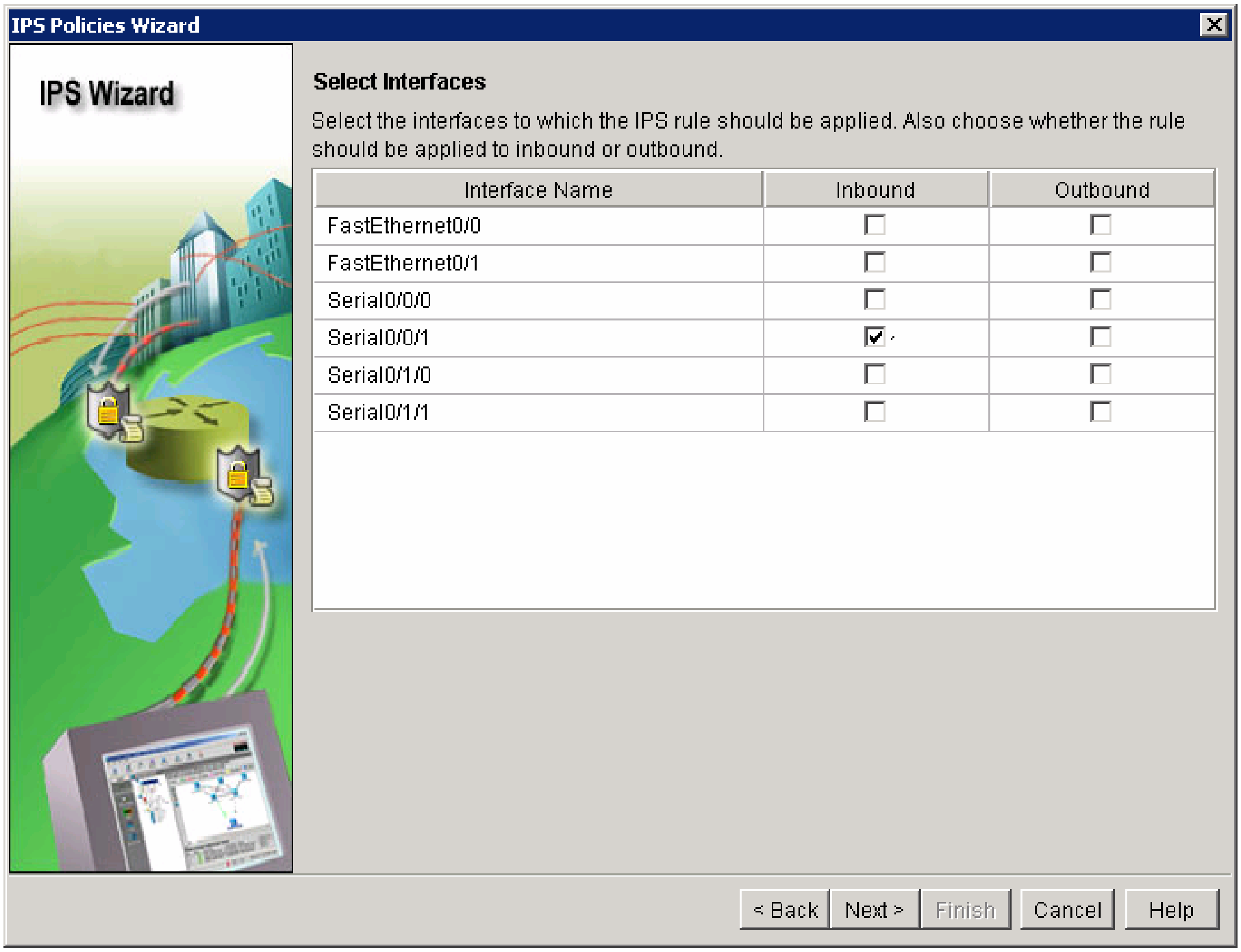
* Select the traffic flow direction that should be inspected by the IPS rules.

* Specify the location of the SDF to be used by the router.

To continue, click Next.

Figure 5-5: SDM IPS Wizard

For inbound inspection, check the checkbox for the interface facing UNTRUSTED, and then click **Next**. Do not select any interfaces for outbound inspection.



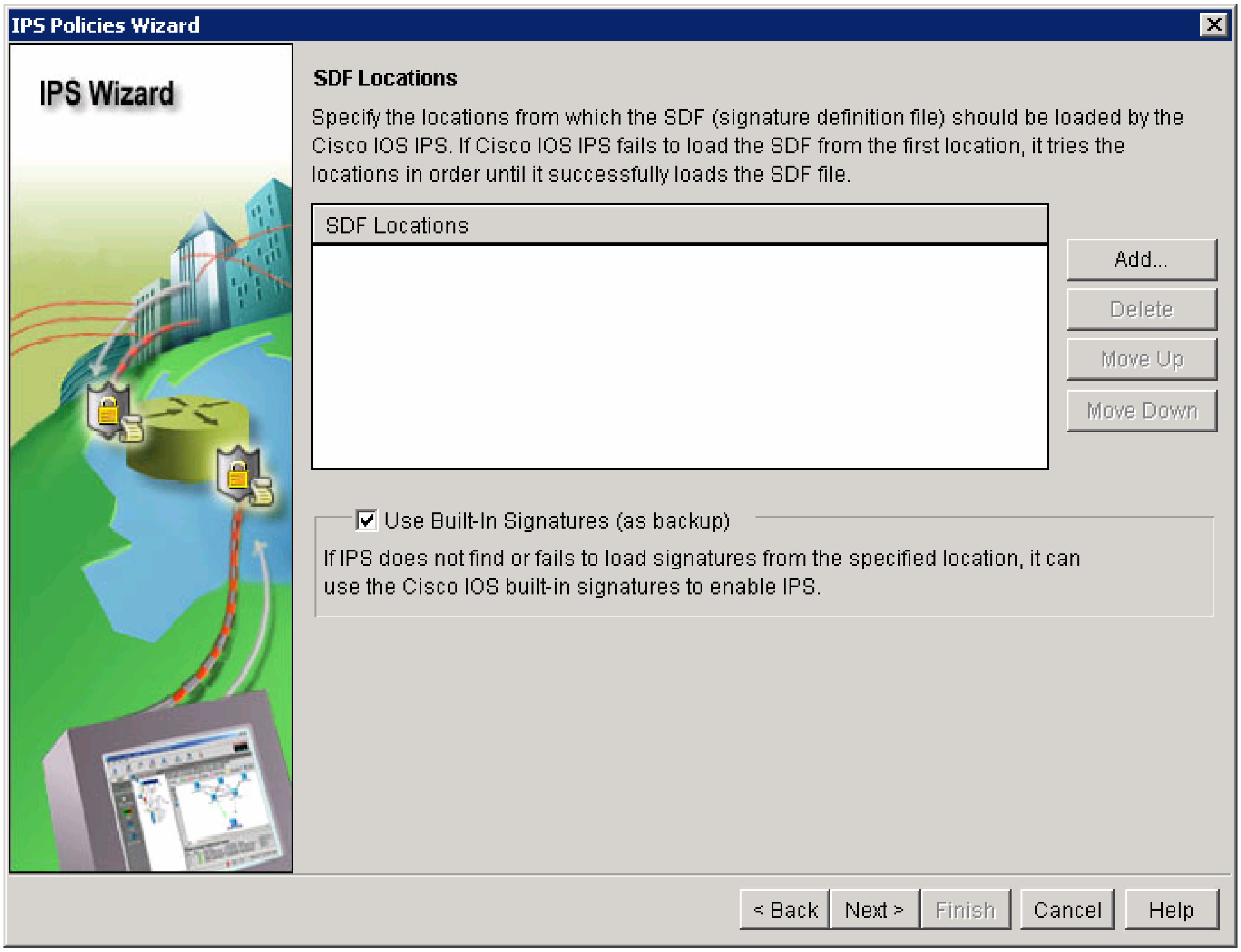
Select Interfaces

Select the interfaces to which the IPS rule should be applied. Also choose whether the rule should be applied to inbound or outbound.

Interface Name	Inbound	Outbound
FastEthernet0/0	□	□
FastEthernet0/1	□	□
Serial0/0/0	□	□
Serial0/0/1	☑	□
Serial0/1/0	□	□
Serial0/1/1	□	□

Figure 5-6: IPS Interface Selection

Signature definition files (SDF) are files that contain intrusion signature definitions. In a production environment, you would use the **Add** button to specify SDF locations. However, do not specify any SDF locations; instead, you will load basic signatures that are built into the Cisco IOS. Click **Next** on this page of the wizard.



IPS Policies Wizard

SDF Locations

Specify the locations from which the SDF (signature definition file) should be loaded by the Cisco IOS IPS. If Cisco IOS IPS fails to load the SDF from the first location, it tries the locations in order until it successfully loads the SDF file.

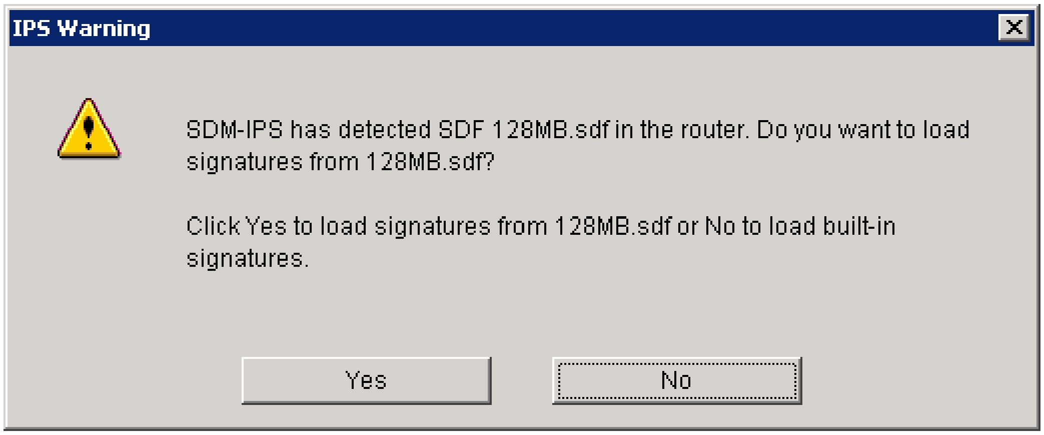
SDF Locations

☑ Use Built-In Signatures (as backup)

If IPS does not find or fails to load signatures from the specified location, it can use the Cisco IOS built-in signatures to enable IPS.

Figure 5-7: SDF Locations

A warning dialog that is similar to the one shown in Figure 5-8 may appear if your router contains an SDF file in its flash memory. If you do receive the warning, click **No** to use the built-in signatures.



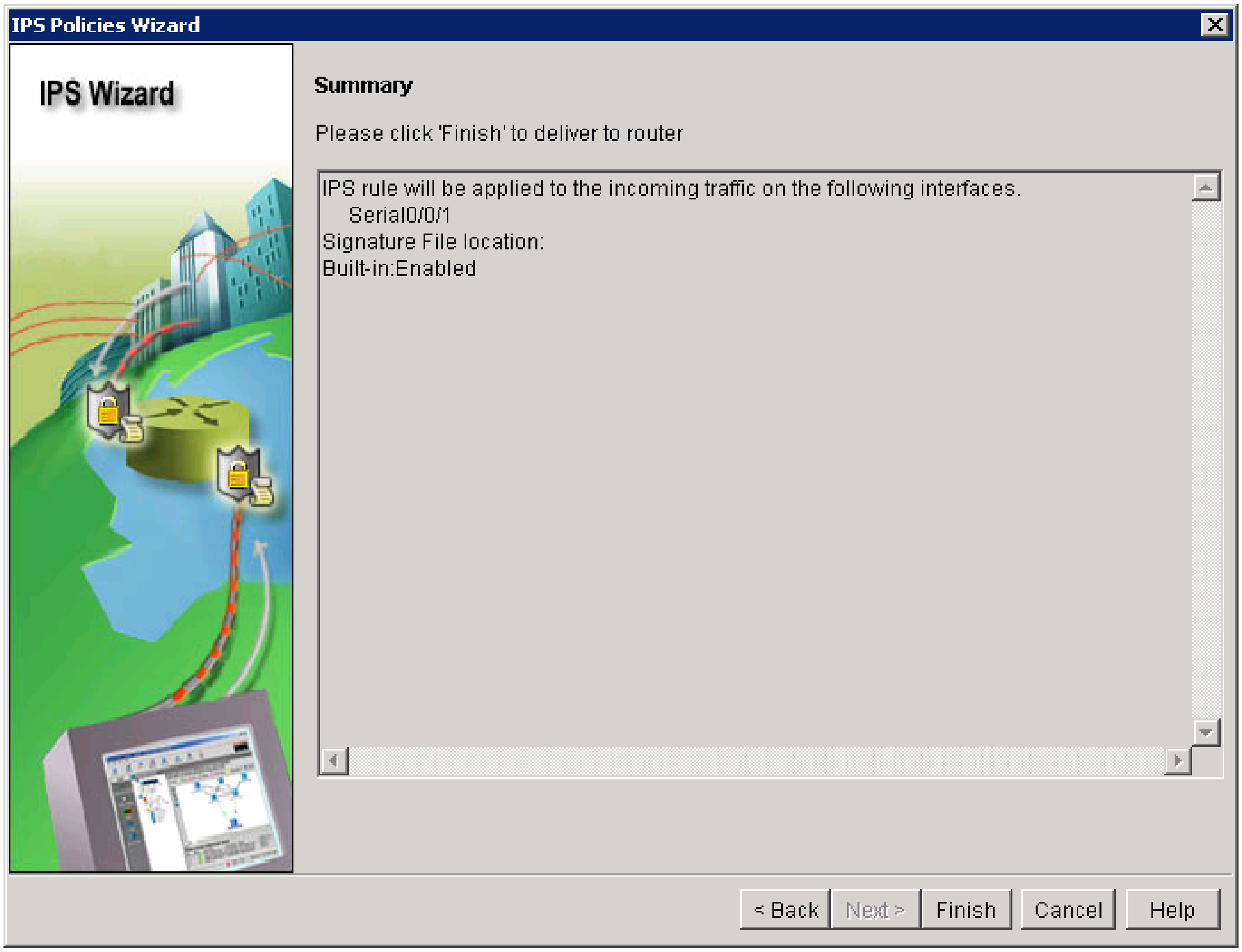
IPS Warning

SDM-IPS has detected SDF 128MB.sdf in the router. Do you want to load signatures from 128MB.sdf?

