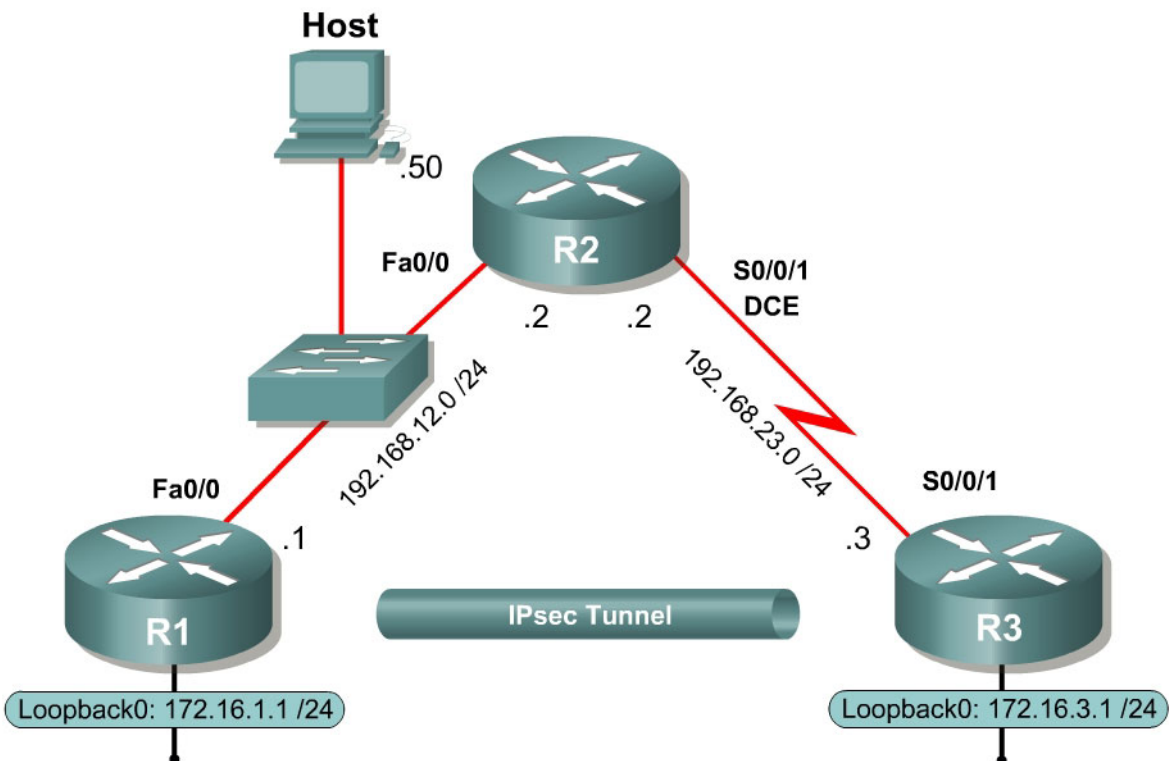


Lab 3.4 Configuring Site-to-Site IPsec VPNs with SDM

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 }
```

```
telquit
```

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.

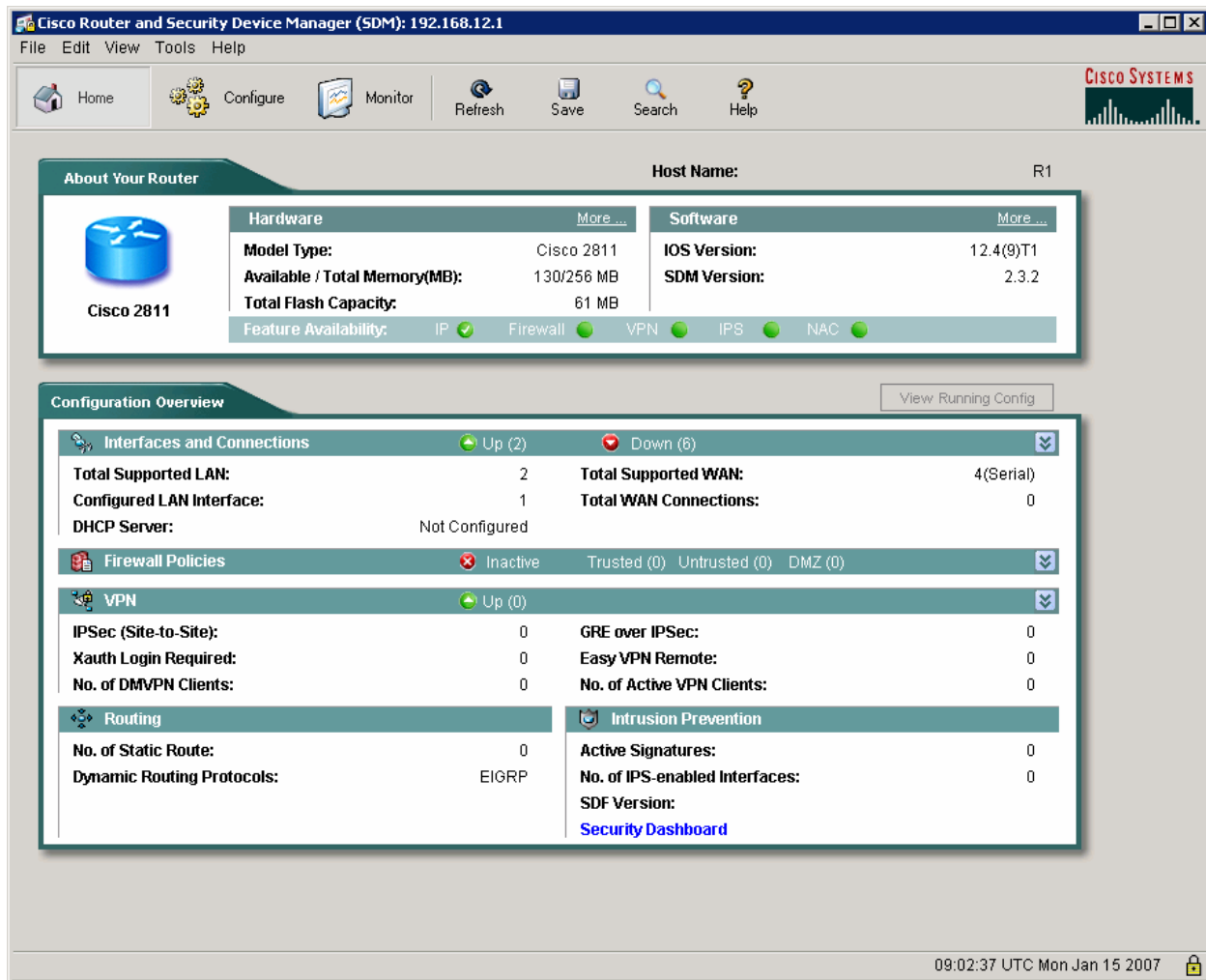


Figure 3-1: SDM Home Page

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.

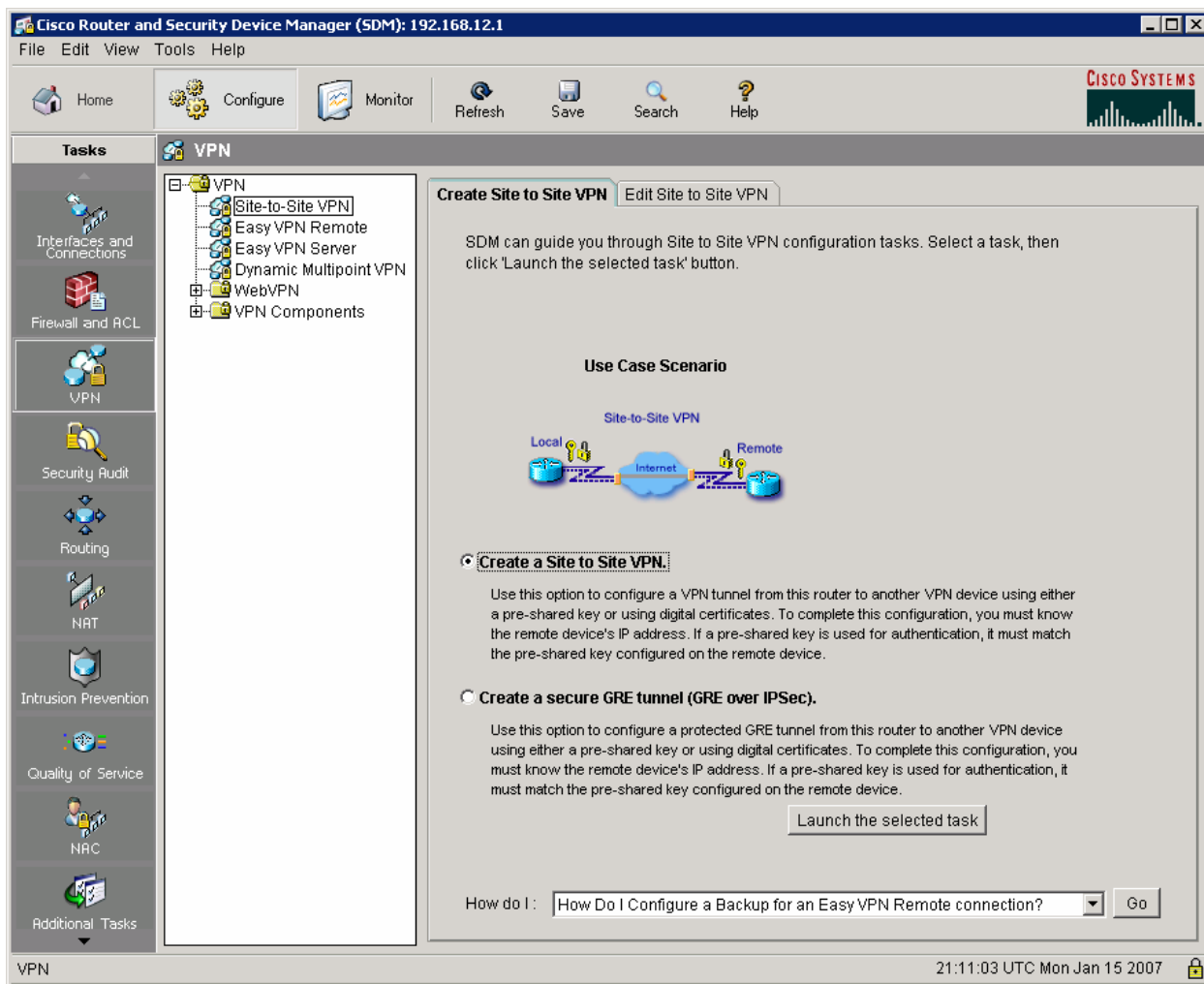


Figure 4-1: VPN Configuration Screen

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.

