SNRS

Securing Networks with Cisco Routers and Switches

Version 2.0

Lab Guide

Editorial, Production, and Web Services: 02.06.07

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

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Europe Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: +31 0 800 020 0791
Fax: +31 0 20 357 1100

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

SNRS

Lab Guide

Overview

This guide presents the instructions and other information concerning the lab activities for this course. You can find the solutions in the lab activity Answer Key.

Outline

This guide includes these activities:

- Lab 1-1: Configure Layer 2 Security
- Lab 1-2: Configure DHCP Snooping
- Lab 2-1: Configure Cisco Secure ACS as a AAA Server
- Lab 2-2: Configure 802.1x Port-Based Authentication
- Lab 3-1: Configure Cisco NFP
- Lab 4-1: Configure a Site-to-Site VPN Using Pre-Shared Keys
- Lab 4-2: Configure a Site-to-Site VPN Using PKI
- Lab 4-3: Configure a GRE Tunnel to a Remote Site
- Lab 4-4: Configure a DMVPN
- Lab 4-5: Configure a Cisco IOS SSL VPN (WebVPN)
- Lab 4-6: Configure Cisco Easy VPN Remote Access
- Lab 5-1: Configure Cisco IOS Classic Firewall
- Lab 5-2: Configure Cisco IOS Application Policy Firewall
- Lab 5-3: Configure a Cisco IOS Zone-Based Policy Firewall
- Lab 5-4: Configure Cisco IOS Firewall Authentication Proxy on a Cisco Router
- Lab 5-5: Configure a Cisco Router with Cisco IOS IPS

Lab 1-1: Configure Layer 2 Security

Complete this lab activity to practice what you learned in the related module.

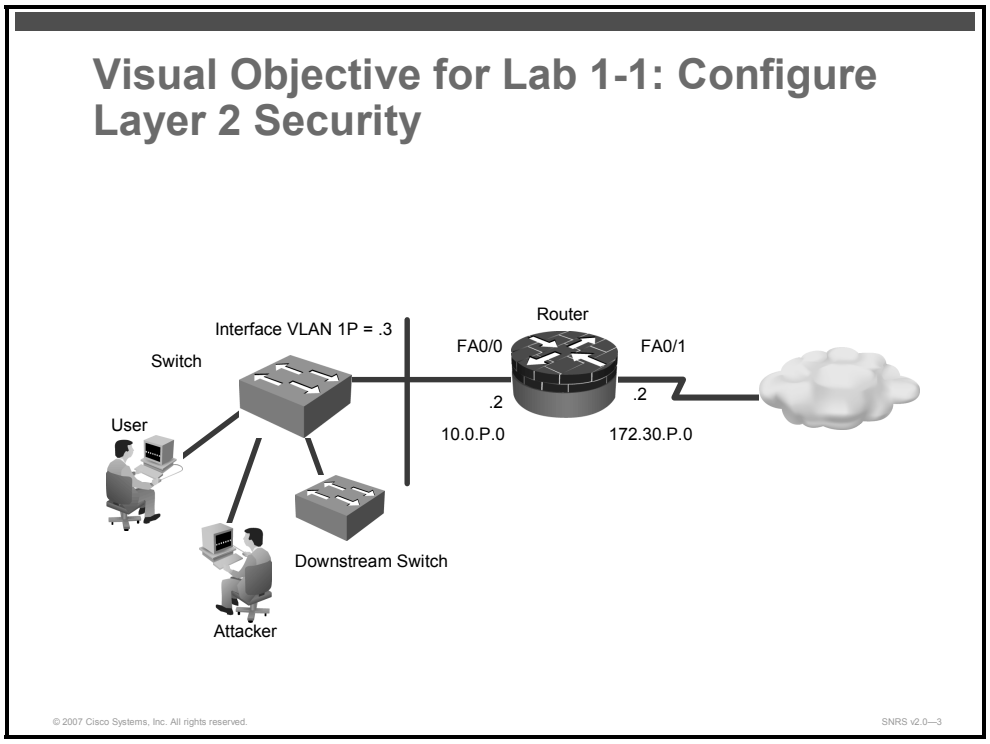
Activity Objective

In this activity, you will configure Layer 2 security on a Cisco Catalyst switch. After completing this activity, you will be able to meet these objectives:

- Mitigate a CAM table overflow attack using the appropriate Cisco IOS commands
- Mitigate a VLAN hopping attack using the appropriate Cisco IOS commands
- Prevent STP manipulation using the appropriate Cisco IOS commands
- Mitigate a MAC spoofing attack using the appropriate Cisco IOS commands
- Defend a PVLAN attack using the appropriate Cisco IOS commands

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers
- Pod switches

Command List

The table describes the commands that are used in this activity.