Click Yes to load signatures from 128MB.sdf or No to load built-in signatures.

Figure 5-8: SDF File Detection

After verifying the changes SDM will make to the router, click **Finish** to begin applying the changes.



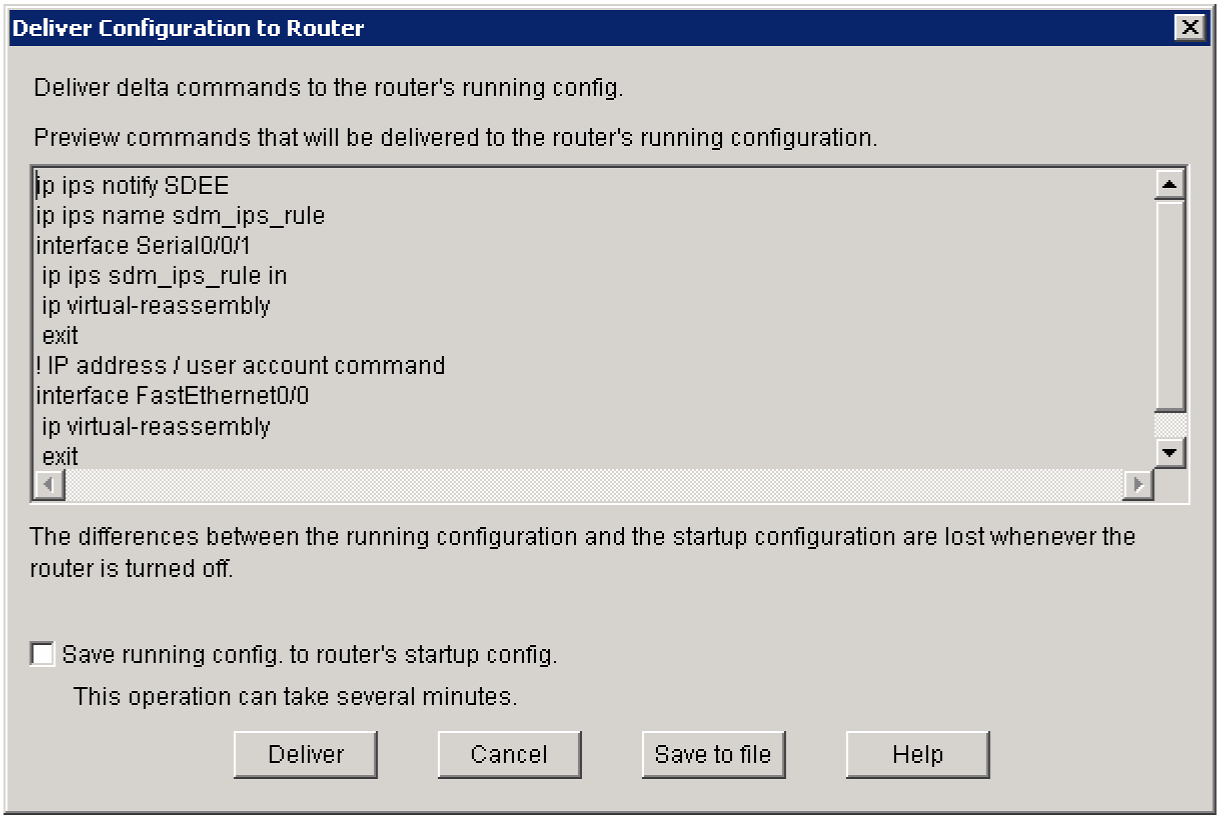
Summary

Please click 'Finish' to deliver to router

IPS rule will be applied to the incoming traffic on the following interfaces.
    Serial0/0/1
Signature File location:
Built-in:Enabled

Figure 5-9: IPS Summary

Verify the commands that SDM will use on the router, and then click **Deliver** to add the configuration.



Deliver delta commands to the router's running config.

Preview commands that will be delivered to the router's running configuration.

```
ip ips notify SDEE
ip ips name sdm_ips_rule
interface Serial0/0/1
 ip ips sdm_ips_rule in
 ip virtual-reassembly
 exit
! IP address / user account command
interface FastEthernet0/0
 ip virtual-reassembly
 exit
```

The differences between the running configuration and the startup configuration are lost whenever the router is turned off.

□ Save running config. to router's startup config.

This operation can take several minutes.

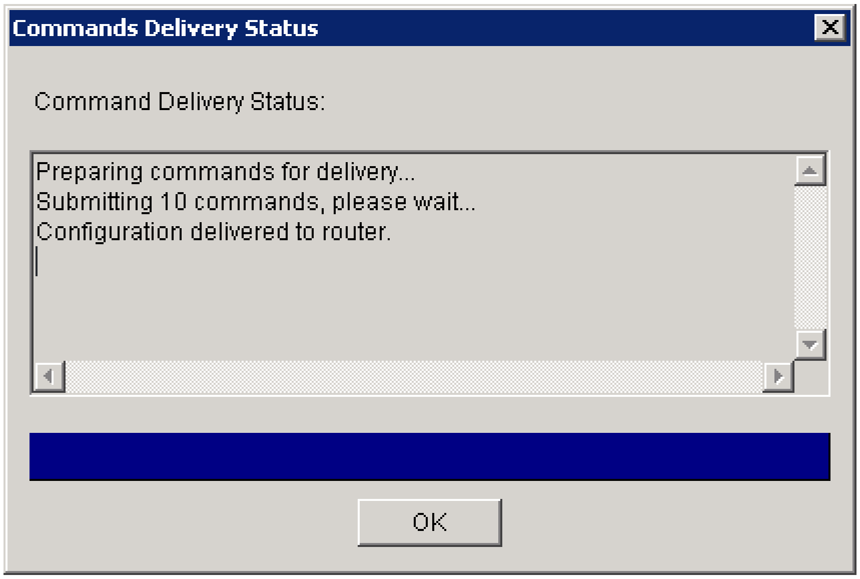
Figure 5-10: IPS Configuration Summary

When the configuration is added, you may see many IPS engine messages logged on FW. Do not be alarmed; these are normal messages.

```
FW#
*Feb 19 04:19:52.375: %IPS-6-BUILTIN_SIGS: Configured to load builtin
signatures
*Feb 19 04:19:52.511: %IPS-6-SDF_LOAD_SUCCESS: SDF loaded successfully from
builtin
*Feb 19 04:19:52.519: %IPS-6-ENGINE_BUILDING: OTHER - 3 signatures - 1 of 15
engines
*Feb 19 04:19:52.519: %IPS-6-ENGINE_READY: OTHER - 0 ms - packets for this
engine will be scanned
*Feb 19 04:19:52.519: %IPS-6-ENGINE_BUILDING: MULTI-STRING - 0 signatures - 2
of 15 engines
*Feb 19 04:19:52.519: %IPS-6-ENGINE_BUILD_SKIPPED: MULTI-STRING - there are no
new signature definitions for this engine
*Feb 19 04:19:52.519: %IPS-6-ENGINE_BUILDING: STRING.ICMP - 0 signatures - 3
of 15 engines
*Feb 19 04:19:52.519: %IPS-6-ENGINE_BUILD_SKIPPED: STRING.ICMP - there are no
new signature definitions for this engine
*Feb 19 04:19:52.519: %IPS-6-ENGINE_BUILDING: STRING.UDP - 1 signatures - 4 of
15 engines
*Feb 19 04:19:52.531: %IPS-6-ENGINE_READY: STRING.UDP - 12 ms - packets for
this engine will be scanned
*Feb 19 04:19:53.275: %IPS-6-ENGINE_READY: SERVICE.HTTP - 460 ms - packets for
this engine will be scanned
*Feb 19 04:19:53.275: %IPS-6-ENGINE_BUILDING: ATOMIC.TCP - 6 signatures - 11
of 15 engines
```

```
*Feb 19 04:19:53.279: %IPS-6-ENGINE_READY: ATOMIC.TCP - 4 ms - packets for
this engine will be scanned
*Feb 19 04:19:53.279: %IPS-6-ENGINE_BUILDING: ATOMIC.UDP - 7 signatures - 12
of 15 engines
*Feb 19 04:19:53.283: %IPS-6-ENGINE_READY: ATOMIC.UDP - 4 ms - packets for
this engine will be scanned
*Feb 19 04:19:53.283: %IPS-6-ENGINE_BUILDING: ATOMIC.ICMP - 14 signatures - 13
of 15 engines
*Feb 19 04:19:53.283: %IPS-7-UNSUPPORTED_PARAM: ATOMIC.ICMP 2000:0 IcmpType=0
- This parameter is not supported
*Feb 19 04:19:53.287: %IPS-6-ENGINE_READY: ATOMIC.ICMP - 4 ms - packets for
this engine will be scanned
*Feb 19 04:19:53.287: %IPS-6-ENGINE_BUILDING: ATOMIC.IPOPTIONS - 7 signatures
- 14 of 15 engines
*Feb 19 04:19:53.287: %IPS-6-ENGINE_READY: ATOMIC.IPOPTIONS - 0 ms - packets
for this engine will be scanned
```

Once the commands are delivered, click **OK** to close the dialog box.