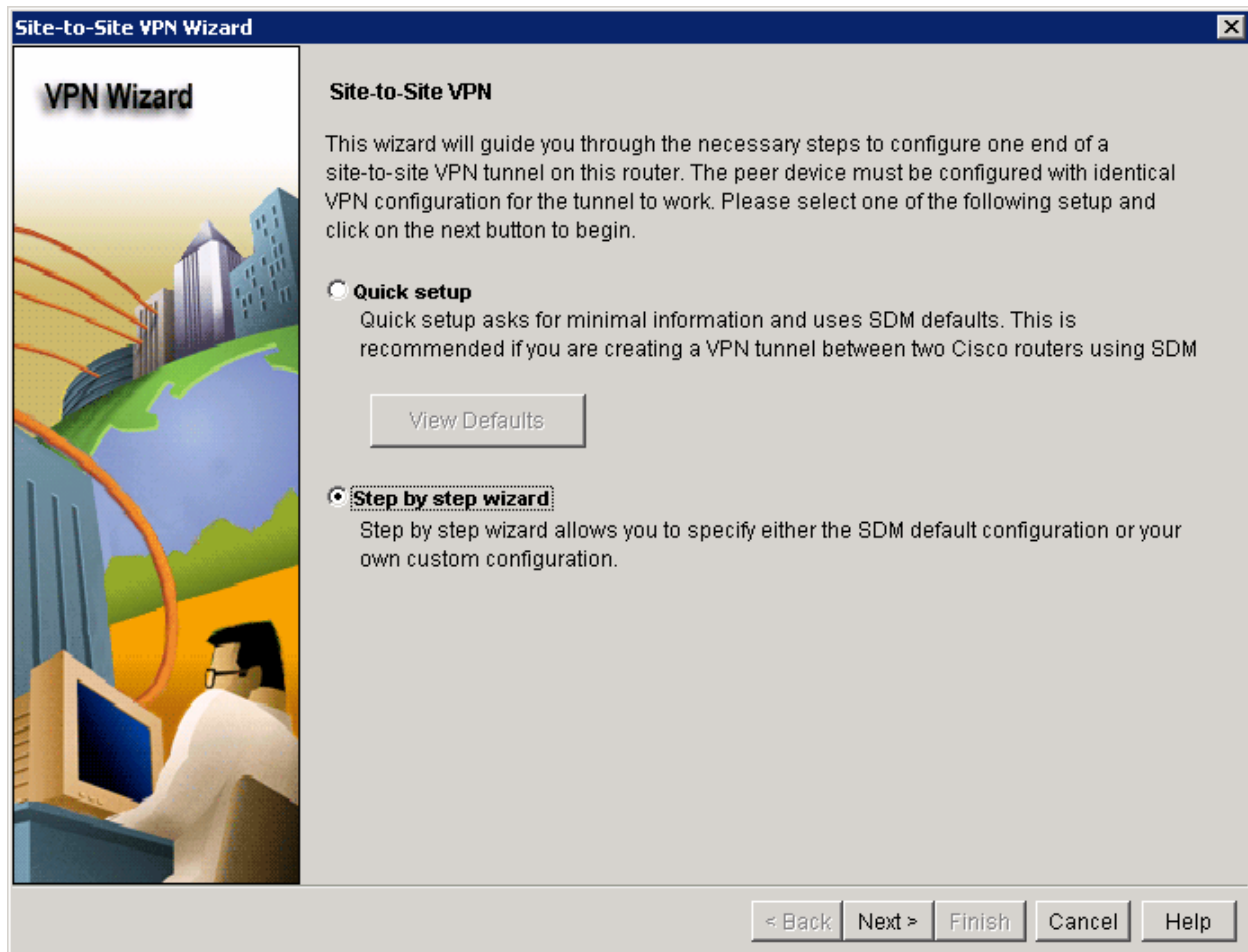


Figure 4-2: Site-to-Site VPN Wizard

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**.

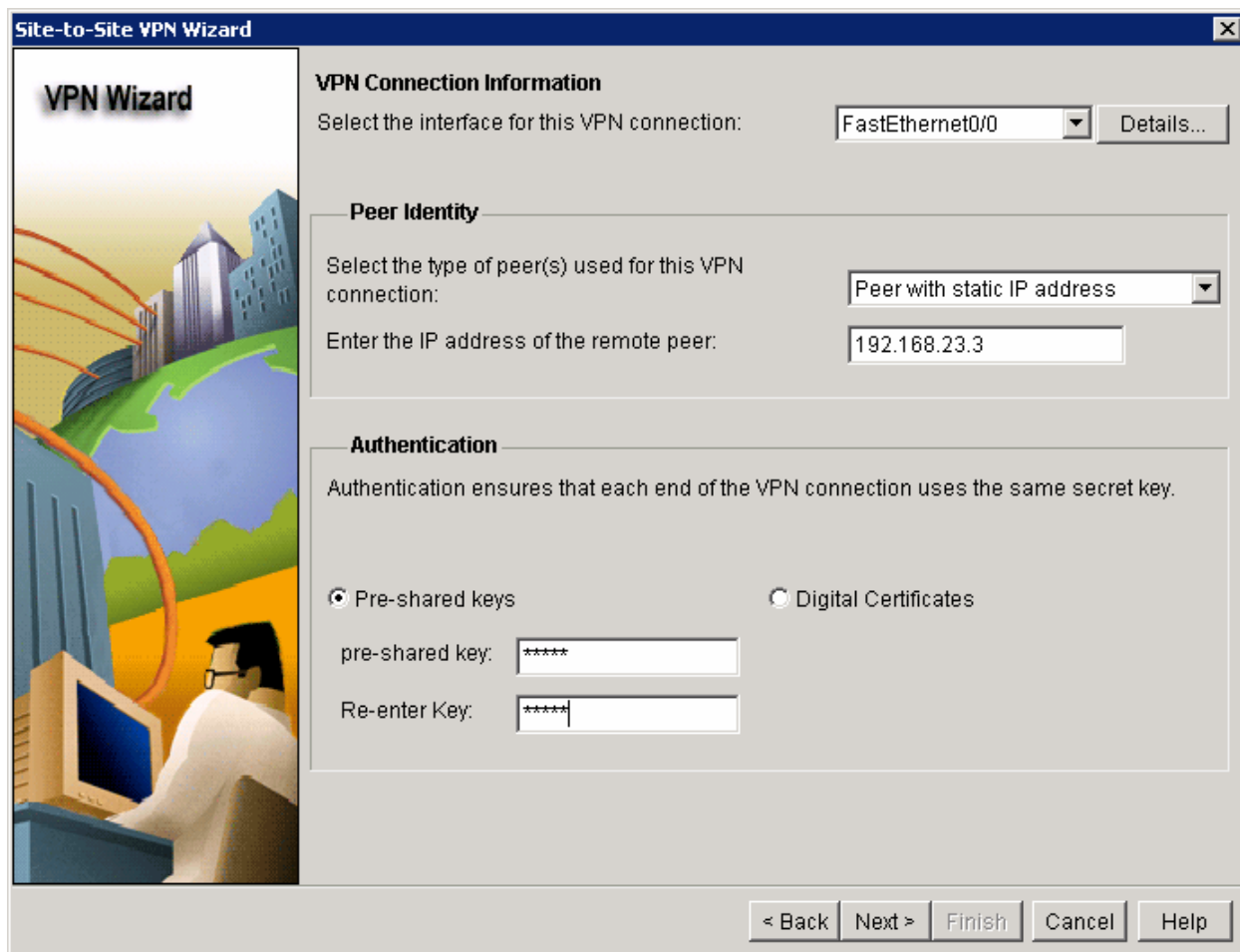


Figure 4-3: VPN Connection and Authentication Information

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.

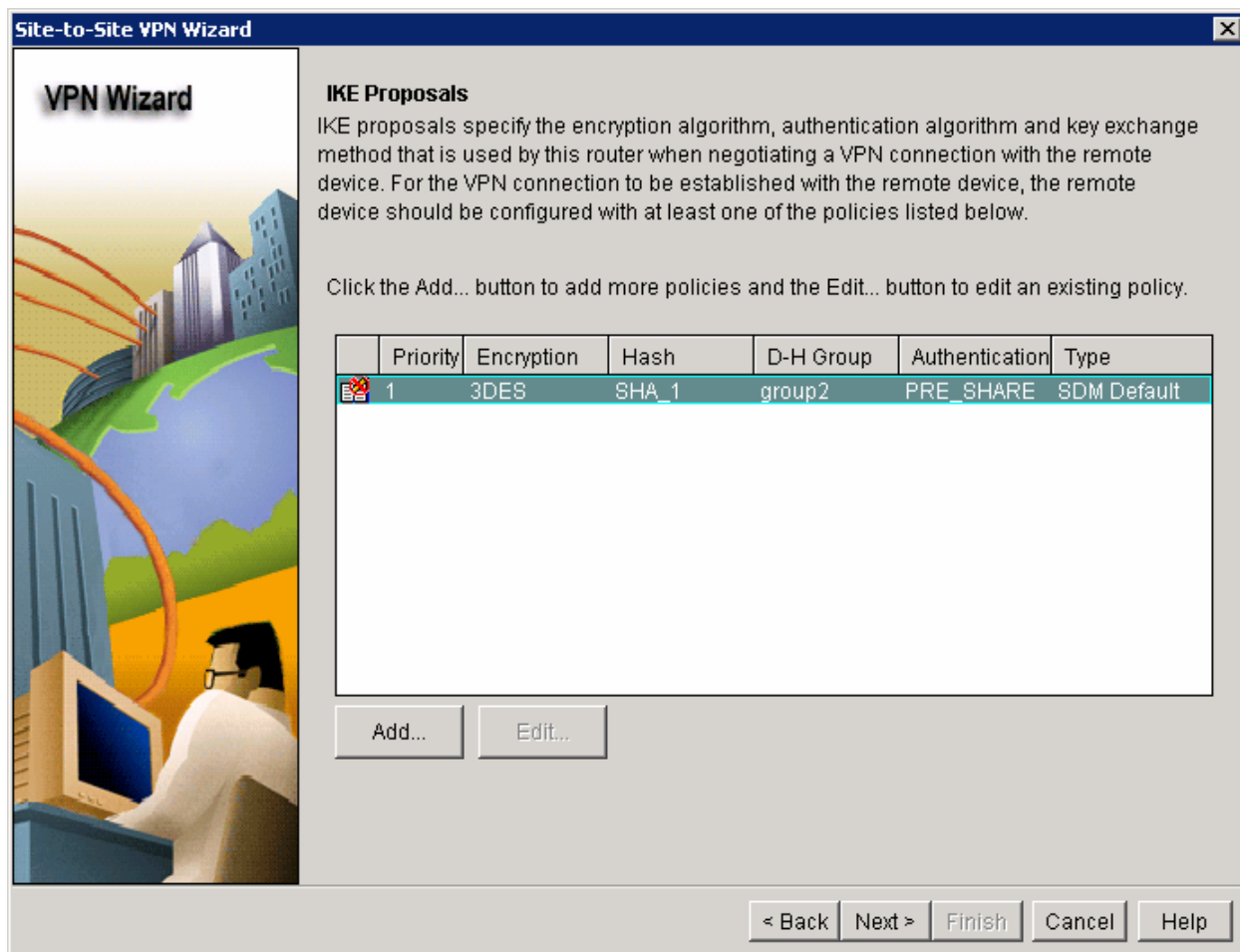


Figure 4-4: IKE Proposals List

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.

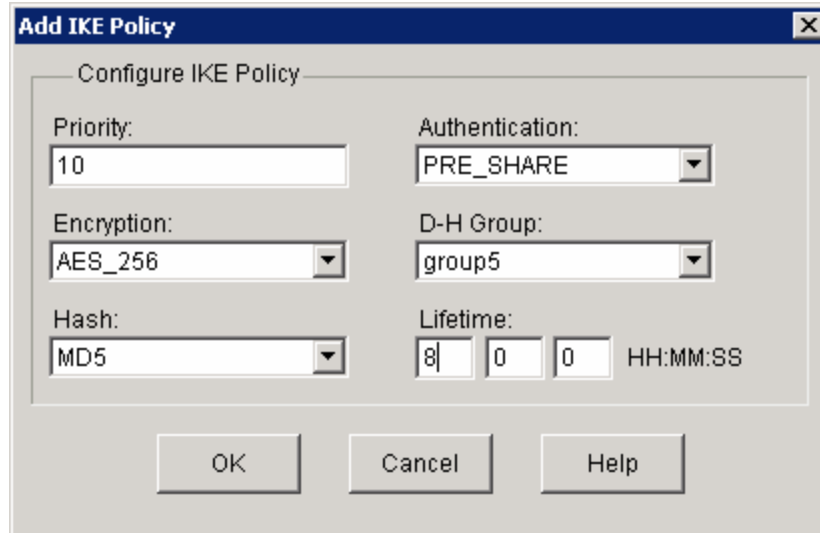


Figure 4-5: Add IKE Policy Dialog

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**.

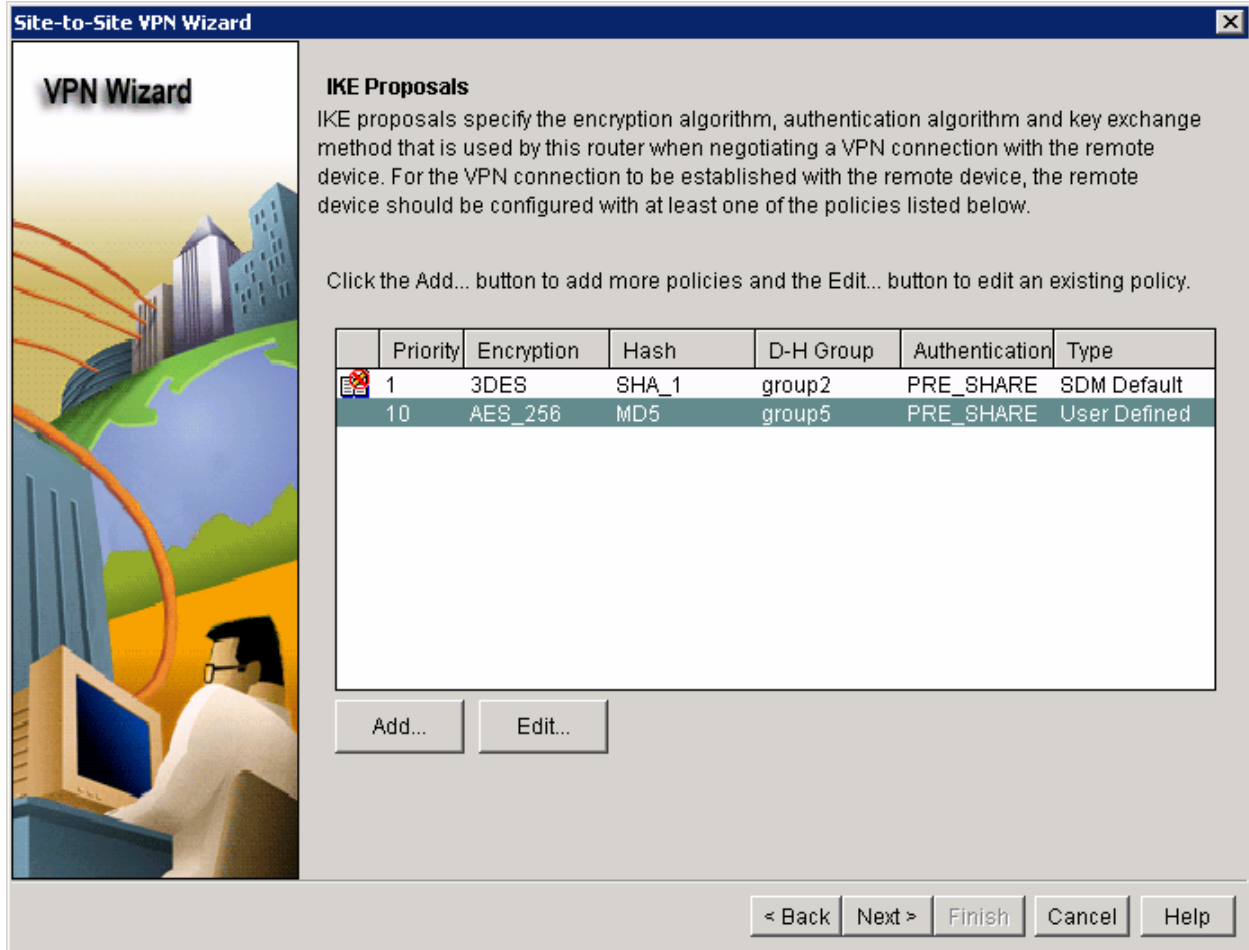


Figure 4-6: IKE Proposals with Changes Applied

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

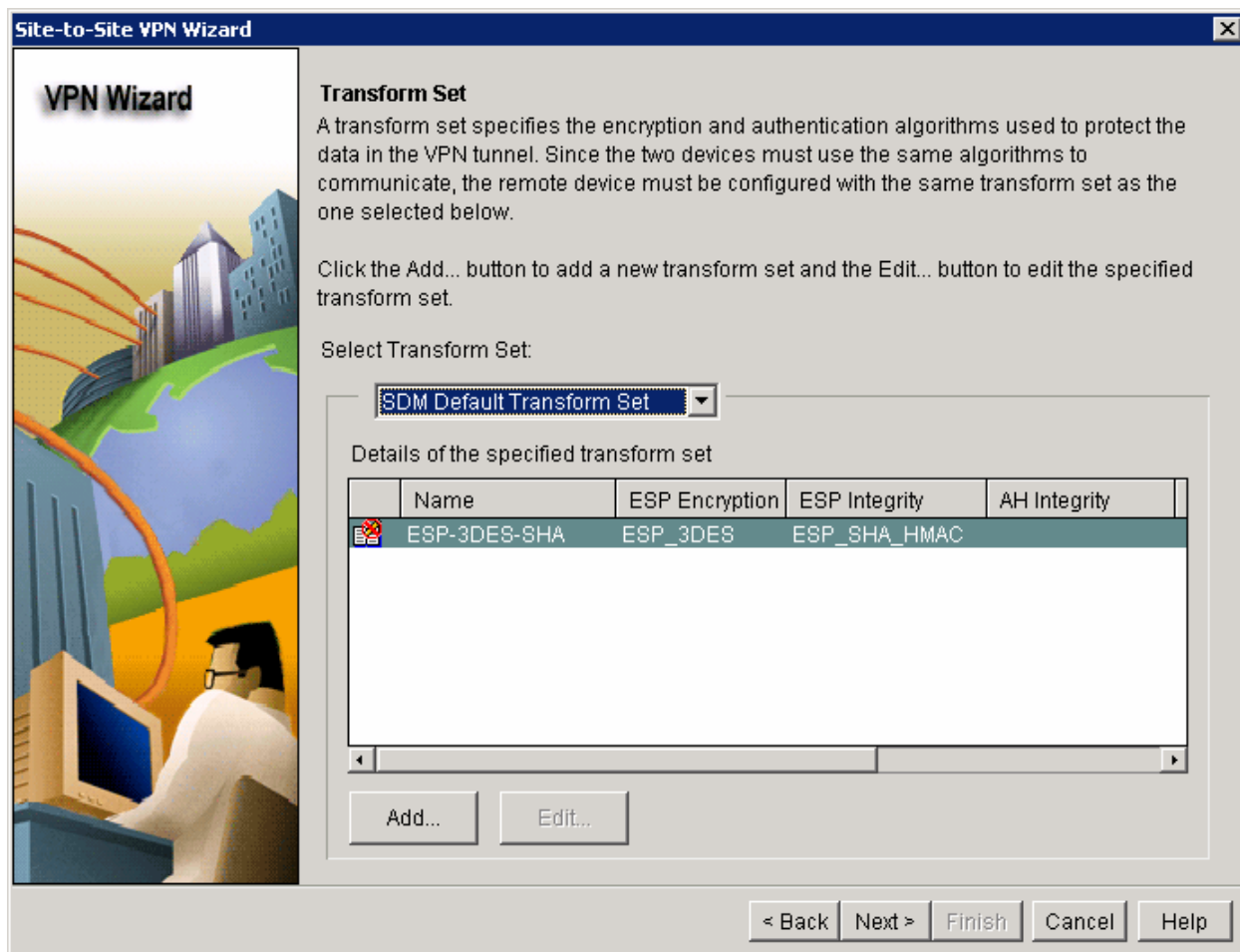


Figure 4-7: IPsec Transform Set List

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.