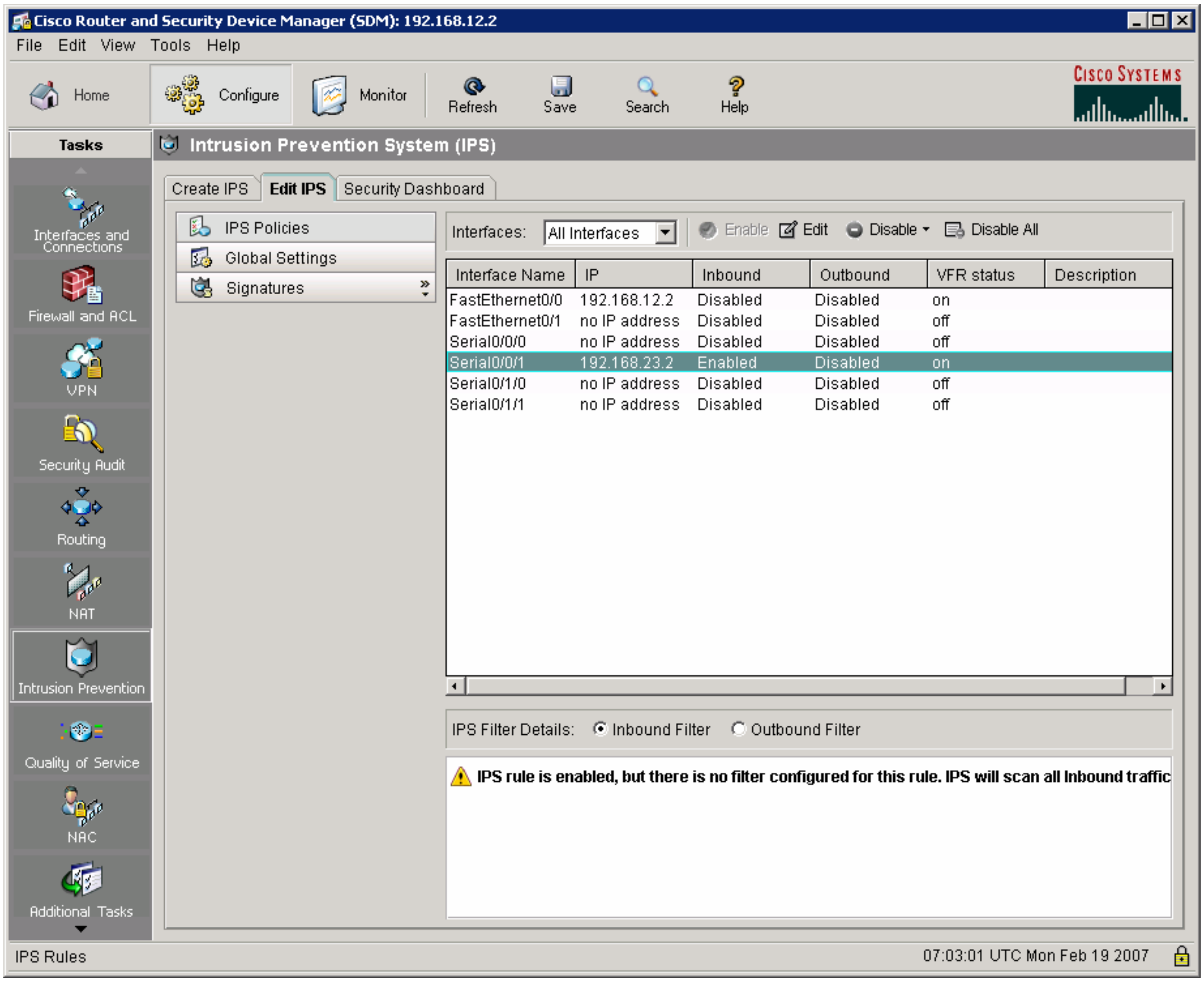


Command Delivery Status:

Preparing commands for delivery...
Submitting 10 commands, please wait...
Configuration delivered to router.

Figure 5-11: IPS Command Delivery Progress Indicator

After clicking OK on the Commands Delivery Status dialog box, SDM displays the **Edit IPS** tab of SDM as a prompt.



Interfaces: All Interfaces

Interface Name	IP	Inbound	Outbound	VFR status	Description
FastEthernet0/0	192.168.12.2	Disabled	Disabled	on	
FastEthernet0/1	no IP address	Disabled	Disabled	off	
Serial0/0/0	no IP address	Disabled	Disabled	off	
Serial0/0/1	192.168.23.2	Enabled	Disabled	on	
Serial0/1/0	no IP address	Disabled	Disabled	off	
Serial0/1/1	no IP address	Disabled	Disabled	off	

IPS Filter Details: ● Inbound Filter ○ Outbound Filter

IPS rule is enabled, but there is no filter configured for this rule. IPS will scan all Inbound traffic

Figure 5-12: SDM Edit IPS Tab

FW now examines each packet passing through it with an ingress interface of Serial 0/0/1. Note that no other packets will be examined.

Step 6: Verify and Modify IPS Behavior

On UNTRUSTED, ping TRUSTED with a high repeat count.

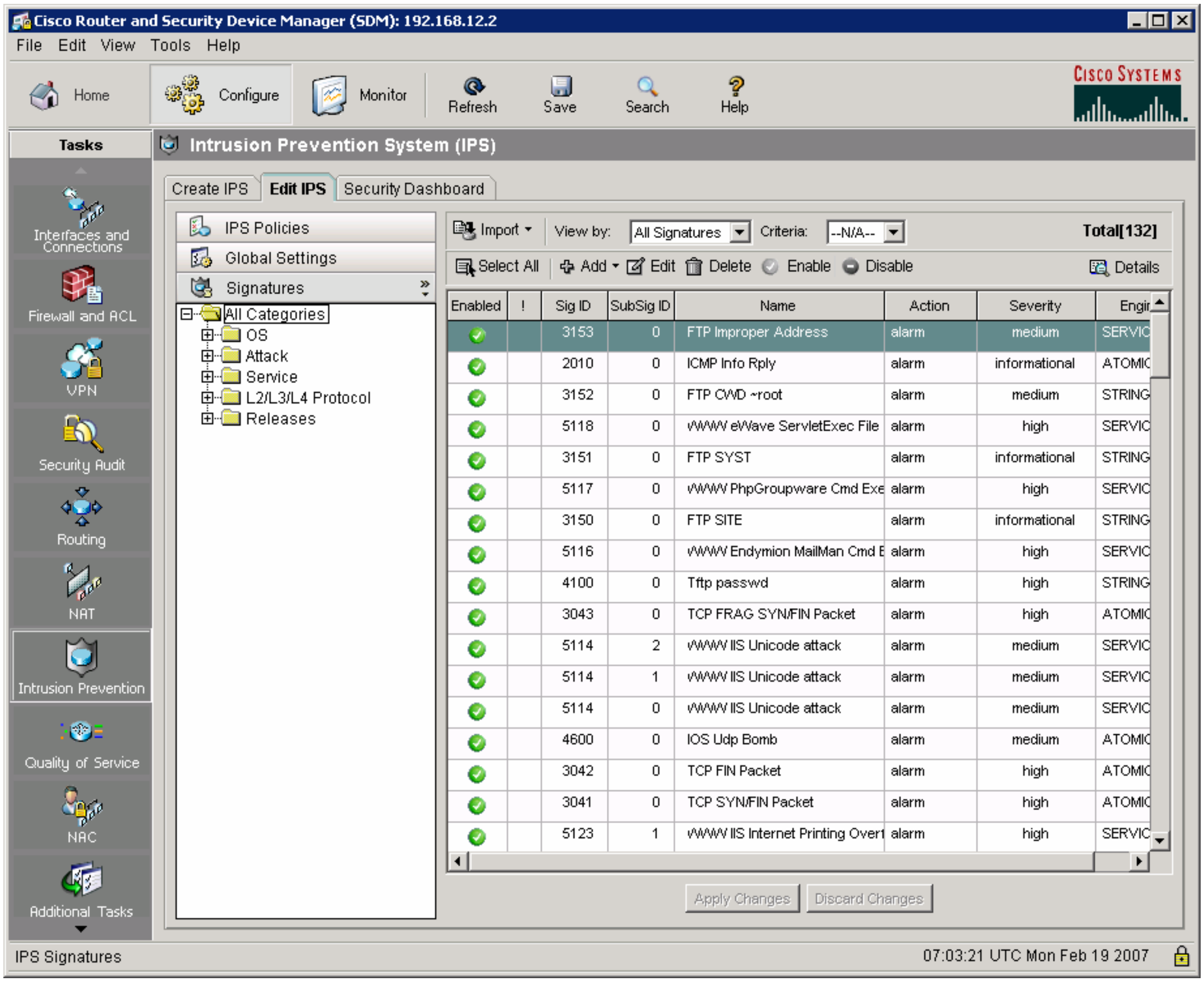
```
UNTRUSTED# ping 192.168.12.1 repeat 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 28/29/32 ms
```

View the messages logged to FW's console line. Initially, IPS logs each individual signature catch (which is triggered by each Internet Control Message Protocol (ICMP) packet sent by the **ping** command), but eventually stops logging each one individually. Finally, it shows a summary log message.

```
FW#
*Feb 19 06:55:05.603: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.635: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.663: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.695: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.723: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.751: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.783: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.811: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.843: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.871: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.899: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.931: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.959: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:05.991: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.019: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.047: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.079: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.107: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.139: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.167: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.907: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.935: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.967: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:06.995: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.023: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.055: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.083: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.115: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.143: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.171: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.883: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
```

```
*Feb 19 06:55:07.915: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.943: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:07.971: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:08.003: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:08.031: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:08.063: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:08.091: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:08.119: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 06:55:08.151: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
FW#
*Feb 19 06:55:35.603: %IPS-4-SIG_SUMMARY: Sig:2004 Subsig:0 Global Summary:
100 alarms in this interval
```

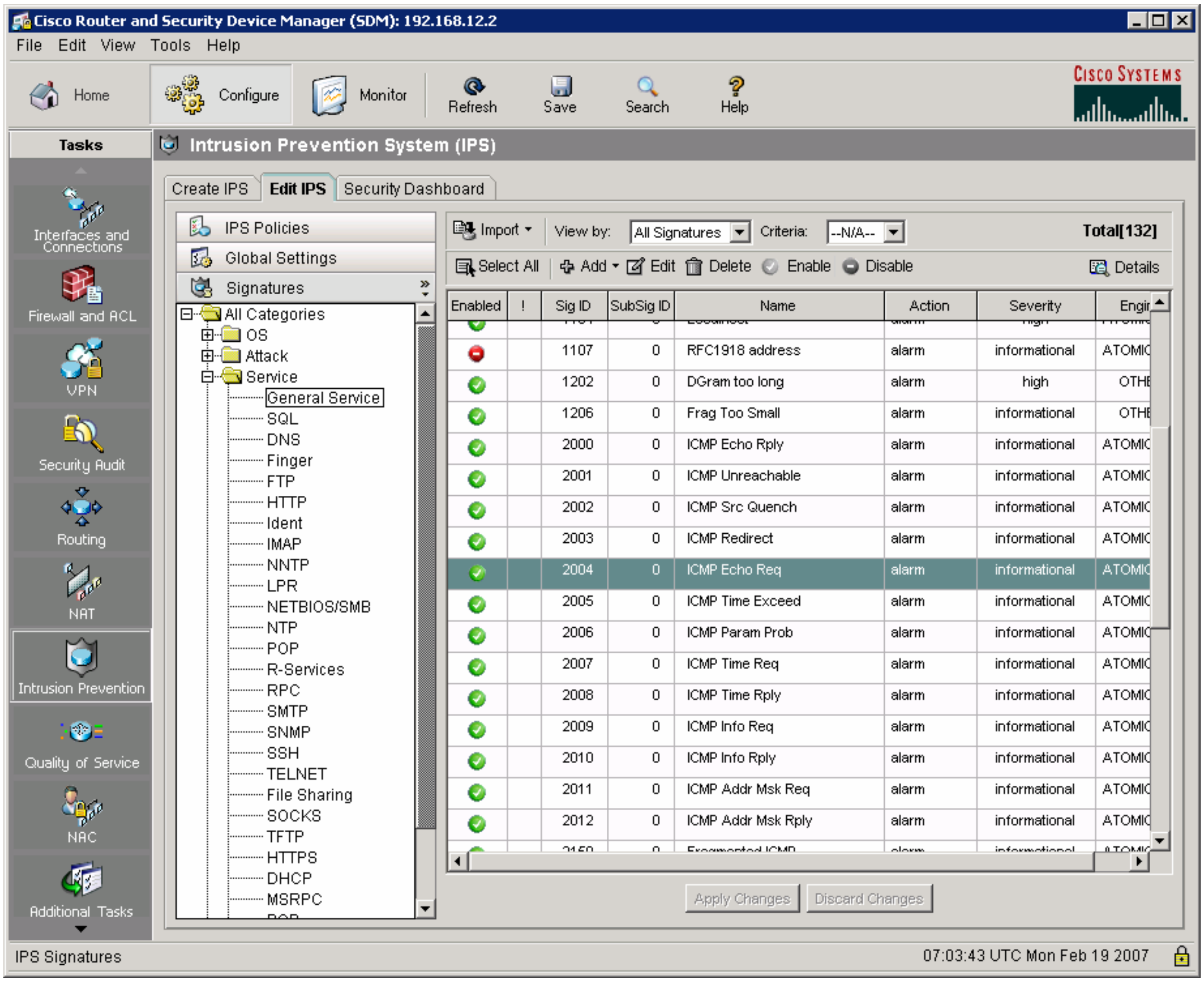
Signature number 2004 detected the previous potential ICMP attack indicated in the output appearing before this paragraph. For this lab, you will disable the signature numbered 2004, which was being set off by the **ping** command that was issued previously. Begin by clicking **Signatures** in the first pane of the tab.