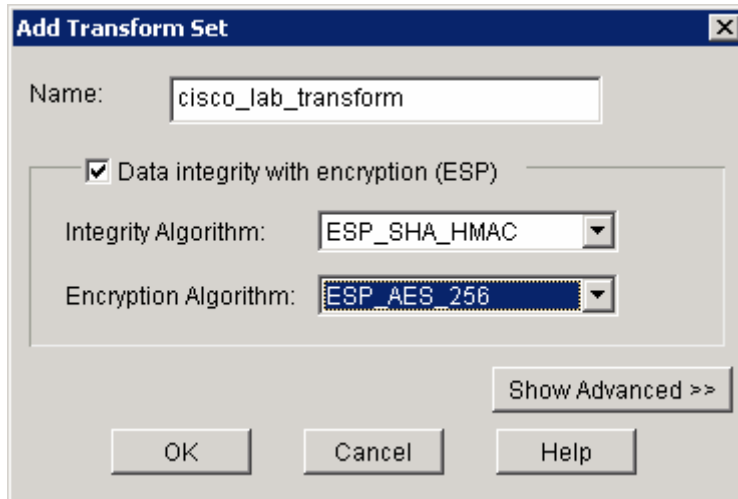


Figure 4-8: Add IPsec Transform Set Dialog

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

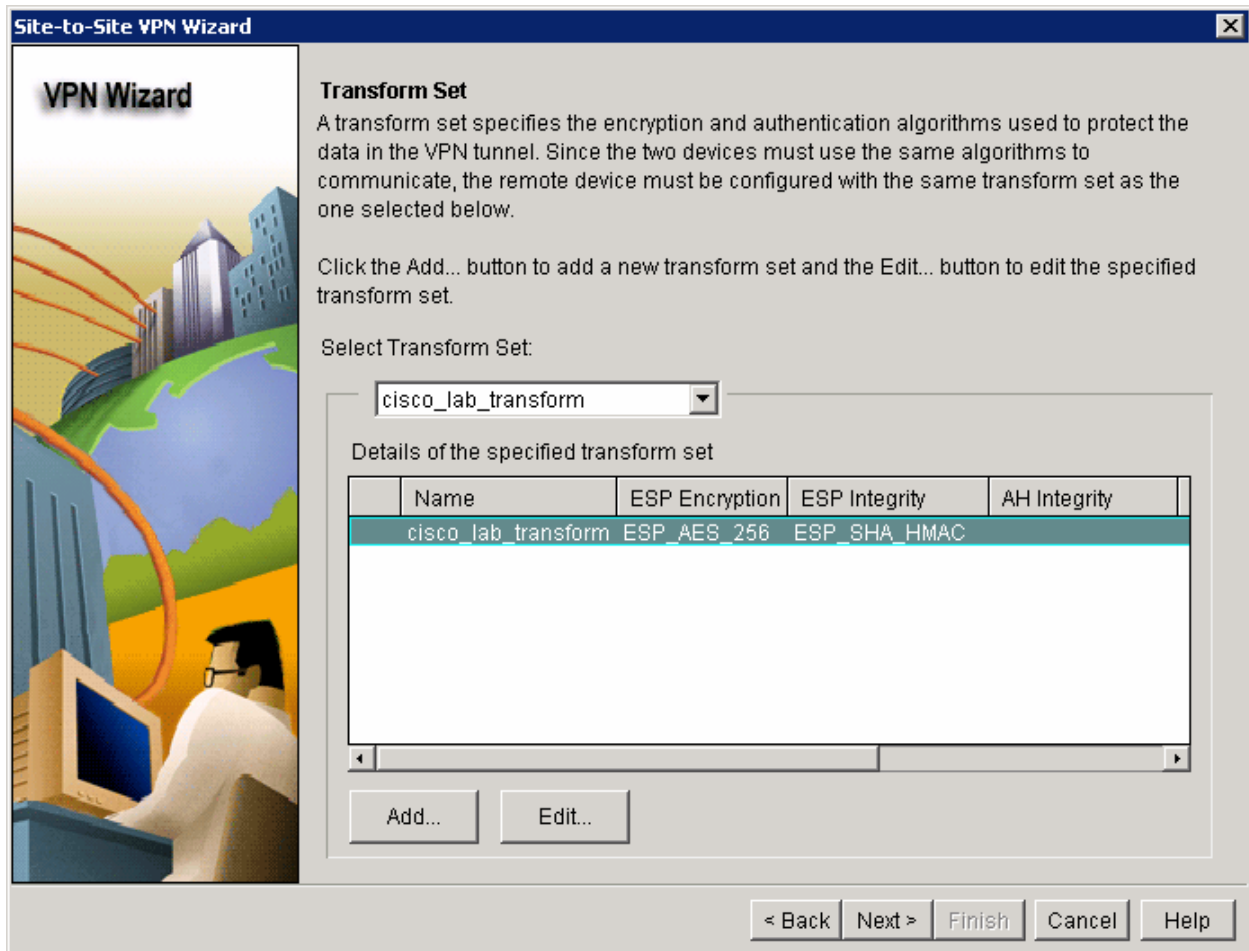


Figure 4-9: IPsec Transform Set List with Changes Applies

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.

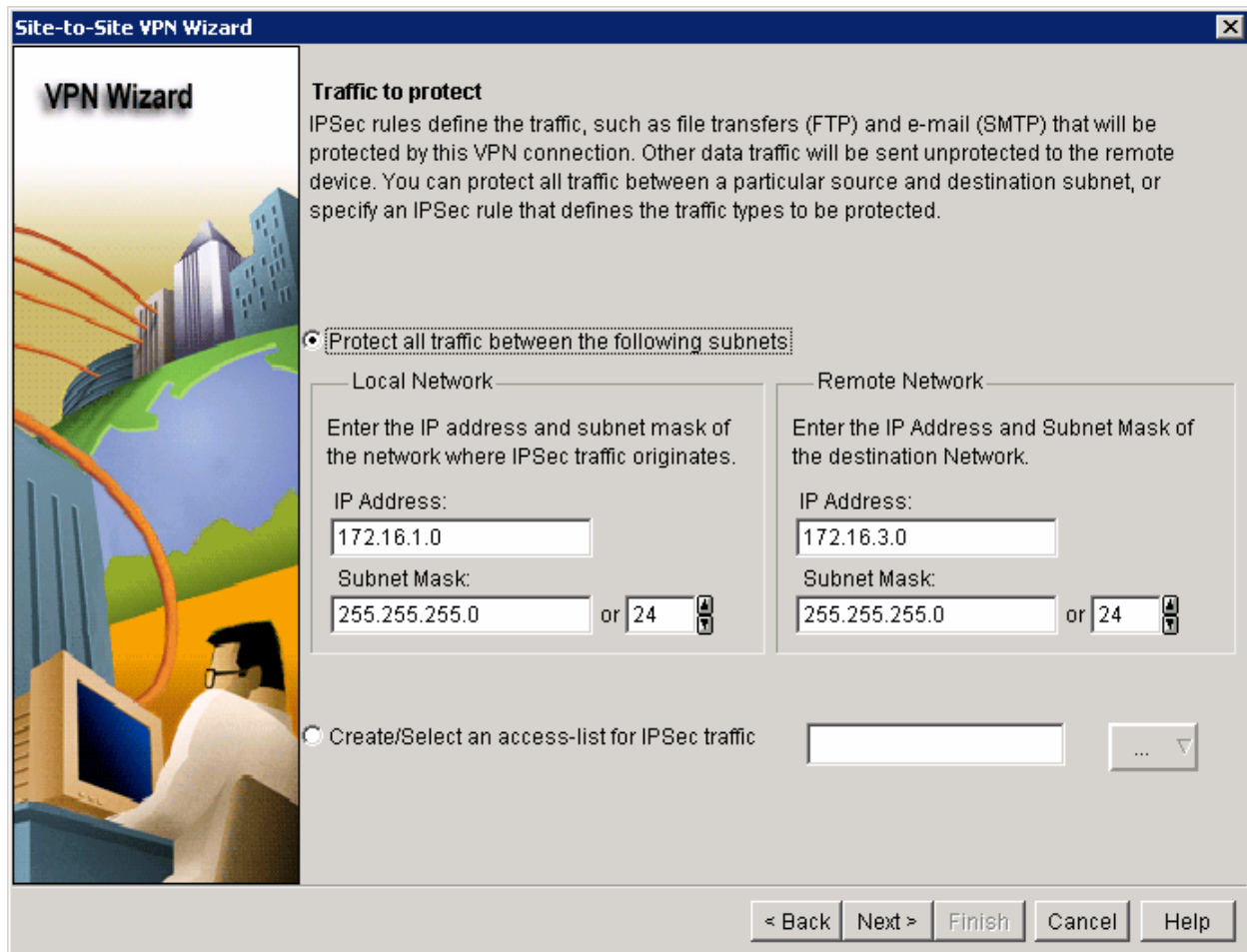


Figure 4-10: Access List Definition

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.

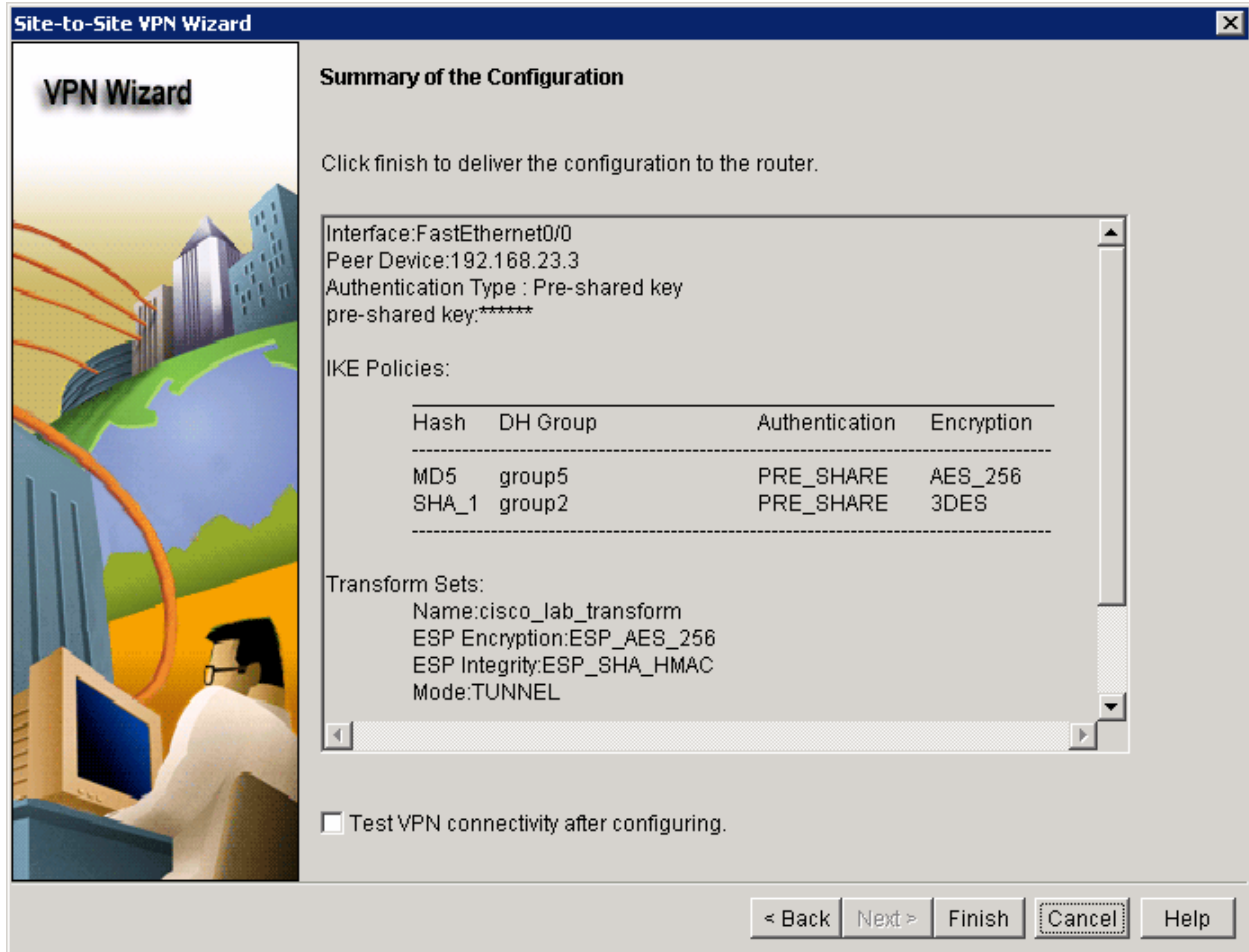


Figure 4-11: Site-to-Site VPN Configuration Summary

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.

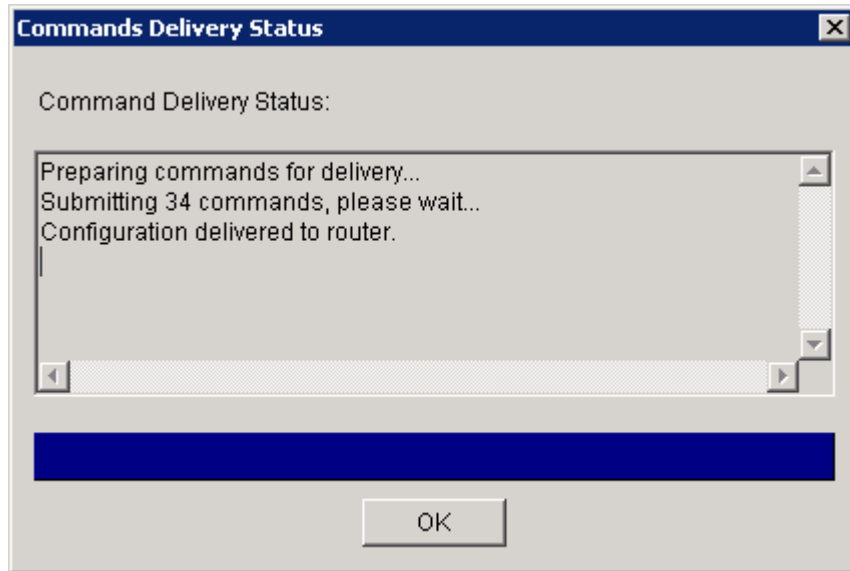


Figure 4-12: Command Delivery Progress Indicator

Step 5: Generate a Mirror Configuration for R3

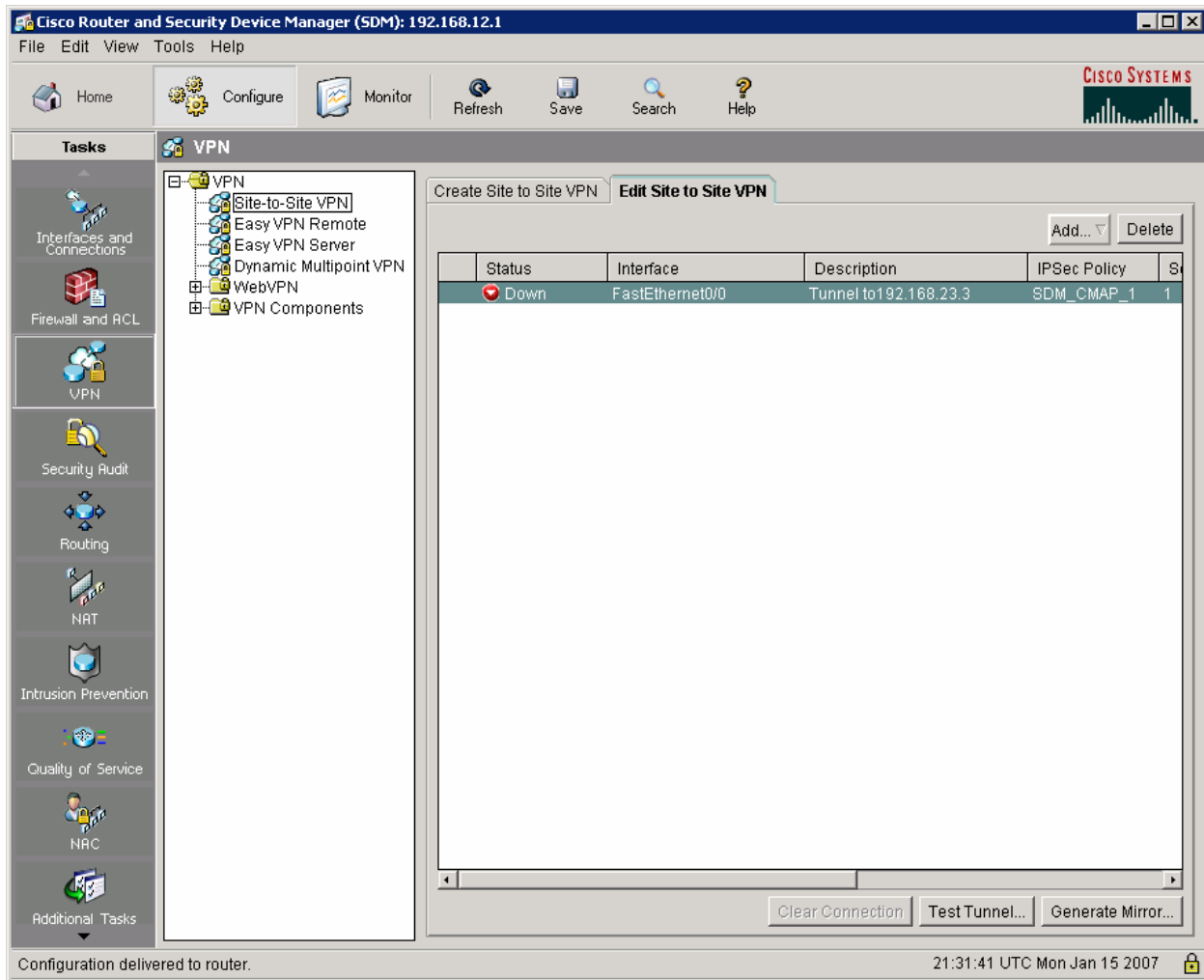


Figure 5-1: VPN Configuration Screen

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.

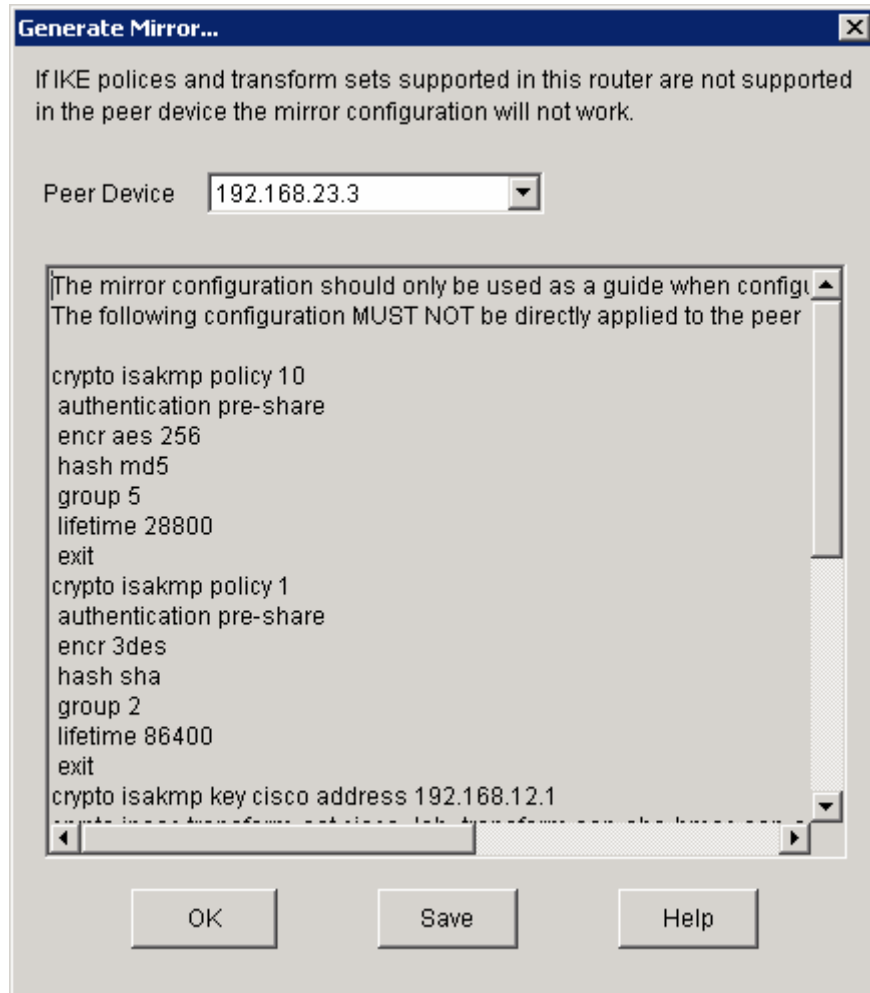


Figure 5-2: Mirror VPN Configuration

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.