Enabled	!	Sig ID	SubSig ID	Name	Action	Severity	Engir
✓		3153	0	FTP Improper Address	alarm	medium	SERVIC
✓		2010	0	ICMP Info Rply	alarm	informational	ATOMIC
✓		3152	0	FTP CWD ~root	alarm	medium	STRING
✓		5118	0	WWW eWave ServletExec File	alarm	high	SERVIC
✓		3151	0	FTP SYST	alarm	informational	STRING
✓		5117	0	WWW PhpGroupware Cmd Exe	alarm	high	SERVIC
✓		3150	0	FTP SITE	alarm	informational	STRING
✓		5116	0	WWW Endymion MailMan Cmd E	alarm	high	SERVIC
✓		4100	0	Tftp passwd	alarm	high	STRING
✓		3043	0	TCP FRAG SYN/FIN Packet	alarm	high	ATOMIC
✓		5114	2	WWW IIS Unicode attack	alarm	medium	SERVIC
✓		5114	1	WWW IIS Unicode attack	alarm	medium	SERVIC
✓		5114	0	WWW IIS Unicode attack	alarm	medium	SERVIC
✓		4600	0	IOS Udp Bomb	alarm	medium	ATOMIC
✓		3042	0	TCP FIN Packet	alarm	high	ATOMIC
✓		3041	0	TCP SYN/FIN Packet	alarm	high	ATOMIC
✓		5123	1	WWW IIS Internet Printing Overf	alarm	high	SERVIC

Figure 6-1: Edit IPS Signatures Tab

Choose **Service > General Service** on the signature tree. Find signature 2004 in the list, and choose it. Then, disable the signature by clicking the **Disable** icon in the menu bar for the list.



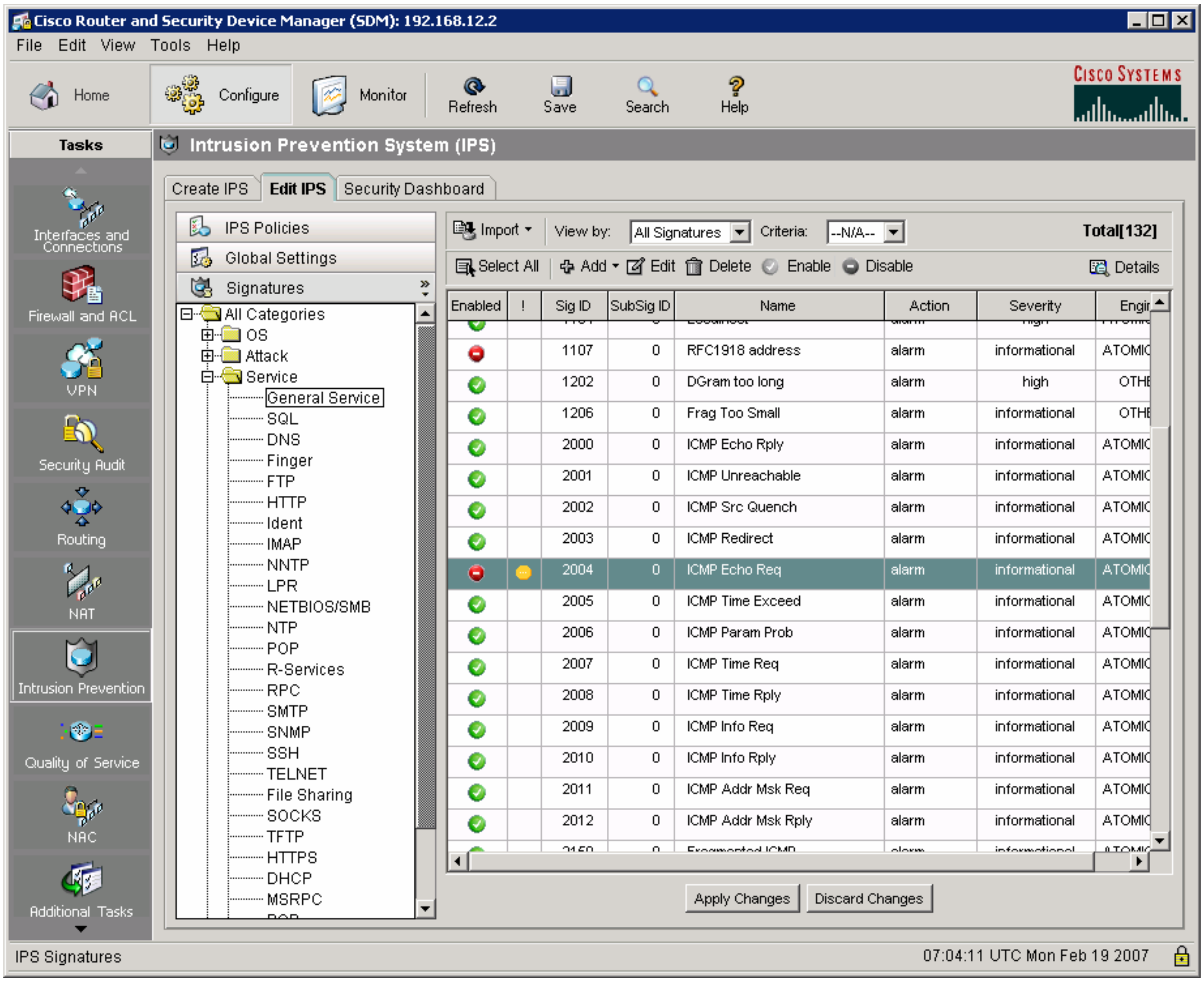
View by: All Signatures Criteria: --N/A-- Total[132]

- All Categories
  - OS
  - Attack
  - Service
    - General Service
    - SQL
    - DNS
    - Finger
    - FTP
    - HTTP
    - Ident
    - IMAP
    - NNTP
    - LPR
    - NETBIOS/SMB
    - NTP
    - POP
    - R-Services
    - RPC
    - SMTP
    - SNMP
    - SSH
    - TELNET
    - File Sharing
    - SOCKS
    - TFTP
    - HTTPS
    - DHCP
    - MSRPC

Enabled	!	Sig ID	SubSig ID	Name	Action	Severity	Engin
⊖		1107	0	RFC1918 address	alarm	informational	ATOMIC
✓		1202	0	DGram too long	alarm	high	OTHE
✓		1206	0	Frag Too Small	alarm	informational	OTHE
✓		2000	0	ICMP Echo Rply	alarm	informational	ATOMIC
✓		2001	0	ICMP Unreachable	alarm	informational	ATOMIC
✓		2002	0	ICMP Src Quench	alarm	informational	ATOMIC
✓		2003	0	ICMP Redirect	alarm	informational	ATOMIC
✓		2004	0	ICMP Echo Req	alarm	informational	ATOMIC
✓		2005	0	ICMP Time Exceed	alarm	informational	ATOMIC
✓		2006	0	ICMP Param Prob	alarm	informational	ATOMIC
✓		2007	0	ICMP Time Req	alarm	informational	ATOMIC
✓		2008	0	ICMP Time Rply	alarm	informational	ATOMIC
✓		2009	0	ICMP Info Req	alarm	informational	ATOMIC
✓		2010	0	ICMP Info Rply	alarm	informational	ATOMIC
✓		2011	0	ICMP Addr Msk Req	alarm	informational	ATOMIC
✓		2012	0	ICMP Addr Msk Rply	alarm	informational	ATOMIC

Figure 6-2: IPS ICMP Echo Request Signature, Currently Enable

The **Enabled** column should change its icon to reflect that it is now disabled. Click the **Apply Changes** button to deliver the changes to the router.

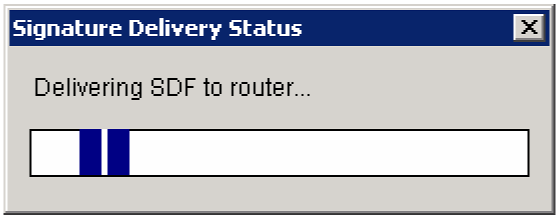


View by: All Signatures Criteria: --N/A-- Total[132]

Enabled	!	Sig ID	SubSig ID	Name	Action	Severity	Engi
⊖		1107	0	RFC1918 address	alarm	informational	ATOMIC
✓		1202	0	DGram too long	alarm	high	OTHE
✓		1206	0	Frag Too Small	alarm	informational	OTHE
✓		2000	0	ICMP Echo Rply	alarm	informational	ATOMIC
✓		2001	0	ICMP Unreachable	alarm	informational	ATOMIC
✓		2002	0	ICMP Src Quench	alarm	informational	ATOMIC
✓		2003	0	ICMP Redirect	alarm	informational	ATOMIC
⊖	●	2004	0	ICMP Echo Req	alarm	informational	ATOMIC
✓		2005	0	ICMP Time Exceed	alarm	informational	ATOMIC
✓		2006	0	ICMP Param Prob	alarm	informational	ATOMIC
✓		2007	0	ICMP Time Req	alarm	informational	ATOMIC
✓		2008	0	ICMP Time Rply	alarm	informational	ATOMIC
✓		2009	0	ICMP Info Req	alarm	informational	ATOMIC
✓		2010	0	ICMP Info Rply	alarm	informational	ATOMIC
✓		2011	0	ICMP Addr Msk Req	alarm	informational	ATOMIC
✓		2012	0	ICMP Addr Msk Rply	alarm	informational	ATOMIC

Figure 6-3: IPS ICMP Echo Request Signature, Now Disabled

A new SDF will be generated by SDM reflecting these changes, and it will be delivered to the router from SDM.

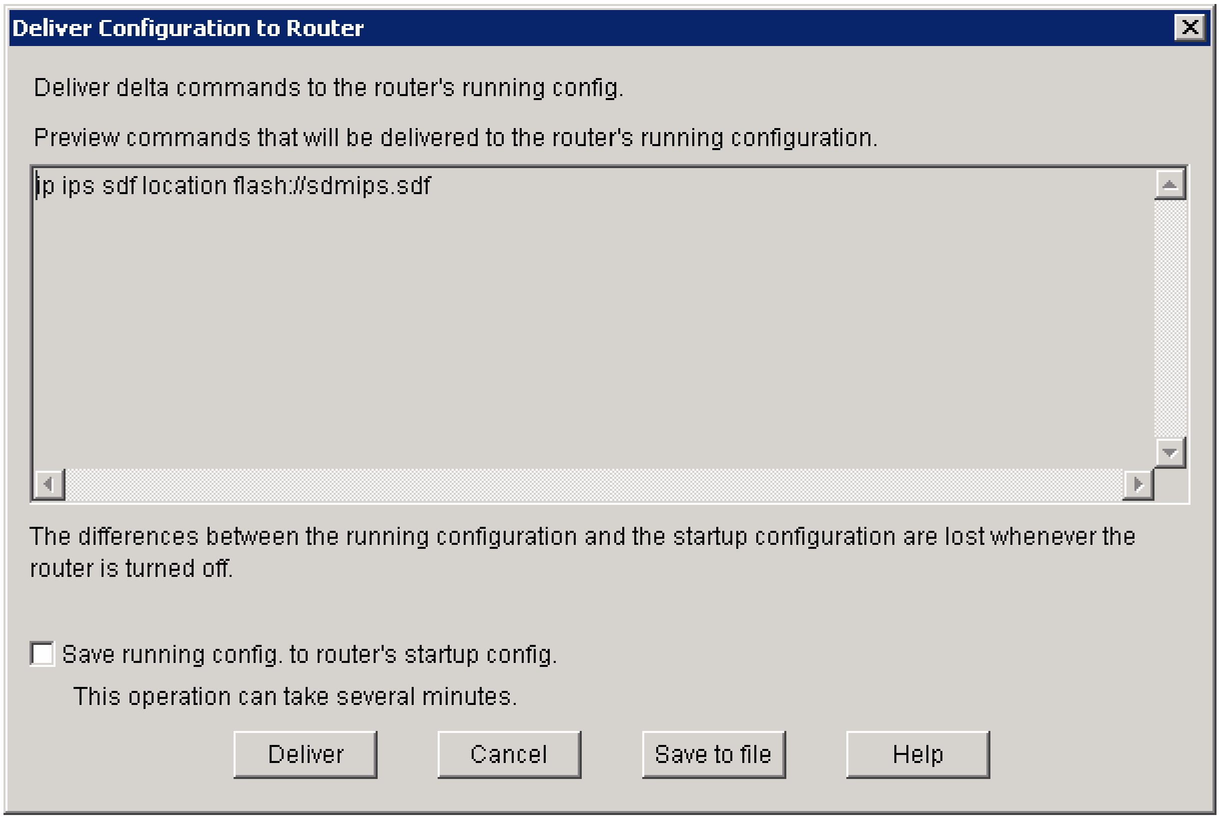


Delivering SDF to router...

Figure 6-4: IPS Signature Delivery Status

Note: At the end of this lab, you will probably want to delete this SDF by using the privileged EXEC command **delete flash:sdmips.sdf**. Be careful not to delete any other files in the flash file system.

SDM will also add a configuration command to reflect the new SDF that it is adding and tell the IPS engine to load signatures from this file. Click **Deliver** after reviewing the configuration command.



Deliver Configuration to Router

Deliver delta commands to the router's running config.

Preview commands that will be delivered to the router's running configuration.

```
ip ips sdf location flash://sdmips.sdf
```

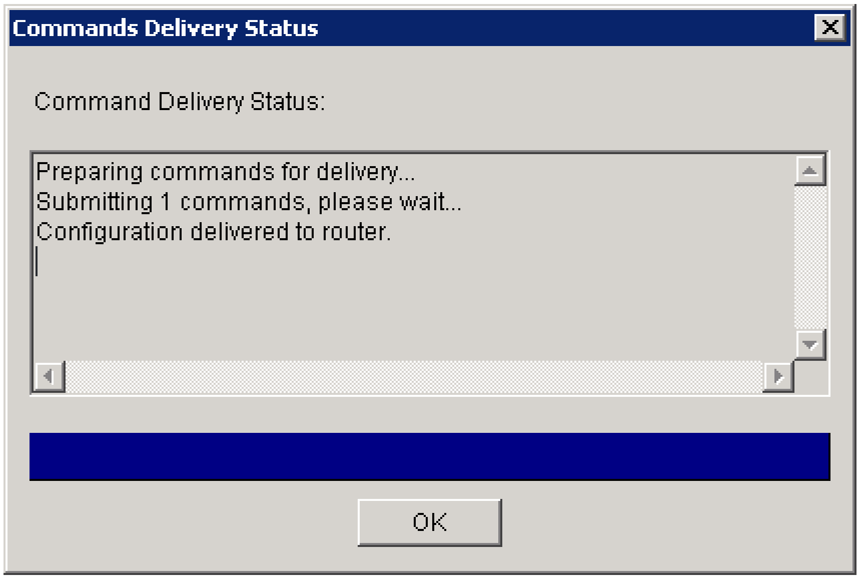
The differences between the running configuration and the startup configuration are lost whenever the router is turned off.

□ Save running config. to router's startup config.

This operation can take several minutes.

Figure 6-5: IPS Configuration Command Delivery Notification

Click **OK** after the command is delivered. You may see some IPS engine log messages on FW.



Preparing commands for delivery...
Submitting 1 commands, please wait...
Configuration delivered to router.

Figure 6-6: Command Delivery Progress Indicator

When performing a ping from UNTRUSTED to TRUSTED, the log messages are not generated. Ping TRUSTED from UNTRUSTED.

```
UNTRUSTED# ping 192.168.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
```

Why would you want to disable IPS signatures?

On FW, execute **show ip ips all** to view various IPS configuration parameters. Notice that one of the signatures is disabled by default in addition to the signature we disabled earlier.

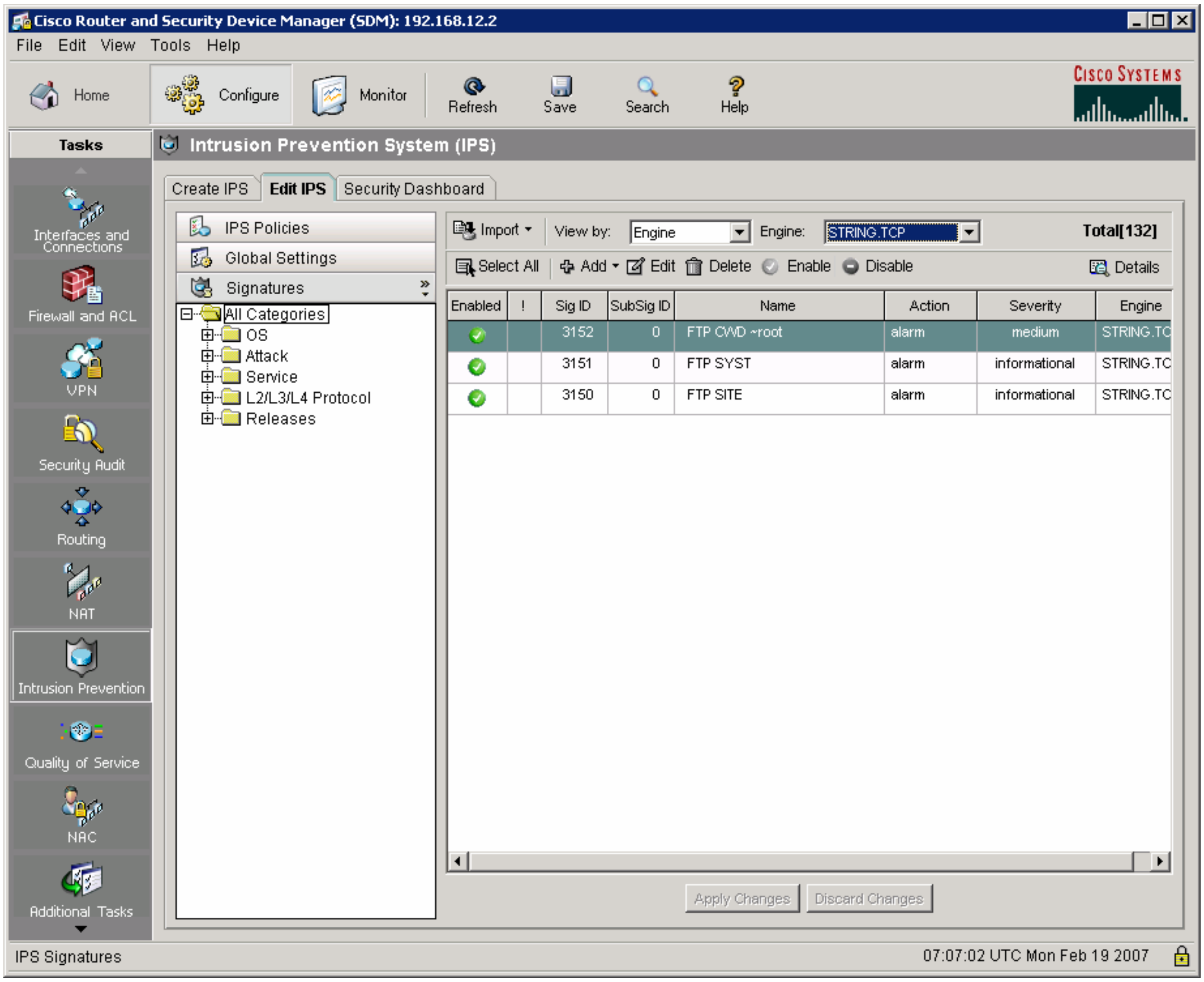
```
FW# show ip ips all
Configured SDF Locations:
 flash://sdmips.sdf
Builtin signatures are enabled and loaded
Last successful SDF load time: 08:01:10 UTC Feb 19 2007
IPS fail closed is disabled
Fastpath ips is enabled
Quick run mode is enabled
Event notification through syslog is enabled
Event notification through SDEE is enabled
Total Active Signatures: 132
Total Inactive Signatures: 0
Signature 1107:0 disable
Signature 2004:0 disable
IPS Rule Configuration
 IPS name sdm_ips_rule
Interface Configuration
 Interface Serial0/0/1
  Inbound IPS rule is sdm_ips_rule
  Outgoing IPS rule is not set
FW#
```

Note: If you are ending the lab here, delete the SDF file you created by using the **delete flash:sdmips.sdf** command in privileged EXEC mode.

Challenge: Add a Signature

Using SDM allows you to create custom IPS signatures. In this lab, we will create a signature that detects a certain string in Telnet text and will terminate the connection if found.

Under the **All Categories** level of the signature tree, filter the view by choosing **Engine** in the View By drop-down list. Also, choose **STRING.TCP** in the Engine drop-down list. Once the two drop-down boxes are selected properly and the view is filtered to only show string-based TCP signatures, drag down on the **Add** icon to **Add New...**.



View by: Engine Engine: STRING.TCP Total[132]

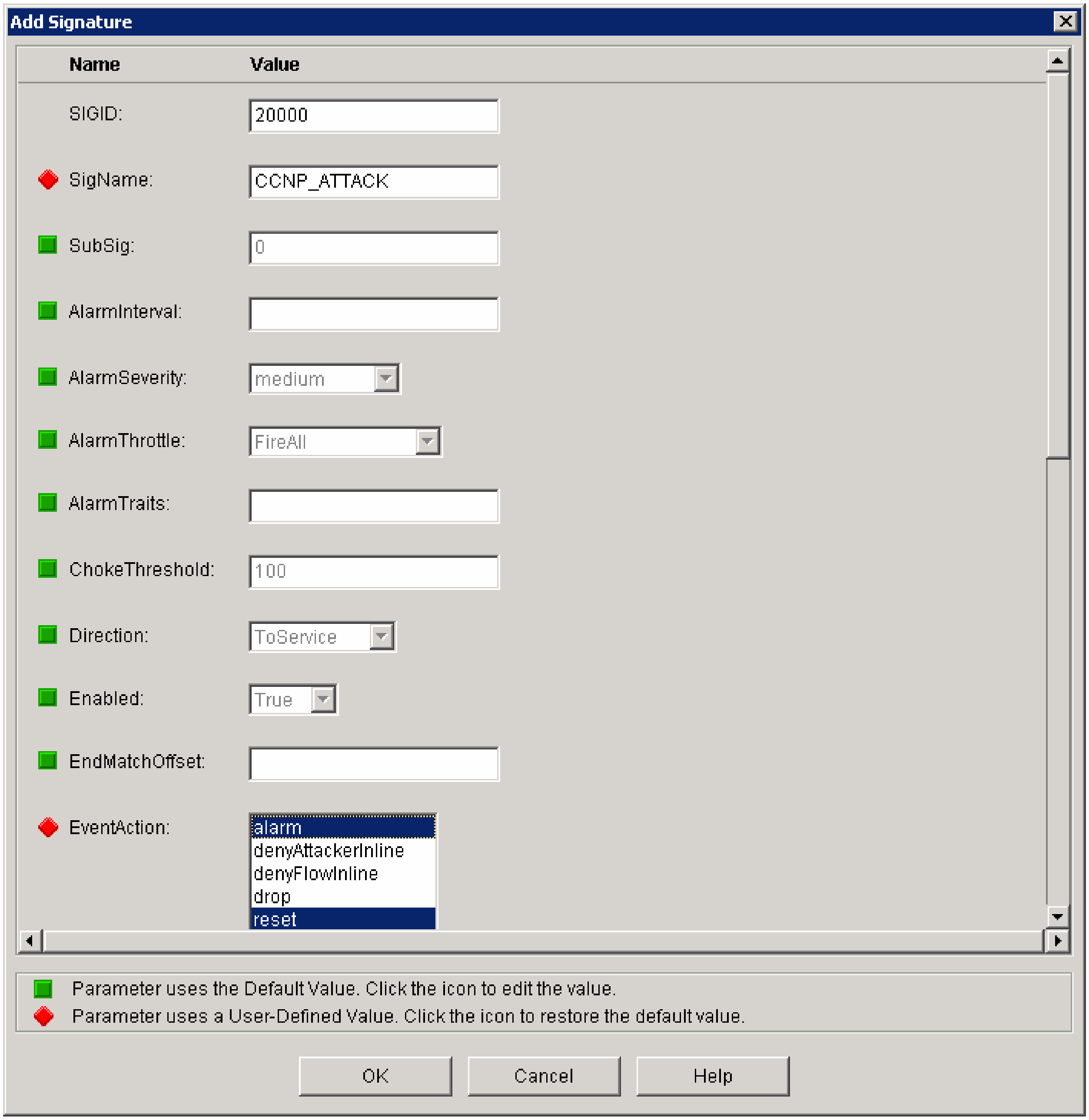
Enabled	!	Sig ID	SubSig ID	Name	Action	Severity	Engine
✓		3152	0	FTP CWD ~root	alarm	medium	STRING.TC
✓		3151	0	FTP SYST	alarm	informational	STRING.TC
✓		3150	0	FTP SITE	alarm	informational	STRING.TC

IPS Signatures 07:07:02 UTC Mon Feb 19 2007

Figure 7-1: IPS STRING.TCP Signatures

When creating or modifying signatures, to change the default values on certain fields, click the green square to the left of the field name and it will change to a red diamond, meaning the field will be changed from the default. This does not apply to all fields, only those with green squares.

Leave the default signature ID number as 20000 for this new signature. Name the signature “CCNP_ATTACK,” because the attack string will contain “CCNP” (as you will see subsequently). For EventAction, click the default action of “alarm” and click “reset” to. This will mean that when the signature is detected, a log message will be generated as well as sending a TCP reset to both sides of the connection, terminating the session. To select multiple actions, hold down the **ctrl** key on the keyboard while clicking each one with the mouse.