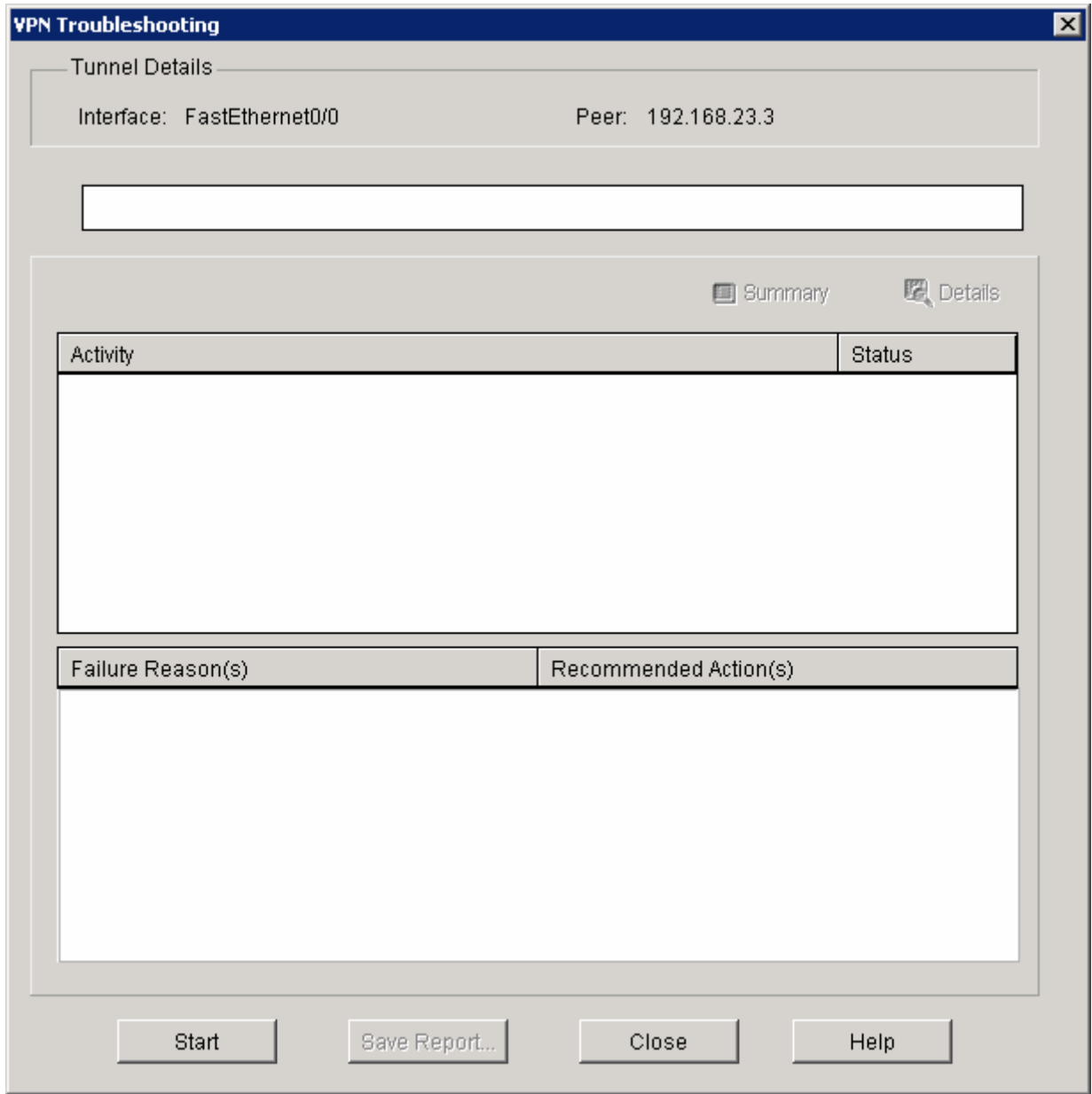


Figure 6-1: VPN Testing Window

This process may take a few moments.

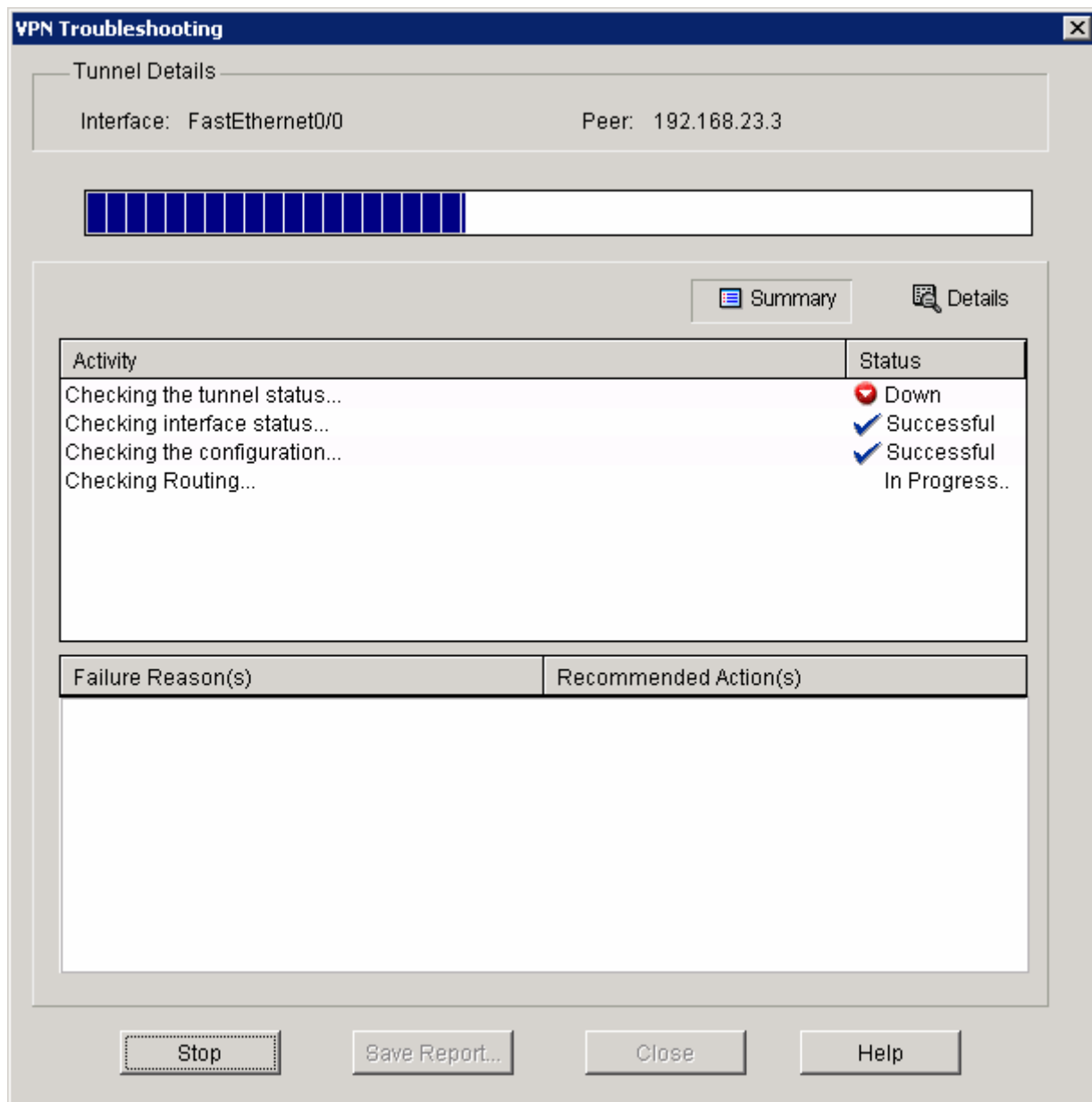


Figure 6-2: VPN Test In Progress

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

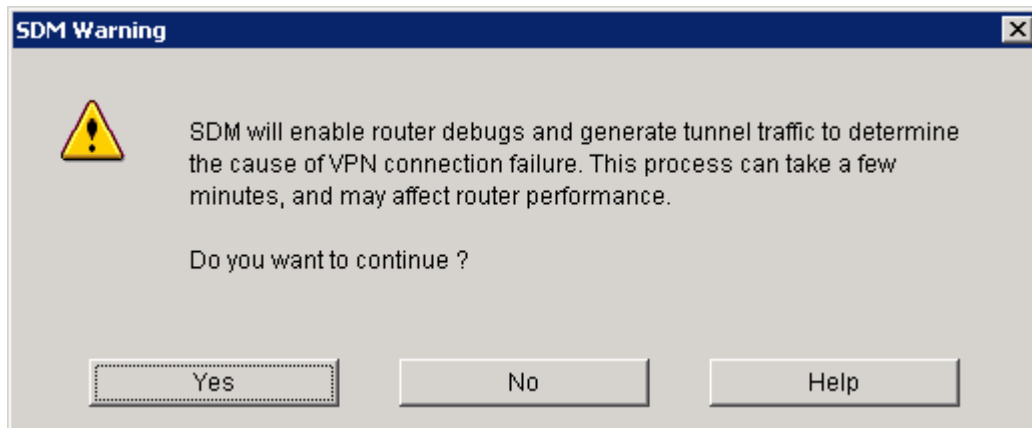


Figure 6-3: SDM Performance Warning

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

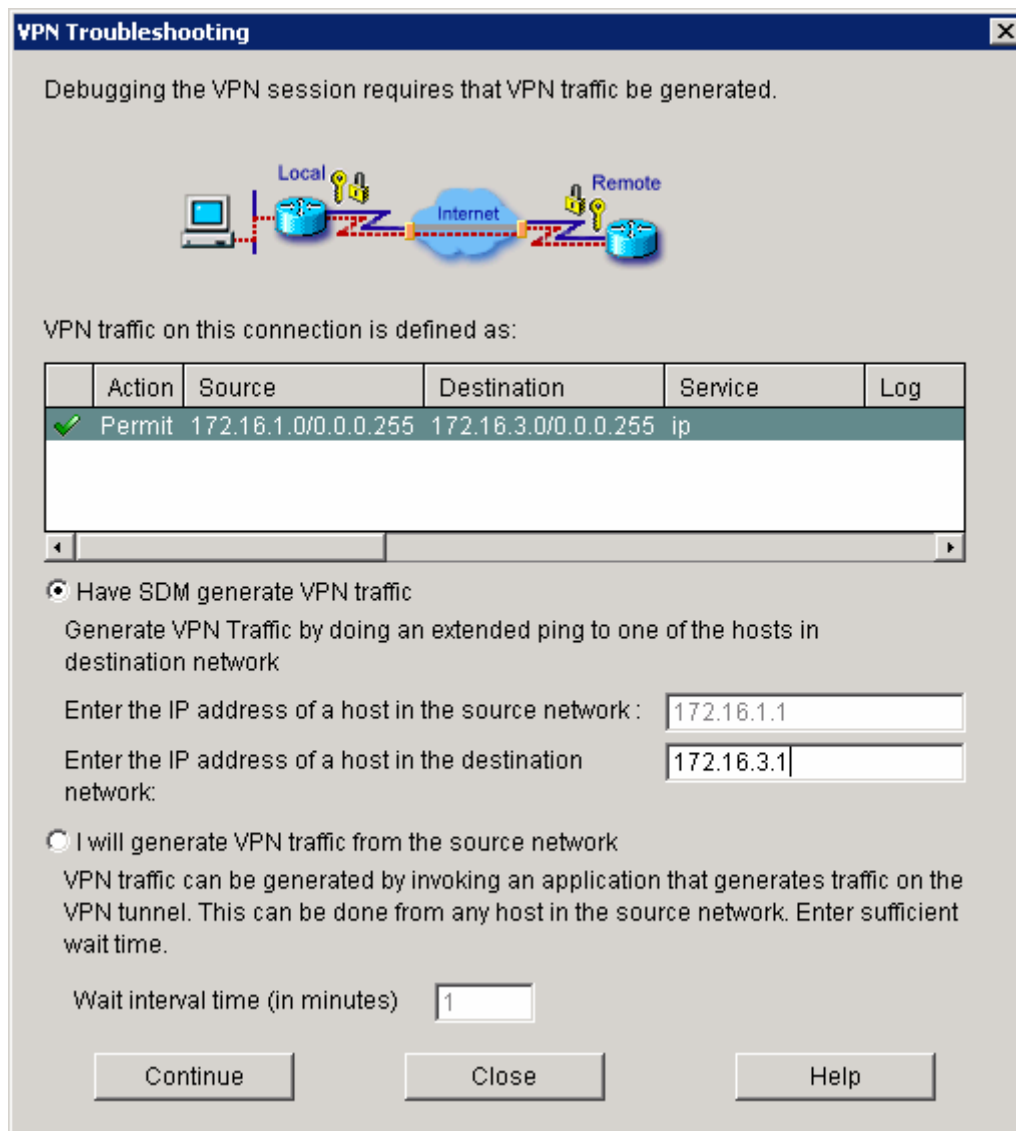


Figure 6-4: Test Traffic Generation Window

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.

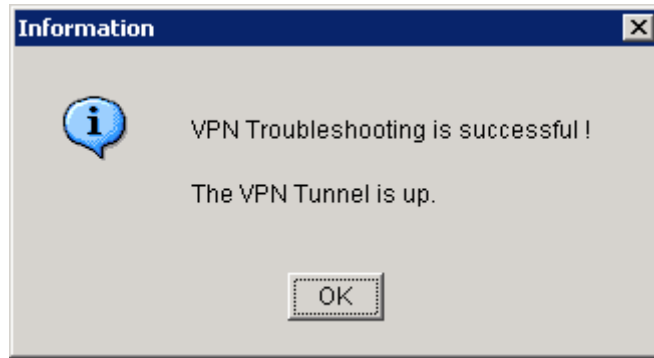


Figure 6-5: Successful VPN Test Status Window

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.

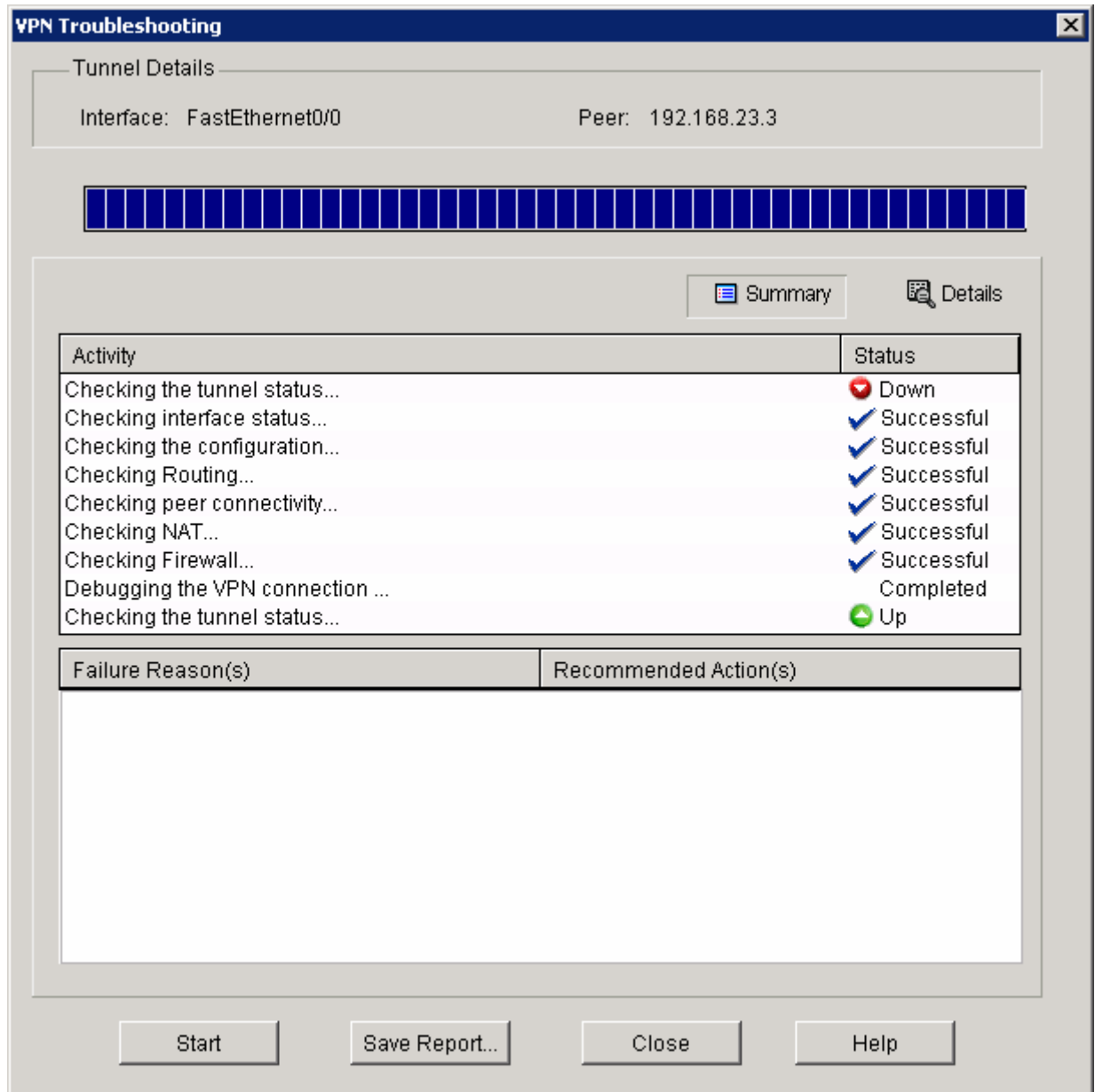


Figure 6-6: Detailed VPN Test Results

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-ISA_KMP_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-ISA_KMP_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.

The image shows a Wireshark capture of network traffic. The main packet list pane displays several packets, with packet 56 selected and highlighted in red. Packet 56 is an EIGRP Hello message from source 192.168.12.2 to destination 224.0.0.10. The packet details pane for packet 56 shows the following structure:

- 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

At the bottom of the packet details pane, the raw data is displayed in hexadecimal and ASCII:

```

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 ....