Add Signature

	Name	Value
	SIGID:	20000
♦	SigName:	CCNP_ATTACK
🟩	SubSig:	0
🟩	AlarmInterval:	
🟩	AlarmSeverity:	medium
🟩	AlarmThrottle:	FireAll
🟩	AlarmTraits:	
🟩	ChokeThreshold:	100
🟩	Direction:	ToService
🟩	Enabled:	True
🟩	EndMatchOffset:	
♦	EventAction:	alarm denyAttackerInline denyFlowInline drop reset

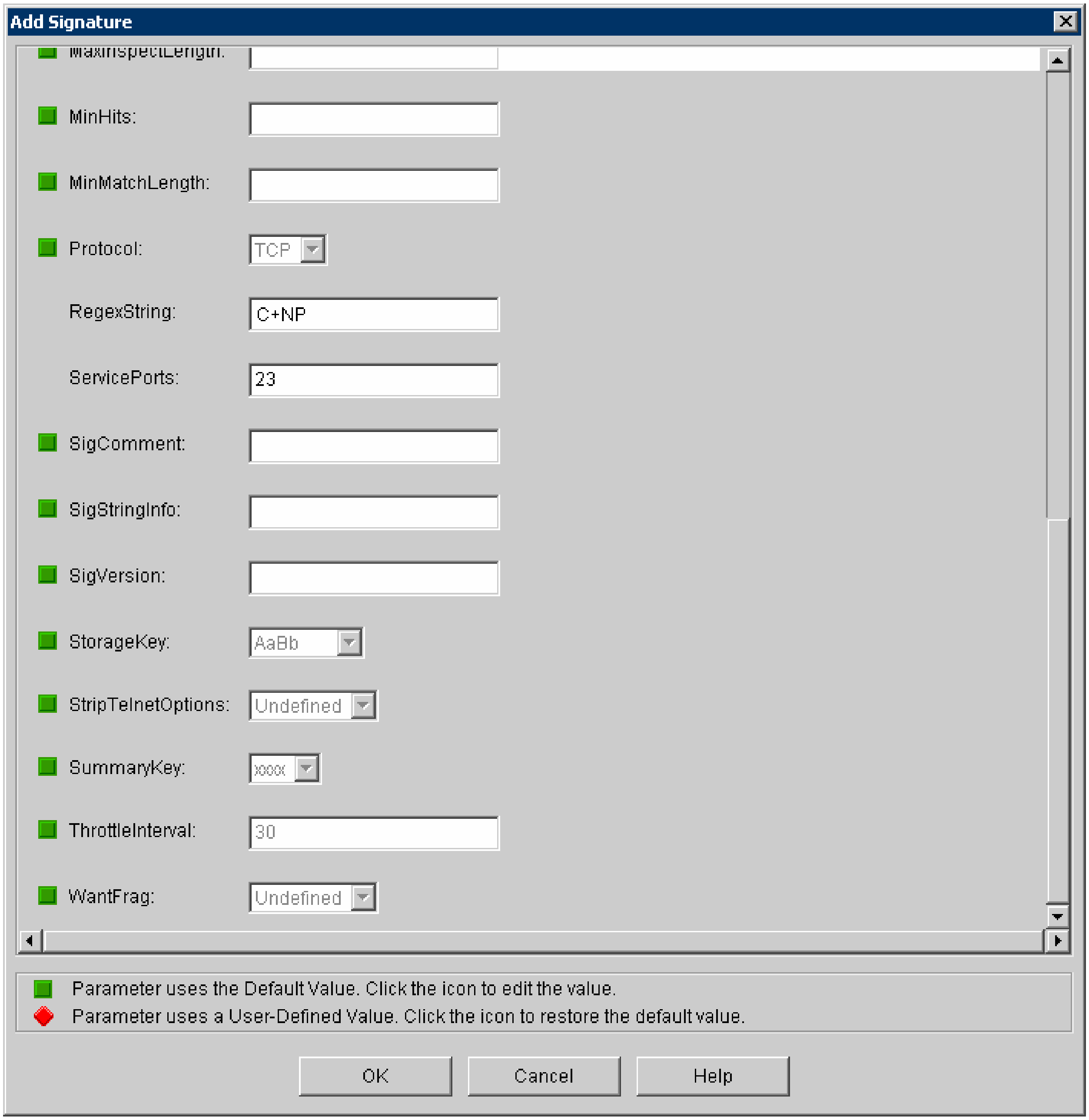
🟩 Parameter uses the Default Value. Click the icon to edit the value.

♦ Parameter uses a User-Defined Value. Click the icon to restore the default value.

Figure 7-2: IPS Add Signature Configuration Dialog

Toward the bottom of the add signature window, use a regular expression string of “C+NP” and a service port of 23. A regular expression is used for parsing and searching for certain strings. In this case, the “+” symbol indicates that there will be one or more of the characters before it in a string that matches. Regular expressions are case sensitive and must be matched exactly according to any regular expression symbols in it. In this case, TCP traffic matching the strings “CNP,” “CCNP,” “CCCNP,” and so forth, will match this signature. The service port simply tells the IPS engine to apply this signature to traffic on TCP port 23 (the Telnet port). Click **OK** when you are done.

For more information on regular expressions, consult the Cisco documentation.

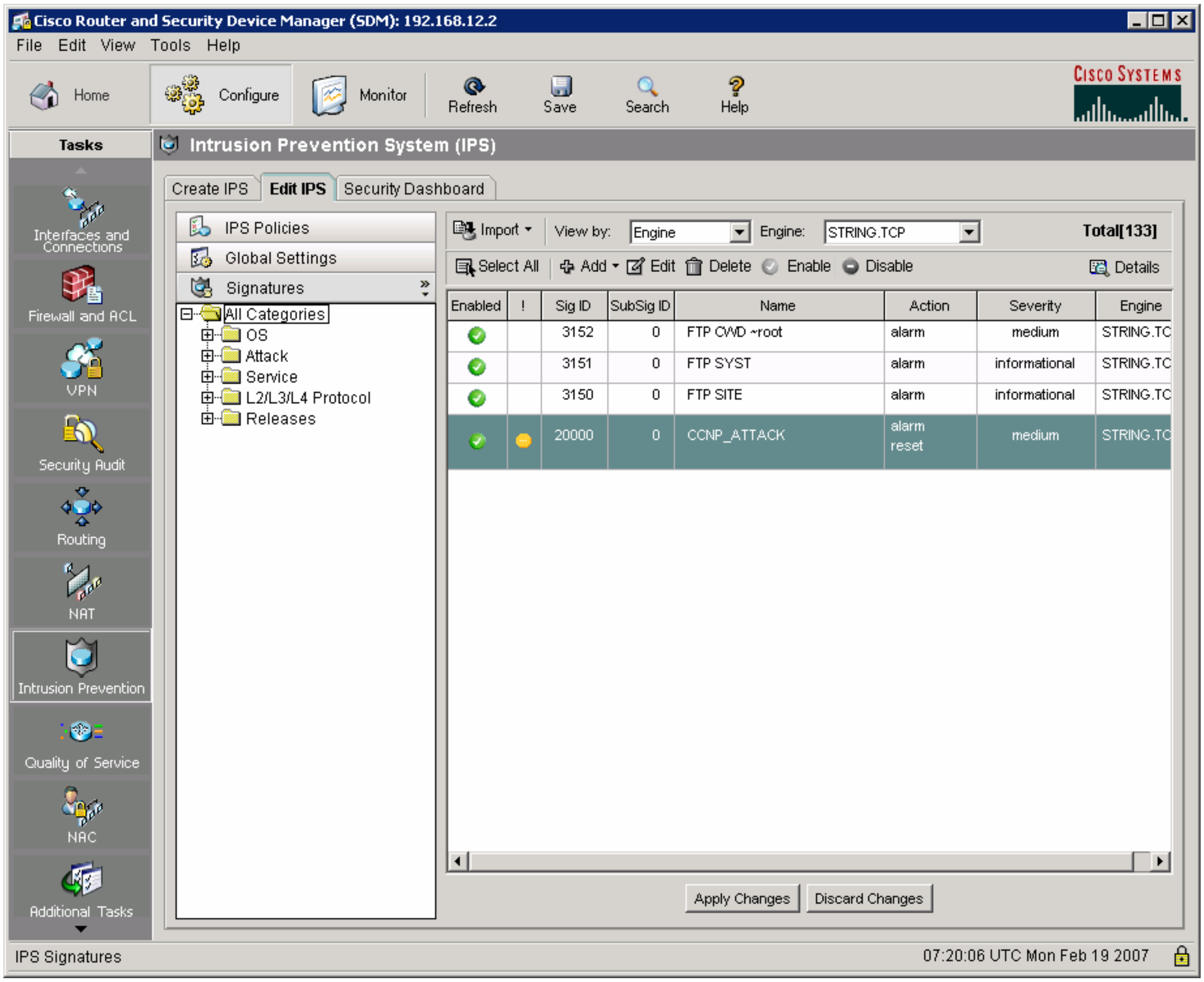


🟩	MaxInspectLength:	
🟩	MinHits:	
🟩	MinMatchLength:	
🟩	Protocol:	TCP
	RegexString:	C+NP
	ServicePorts:	23
🟩	SigComment:	
🟩	SigStringInfo:	
🟩	SigVersion:	
🟩	StorageKey:	AaBb
🟩	StripTelnetOptions:	Undefined
🟩	SummaryKey:	xxxx
🟩	ThrottleInterval:	30
🟩	WantFrag:	Undefined

🟩 Parameter uses the Default Value. Click the icon to edit the value.
♦ Parameter uses a User-Defined Value. Click the icon to restore the default value.

Figure 7-3: IPS Add Signature Regular Expression

After the new signature appears in the signature list for the STRING.TCP engine, click **Apply Changes**.



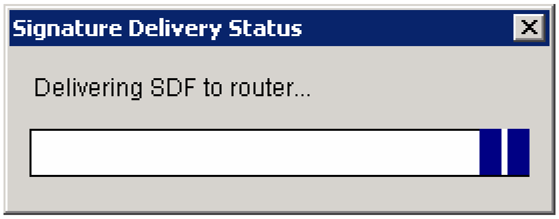
View by: Engine Engine: STRING.TCP Total[133]

Enabled	!	Sig ID	SubSig ID	Name	Action	Severity	Engine
✓		3152	0	FTP CWD ~root	alarm	medium	STRING.TC
✓		3151	0	FTP SYST	alarm	informational	STRING.TC
✓		3150	0	FTP SITE	alarm	informational	STRING.TC
✓	●	20000	0	CCNP_ATTACK	alarm reset	medium	STRING.TC

IPS Signatures 07:20:06 UTC Mon Feb 19 2007

Figure 7-4: IPS STRING.TCP Signatures, with New Signature

SDM will update the SDM it created earlier with the new signature.

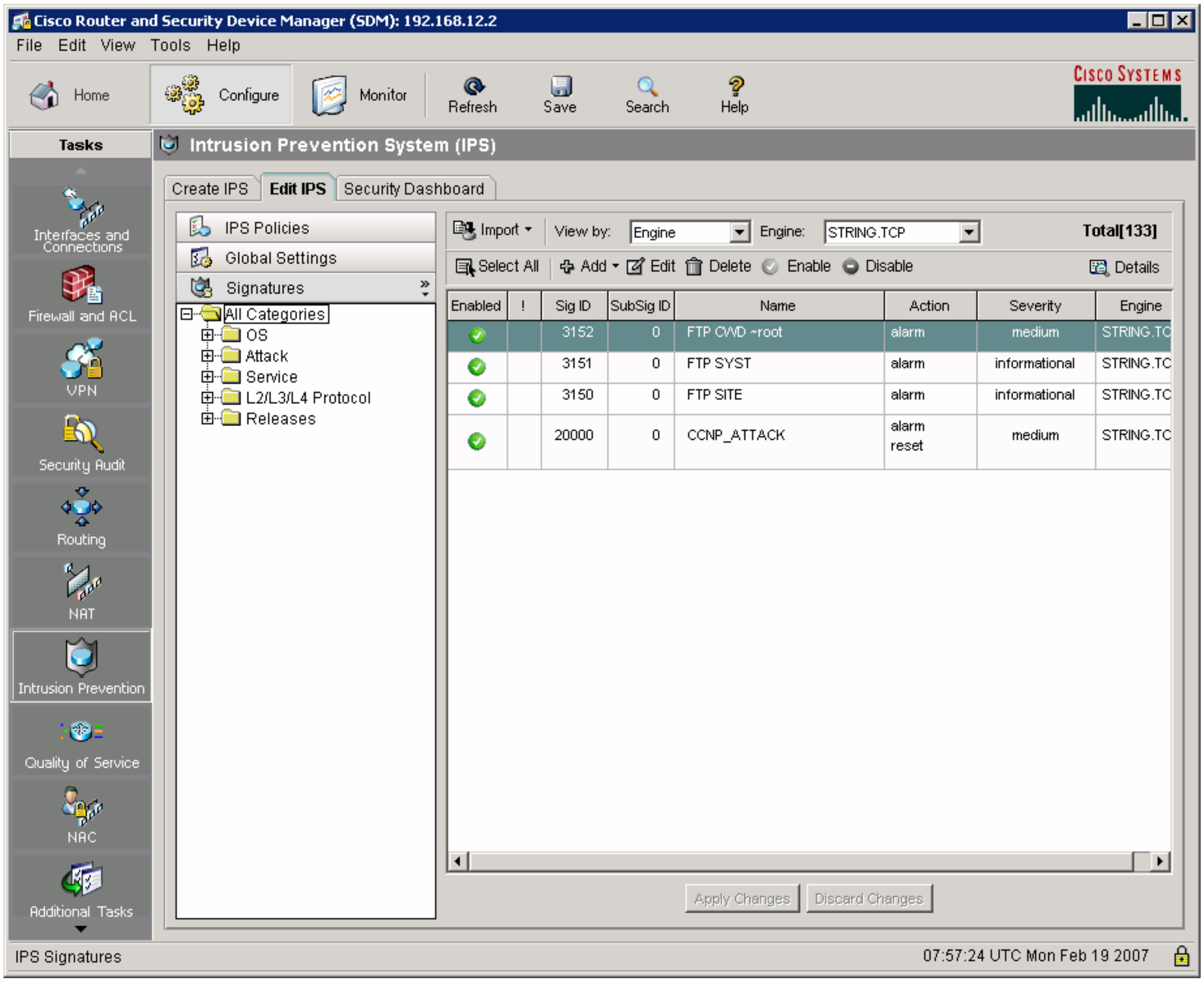


Signature Delivery Status

Delivering SDF to router...

Figure 7-5: IPS Signature Delivery Status

The new signature has been applied to the IPS. Packets with an ingress interface of Serial 0/0/1 will now be examined by IPS to check if they match the new signature as well.



View by: Engine Engine: STRING.TCP Total[133]

Enabled	!	Sig ID	SubSig ID	Name	Action	Severity	Engine
✓		3152	0	FTP CWD ~root	alarm	medium	STRING.TC
✓		3151	0	FTP SYST	alarm	informational	STRING.TC
✓		3150	0	FTP SITE	alarm	informational	STRING.TC
✓		20000	0	CCNP_ATTACK	alarm reset	medium	STRING.TC

Figure 7-6: IPS STRING.TCP Signatures, with Changes Applied

From UNTRUSTED, telnet to TRUSTED and type “CCNP” (this is case sensitive). Notice that IPS closes the Telnet session as soon as you type the “P.” The P is not even echoed from TRUSTED because the IPS closes the Telnet session before it is echoed. You can try this multiple times with a different number of Cs, since the signature will catch all strings for which there is more than one.

```
UNTRUSTED# telnet 192.168.12.1
Trying 192.168.12.1 ... Open


User Access Verification

Password:
TRUSTED> CCN
[Connection to 192.168.12.1 closed by foreign host]
UNTRUSTED# telnet 192.168.12.1
Trying 192.168.12.1 ... Open
```

```
User Access Verification

Password:
TRUSTED> CCCN
[Connection to 192.168.12.1 closed by foreign host]
UNTRUSTED#
```

The attempts are logged on FW.

```
FW#
*Feb 19 08:01:38.847: %IPS-4-SIGNATURE: Sig:20000 Subsig:0 Sev:4 CCNP_ATTACK
[192.168.23.3:33186 -> 192.168.12.1:23]
FW#
*Feb 19 08:01:50.175: %IPS-4-SIGNATURE: Sig:20000 Subsig:0 Sev:4 CCNP_ATTACK
[192.168.23.3:60904 -> 192.168.12.1:23]
```

Note: When you are done with the lab, delete the SDF file you created by using the privileged EXEC command **delete flash:sdmips.sdf**.

Final Configurations

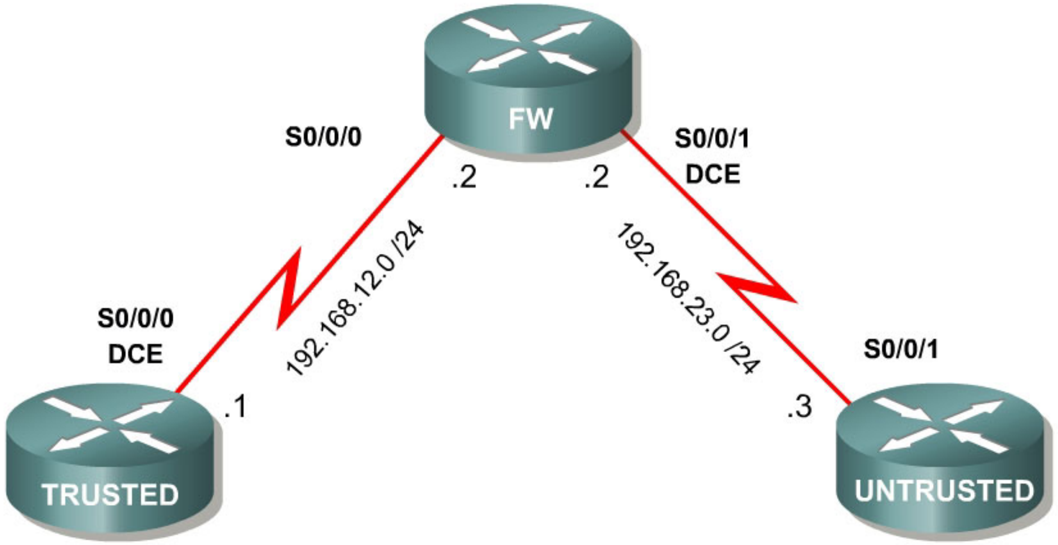
```
TRUSTED# show run
hostname TRUSTED
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 192.168.12.2
!
line vty 0 4
 password cisco
 login
end

FW# show run
hostname FW
!
ip ips sdf location flash://sdmips.sdf
ip ips notify SDEE
ip ips name sdm_ips_rule
!
crypto pki trustpoint TP-self-signed-3043721146
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-3043721146
 revocation-check none
 rsakeypair TP-self-signed-3043721146
!
crypto pki certificate chain TP-self-signed-3043721146
 certificate self-signed 01
  3082023A 308201A3 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303433 37323131 3436301E 170D3037 30323139 30373435
  35365A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D33 30343337
  32313134 3630819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100B33E 12B524DF 69D1AD2F D55A1A8C 3F6E45A5 1595DDFA C8FB946B 1EE7449C
  57BF61A2 5960CD54 816BFE11 411DFCDD DA159B9B 8CD34EA3 C92F0EE9 3B0251D8
  F341689D CDCC9A90 28E813BF 9555BE16 F6C6FE03 2E68E3E9 64924766 4264C47E
  939856EF 783FDE31 3DAB36EE 85D27B91 BF9EBC24 20854694 8ACDAD8A 955B77CF
  014B0203 010001A3 62306030 0F060355 1D130101 FF040530 030101FF 300D0603
  551D1104 06300482 02465730 1F060355 1D230418 30168014 BE06B151 CE3642B2
  335FAC80 62110167 3FA5E761 301D0603 551D0E04 160414BE 06B151CE 3642B233
  5FAC8062 1101673F A5E76130 0D06092A 864886F7 0D010104 05000381 810035BB
  291CFD03 B6B65C69 96922357 4A1E19F6 B81D2533 E58FE0CE E73CC6D2 B610F5E1
  08F10391 9303BCEE 3D587635 DE4546D6 AC86A980 B6412DF2 1FA73933 8BEEDAF2
  5A6A4D25 E4B1F88E 2C41F2CD A2FE72D3 0DC048CC A7EBC057 C238E46D 4C848298
  67059914 5D8743B1 E287C470 1EFB7CE0 98A833F4 D22E641D 4C3C0C05 E360
  quit
username ciscosdm privilege 15 password 0 ciscosdm
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 ip virtual-reassembly
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 ip ips sdm_ips_rule in
 ip virtual-reassembly
 clock rate 64000
 no shutdown
!
ip http server
ip http authentication local
ip http secure-server
!
line vty 0 4
 login local
end

UNTRUSTED# show run
hostname UNTRUSTED
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 192.168.23.2
end
```

Lab 6.4 Configuring IPS with CLI

Learning Objectives

- Configure Cisco IOS IPS on an interface
- Disable unwanted IPS signatures
- Verify IPS settings

Topology Diagram



Scenario

In this lab, you will configure the Cisco IOS Intrusion Prevention System (IPS), which is part of the Cisco IOS Firewall feature set. IPS examines certain attack patterns and will alert and/or mitigate when those patterns occur.

In this scenario, TRUSTED represents a trusted inside router, FW represents a router serving as an intrusion prevention router, and UNTRUSTED represents an untrusted outside router. Since UNTRUSTED is outside, FW will examine packets inbound from it. IPS alone is not enough to make a router into a secure Internet firewall, but in addition to other security features it can be a powerful defense.

Step 1: Configure Addressing

Configure the serial interfaces shown in the diagram. Set the clock rate on the appropriate interface and issue the **no shutdown** command on all serial connections. Verify that you have connectivity across the local subnet using the **ping** command.

```
TRUSTED(config)# interface serial0/0/0
TRUSTED(config-if)# ip address 192.168.12.1 255.255.255.0
TRUSTED(config-if)# clockrate 64000
TRUSTED(config-if)# no shutdown

FW(config)# interface serial0/0/0
FW(config-if)# ip address 192.168.12.2 255.255.255.0
FW(config-if)# no shutdown
FW(config-if)# interface serial0/0/1
FW(config-if)# ip address 192.168.23.2 255.255.255.0
FW(config-if)# clockrate 64000
FW(config-if)# no shutdown

UNTRUSTED(config)# interface serial0/0/1
UNTRUSTED(config-if)# ip address 192.168.23.3 255.255.255.0
UNTRUSTED(config-if)# no shutdown
```

Step 2: Configure Static Default Routes

On the TRUSTED and UNTRUSTED routers, configure static default routes directing traffic to unknown destinations to be forwarded to the FW router. FW will not need any routes because it has interfaces directly connected to both networks.

```
TRUSTED(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.2

UNTRUSTED(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2
```

Your network should have full IP connectivity at this point. If it does not have full connectivity, troubleshoot.

Normally a single-homed company might use Network Address Translation (NAT) at its corporate edge to protect its network and allow private addressing within the bounds of its network. In that case, the UNTRUSTED router, normally a provider edge router would have a static route directing traffic to the address owned by the customer out one of its interfaces. In this scenario you will not configure NAT, and you will use a default route for simplicity.

Step 3: Create and Apply an IPS rule

On FW, create an IPS rule using the **ip ips name** *name* command in global configuration mode. Use the name “myips” for the ips name. Although it will not be used in this lab, to create an IPS rule that only operates on certain traffic (specified with an access list), you would use the **ip ips name** *name* **list** *list* command in global configuration mode.

```
FW(config)# ip ips name myips
```

Note: This lab does not cover using signatures from a signature definition file (SDF). Instead it uses the built-in signatures in the IOS Firewall software. In a production environment, you might want to specify an SDF file using the **ip ips sdf location** *location* command in global configuration mode, where *location* is a relative location to the router, such as “flash:128MB.sdf” which indicates that an SDF file in the router’s flash memory named “128MB.sdf” will be used.