  
```

The status bar at the bottom indicates the file path and packet statistics: File: "C:\DOCU...ADMINI...LOCALS...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

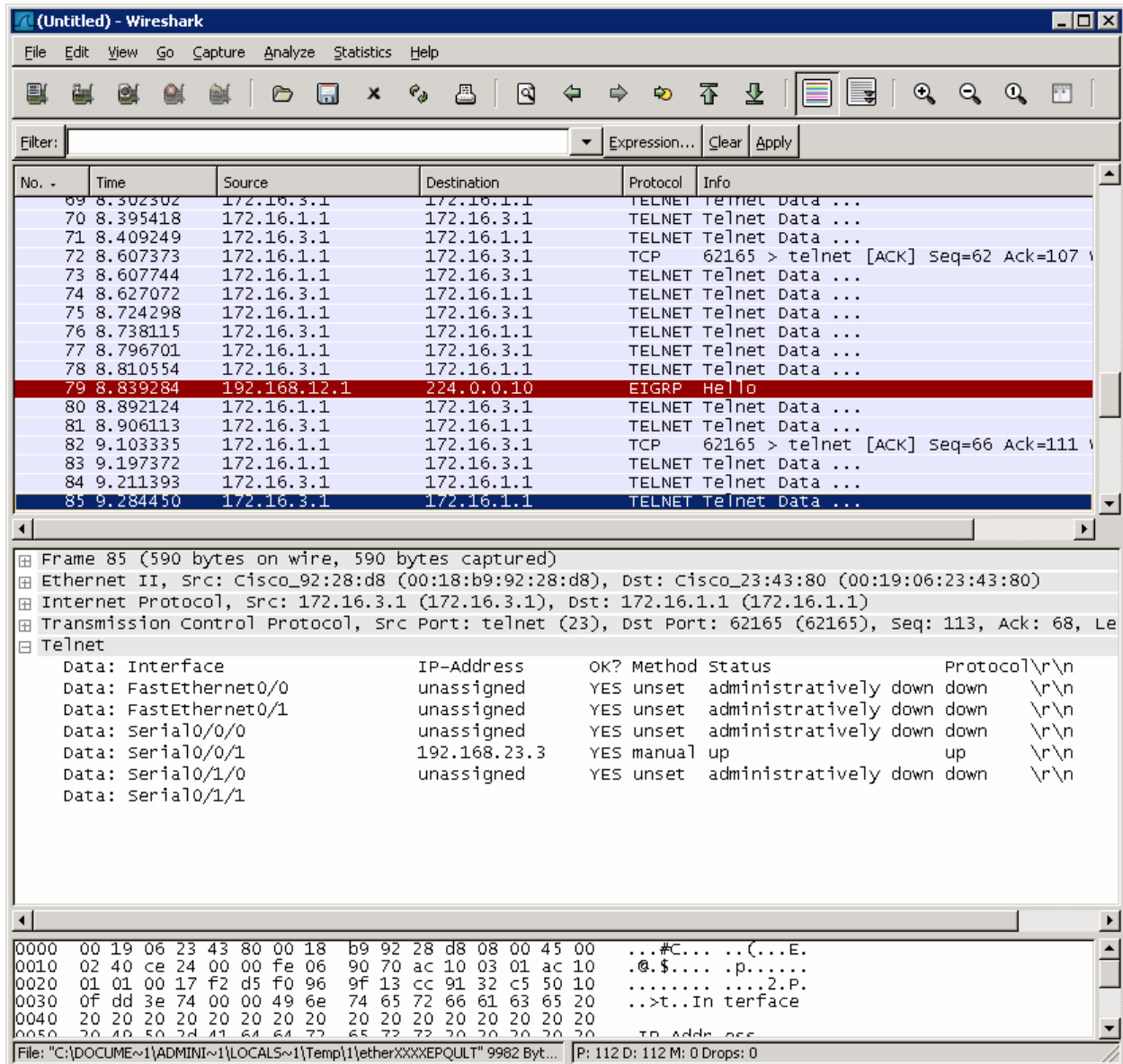
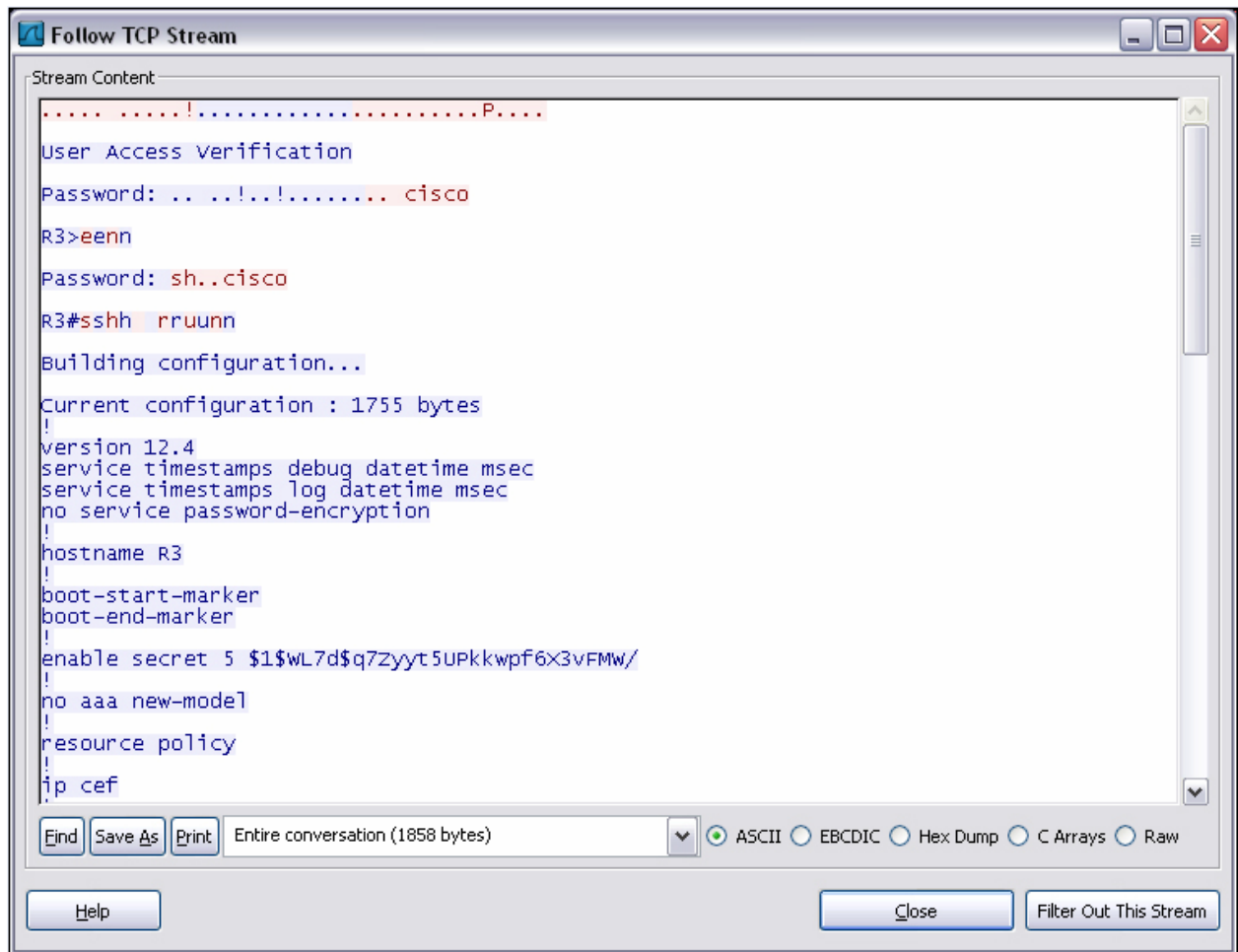


Figure 8-2: Detailed Packet Data on Return Telnet Traffic from R3

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



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-ISA_KMP_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-ISA_KMP_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.

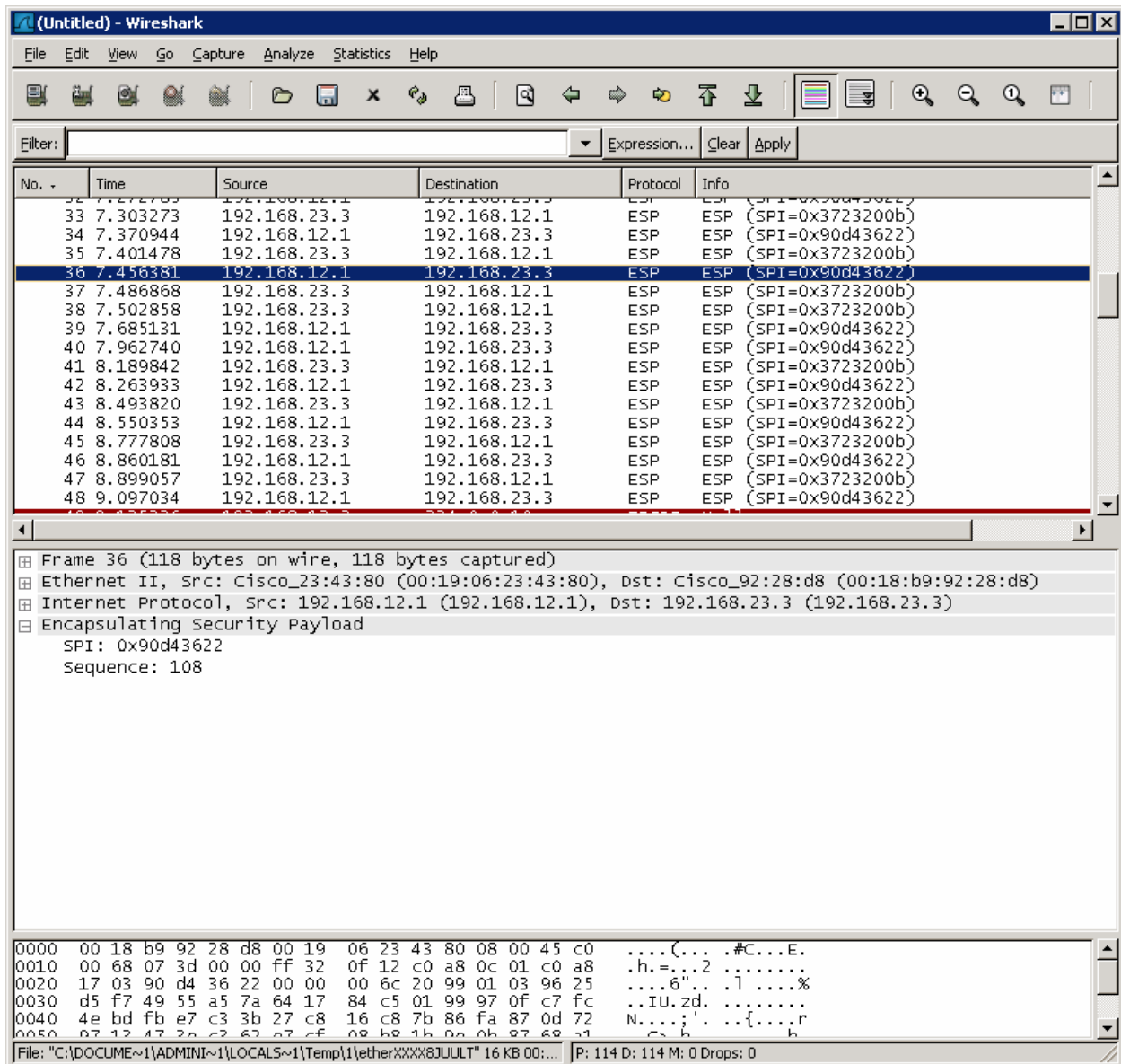


Figure 8-3: Detailed Packet Data on Encrypted Telnet String Sent From R1

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
```