Apply the IPS rule to an interface with the **ip ips** *name direction* command in interface configuration mode. Apply the rule you just created inbound on the interface facing UNTRUSTED. Once you enable IPS, some log messages will be sent to the console line indicating that the IPS engines are being initialized.

```
FW(config)# interface serial0/0/1
FW(config-if)# ip ips myips in
*Feb 19 09:16:09.923: %IPS-6-BUILTIN_SIGS: Configured to load builtin
signatures
*Feb 19 09:16:10.067: %IPS-6-SDF_LOAD_SUCCESS: SDF loaded successfully from
builtin
*Feb 19 09:16:10.075: %IPS-6-ENGINE_BUILDING: OTHER - 3 signatures - 1 of 15
engines
*Feb 19 09:16:10.075: %IPS-6-ENGINE_READY: OTHER - 0 ms - packets for this
engine will be scanned
*Feb 19 09:16:10.075: %IPS-6-ENGINE_BUILDING: MULTI-STRING - 0 signatures - 2
of 15 engines
*Feb 19 09:16:10.075: %IPS-6-ENGINE_BUILD_SKIPPED: MULTI-STRING - there are no
new signature definitions for this engine
*Feb 19 09:16:10.075: %IPS-6-ENGINE_BUILDING: STRING.ICMP - 0 signatures - 3
of 15 engines
*Feb 19 09:16:10.075: %IPS-6-ENGINE_BUILD_SKIPPED: STRING.ICMP - there are no
new signature definitions for this engine
<OUTPUT OMITTED>
```

Step 4: Modify Default IPS Behavior

From UNTRUSTED, **ping** TRUSTED with a high repeat count.

```
UNTRUSTED# ping 192.168.12.1 repeat 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 56/56/88 ms
```

Notice the log messages on FW. It starts off logging each signature match, but after 62 hits, IPS only reports a summary count without logging each match individually.

```
 FW#
*Feb 19 09:30:41.823: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:41.879: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:41.935: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:41.991: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.051: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.107: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
```

```
*Feb 19 09:30:42.163: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.219: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.275: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.335: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.391: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.447: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.903: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:42.959: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.019: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.075: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.131: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.187: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.247: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.303: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.359: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.415: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.927: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:43.987: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.043: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.099: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.159: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.215: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.271: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.327: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.387: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.443: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.899: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:44.955: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.011: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.067: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.127: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
```

```
*Feb 19 09:30:45.183: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.239: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.295: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.355: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.411: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.931: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:45.987: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.047: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.103: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.159: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.215: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.275: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.331: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.387: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.443: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.923: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:46.979: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.039: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.095: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.151: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.207: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.263: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.323: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.379: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
*Feb 19 09:30:47.435: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req
[192.168.23.3:0 -> 192.168.12.1:0]
 *Feb 19 09:31:11.823: %IPS-4-SIG_SUMMARY: Sig:2004 Subsig:0 Global Summary:
 100 alarms in this interval
```

To disable a signature on a router, use the **ip ips signature** *number* **disable** command in global configuration mode. Signature number 2004 detected the previous potential Internet Control Message Protocol (ICMP) attack indicated in the previous output. Disable the signature numbered 2004, which was being set off by the ping previously.

```
FW(config)# ip ips signature 2004 disable
%IPS Signature 2004:0 is disabled
```

Try the **ping** again from UNTRUSTED to TRUSTED. You do not need a high repeat count because there will be no summary messages (since there will not be any matches to begin with).

```
UNTRUSTED# ping 192.168.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
```

Notice that no individual or summary IPS messages have been logged on the FW router.

Why would you want to disable IPS signatures?

You can also verify the current IPS configuration by using the command **show ip ips all**. Notice that in addition to the signature we just disabled, there was another signature disabled by default.

```
FW# show ip ips all
Configured SDF Locations: none
Builtin signatures are enabled and loaded
Last successful SDF load time: 09:29:43 UTC Feb 19 2007
IPS fail closed is disabled
Fastpath ips is enabled
Quick run mode is enabled
Event notification through syslog is enabled
Event notification through SDEE is disabled
Total Active Signatures: 132
Total Inactive Signatures: 0
Signature 2004:0 disable
Signature 1107:0 disable
IPS Rule Configuration
 IPS name myips
Interface Configuration
 Interface Serial0/0/1
  Inbound IPS rule is myips
  Outgoing IPS rule is not set
```

Notice that you can use the syslog protocol to report detections and IPS actions to management stations. For the intrusion detection feature, the IPS router will simply send the notifications to the syslog server.

Final Configurations

```
TRUSTED# show run
hostname TRUSTED
!
interface Serial0/0/0
```

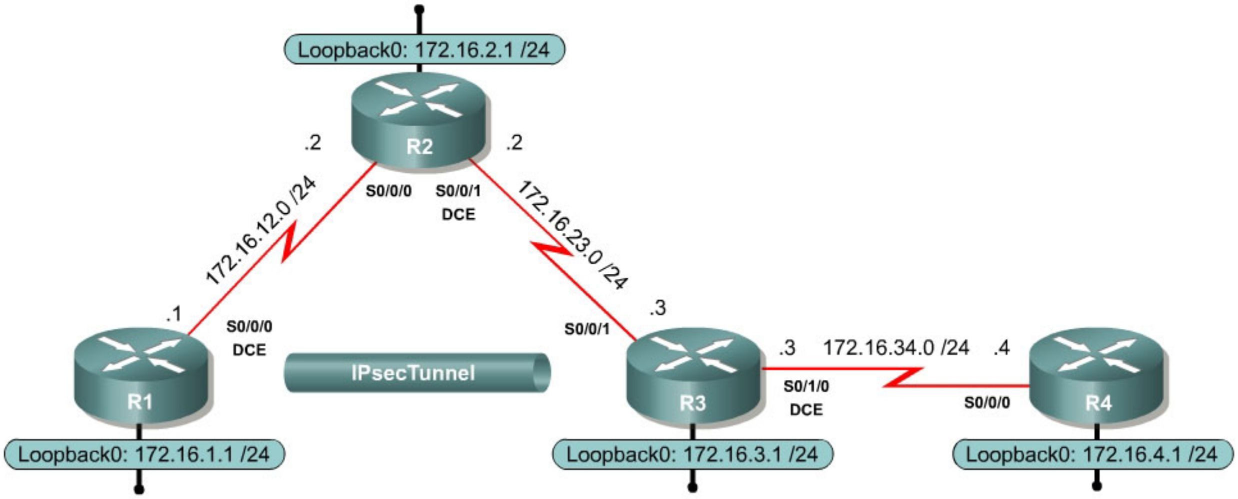
```
 ip address 192.168.12.1 255.255.255.0
 clock rate 64000
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 192.168.12.2
end

FW# show run
hostname FW
!
ip ips signature 2004 0 disable
ip ips name myips
!
interface Serial0/0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.23.2 255.255.255.0
 ip ips myips in
 clock rate 64000
 no shutdown
!
end

UNTRUSTED# show run
hostname UNTRUSTED
!
interface Serial0/0/1
 ip address 192.168.23.3 255.255.255.0
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 192.168.23.2
end
```

Case Study 1: CLI IPsec and Frame-Mode MPLS

Instructions

Implement the International Travel Agency network shown in the topology diagram using the information and instructions in the scenario. Implement the design on the lab set of routers. Verify that all configurations are operational and functioning according to the guidelines.

Topology Diagram



Scenario

The International Travel Agency needs parts of its network set up with IPsec and Multiprotocol Label Switching (MPLS) with the given specifications and the topology shown in the diagram. This case study should be completed using the Cisco IOS command-line interface (CLI), without using the Cisco Security Device Manager (SDM).

- Configure all interfaces using the addressing scheme shown in the topology diagram.
- Run Enhanced Interior Gateway Routing Protocol (EIGRP) AS 1 in the entire International Travel Agency core network. All subnets should be included.
- Create an IPsec tunnel between R1 and R3 with an appropriate transform set and Internet Security Association and Key Management Protocol (ISAKMP) policy.
- This IPsec tunnel should only encrypt traffic between R1's loopback network and R4's loopback network.
- Use pre-shared keys for authentication in the ISAKMP policy.

- Do not create any new interfaces to achieve this task.
- Use any encryption algorithms desired for the tasks listed above that use the crypto suite of protocols.
- Configure MPLS on both ends of the link between R3 and R4.
- Configure R1 to send system logging messages at the error severity level to an imaginary host located at 172.16.2.200.
- Set up the correct time on R4 using the **clock set** command. Use the inline IOS help system if you do not know the syntax of this command.
- Configure R4 as an Network Time Protocol (NTP) master with stratum 5.
- Configure R3 as an NTP client of R4.

Questions

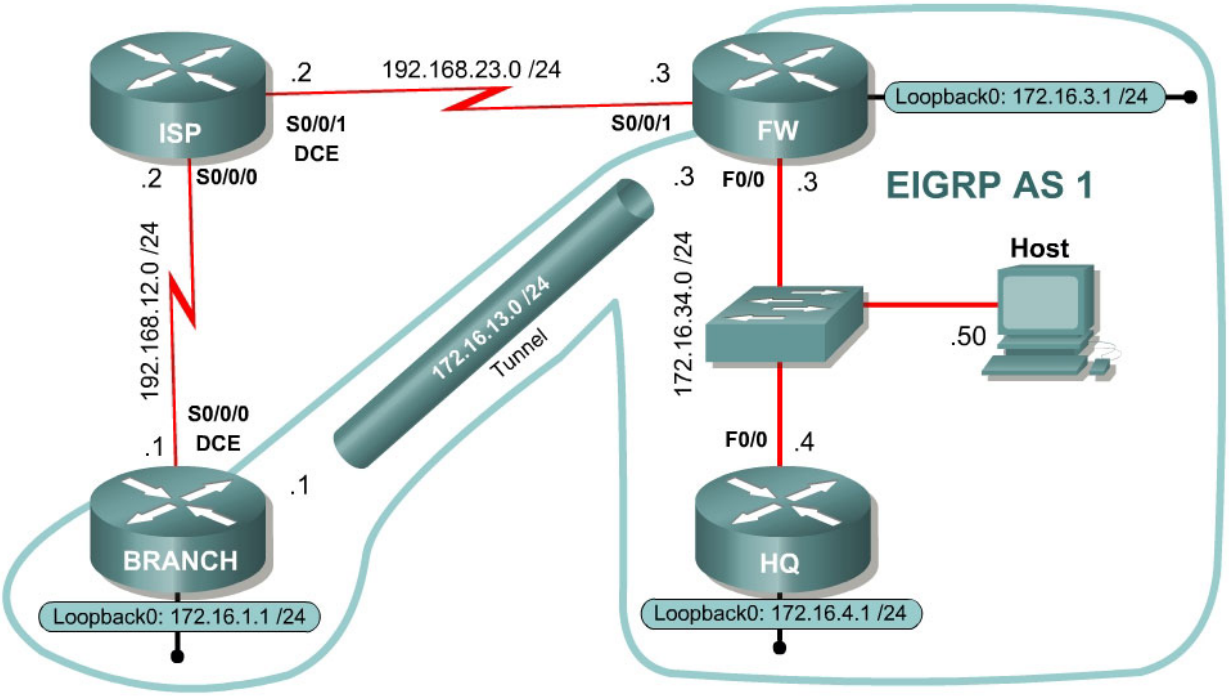
- Will R3 or R4 send the NTP queries as MPLS frames? Explain.
- Will R3 or R4 send any packets destined to the other router as MPLS frames? Explain.
- Will R3 or R4 send any packets at all as MPLS frames? Explain.
- Differentiate among the algorithms by explaining which algorithms in your IPsec policy apply to encryption, which to authentication, and which to message integrity. According to your reading, which of the available algorithms in each category is most secure?
- How does NTP help prepare a network for system logging? Explain.

Case Study 2: SDM

Instructions

Implement the International Travel Agency network shown in the topology diagram using the information and instruction in the scenario. Verify that all configurations are operational and functioning according to the guidelines.

Topology Diagram



Scenario

The International Travel Agency has decided to extend its offices to a new branch location using its existing network tunnel. The CIO has chosen to use a secure generic routing encapsulation (GRE) tunnel to connect the branch office to its headquarters office. The tunnel will terminate at the headquarters end on a Cisco router with the firewall feature set.

Demonstrate that this configuration will allow routing between sites as well as secure intranet traffic as it traverses the service provider’s domain. Implement the security policies defined below both on FW router and the BRANCH router.

Use the Cisco Security Device Manager (SDM) to configure the security tasks except where noted below.

Using the Cisco IOS CLI:

- Configure all interfaces using the addressing scheme shown in the topology diagram.
- Configure HQ, FW, and BRANCH to run Enhanced Interior Gateway Routing Protocol (EIGRP) in AS 1. (Until the tunnel is created, BRANCH will not have any EIGRP adjacencies.)
- Add the major 172.16.0.0 network to EIGRP and disable automatic summarization.
- Configure a static default route on FW towards ISP, and redistribute this into EIGRP.
- Configure a static default route on BRANCH toward ISP.
- Create a static route on ISP for 172.16.0.0/16 toward FW.
- After configuring the static routes, make sure you can ping between FW and BRANCH.
- Configure the host with the IP address shown in the topology diagram and make FW its default gateway.
- Configure FW and BRANCH for SDM access from the host.

Using Cisco SDM:

- Create a secure GRE tunnel between FW and BRANCH using IPsec.
- Use the addressing shown on the diagram for the tunnel addressing.
- Run EIGRP across the tunnel.
- You should use the tunnel wizard to configure one end of the tunnel, and generate a mirror configuration using Cisco SDM for the other end. You may use the command-line interface (CLI) to implement the mirror tunnel configuration on BRANCH.
- Apply any encryption algorithms desired for the secure GRE tunnel.
- Configure FW as a firewall using the basic firewall wizard. Assign the interface facing the ISP router to be the outside interface. Trust traffic from all other interfaces.
- If SDM does not automatically allow IPsec traffic through the firewall, explicitly allow it.
- Use the SDM IPS wizard to configure BRANCH to enable the intrusion prevention system (IPS) on the ingress interface facing the ISP router.