Layer 2 Security Commands

Command	Description
arp timeout *seconds*	This command is used to configure how long an entry remains in the ARP cache. To restore the default value, use the **no** form of this command.
show port-security [**address**] [**interface** *interface-id*]	This command is used to display the port security settings for an interface or for the switch.
switchport mode access	This command is used to configure a switch port as an access port only.
switchport port-security	This command enables port security on an interface.
switchport port-security mac-address [**sticky** | *mac-addr*]	This command is used to set a secure MAC address on an interface or use the **sticky** option to allow the switch to learn the first MAC address. Use the **no** form of this command to remove a MAC address from the list of secure MAC addresses.
switchport port-security maximum *max-addr*	This command sets the maximum number of secure MAC addresses for the interface. The range is 1 to 128; the default is 128.
switchport port-security violation {**shutdown** | **restrict** | **protect**}	This command sets the security violation mode for the interface.

Job Aids

There are no job aids for this activity.

Task 1: Mitigate a CAM Table Overflow Attack

You can mitigate a CAM table overflow attack using the **port-security** command.

Activity Procedure

Complete these steps:

Step 1 Enter interface configuration mode.

```
switch(config)# interface FastEthernet 0/2
```

Step 2 Set the port mode to access.

```
switch(config-if)# switchport mode access
```

Step 3 Enable port security on the selected interface.

```
switch(config-if)# switchport port-security
```

Step 4 Configure the maximum number of MAC addresses to one.

```
switch(config-if)# switchport port-security maximum 1
```

Note The default is one.

Step 5 Configure the action to take if there is a violation.

```
switch(config-if)# switchport port-security violation shutdown
```

Note The default is to shut down.

Step 6 Configure the MAC address for the port.

```
switch(config-if)# switchport port-security mac-address
xxxx.xxxx.xxxx
```

Or

```
switch(config-if)# switchport port-security mac-address sticky
```

Step 7 Plug a laptop into Fa0/2 and try to ping the gateway.

```
C:>ping 10.0.P.2
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show port-security <int>** command when port security is configured using the **sticky** option will look like this:

```
switch# show port-security interface FastEthernet 0/2
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address        : 0016.4111.0d49
Security Violation Count   : 0
```

- The output of the **show port-security** command when port security is configured using the **sticky** option will look like this:

```
switch# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)          (Count)
---------------------------------------------------------------------------
      Fa0/2             1            1                  0         Shutdown
---------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
```



```
Max Addresses limit in System (excluding one mac per port) : 1024
```

- The output of the **show port-security address** command should resemble the following:

```
switch# show port-security address
          Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                Ports   Remaining Age
                                                          (mins)
----    -----------       ----                -----   -------------
  11    0016.4111.0d49    SecureSticky        Fa0/2         -
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 1024
```

- The output of the **show run** command should show the following under interface Fa0/2:

```
!
interface FastEthernet0/2
 switchport access vlan 11
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0016.4111.0d49
!
```

Task 2: Mitigate a MAC Spoofing attack

You can show that, using the **port-security** command, you may also mitigate a MAC spoofing attack.

Activity Procedure

Complete these steps:

Step 1 Enter interface configuration mode.

```
switch(config)# interface FastEthernet 0/2
```

Step 2 Configure the maximum number of MAC addresses.

```
switch(config-if)# switchport port-security maximum 1
```

Step 3 Configure the action to take if there is a violation.

```
switch(config-if)# switchport port-security violation shutdown
```

Step 4 Set the length of time that an entry will stay in the ARP cache to 60 seconds.

```
switch(config-if)# arp timeout 60
```

Activity Verification

You have completed this task when you attain these results:

- You plug another PC into the port without the correct MAC address, and the port is shut down.
- The output from the **show port-security** command should be similar to this:

```
switch# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)          (Count)
-------------------------------------------------------------------------
      Fa0/2              1            1                  0         Shutdown
-------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
     Max Addresses limit in System (excluding one mac per port) : 1024
```

- The output from the **show port-security interface** command should be similar to this:

```
switch# show port-security interface fa0/2
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address        : 0050.daeb.43d4
Security Violation Count   : 1
```

- The output from the **show interface status** command should be similar to this:

```
switch# show interface status

Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1                        notconnect   1            auto   auto 10/100BaseTX
Fa0/2                        err-disabled 11         a-full  a-100 10/100BaseTX
Fa0/3                        notconnect   1            auto   auto 10/100BaseTX
Fa0/4                        notconnect   1            auto   auto 10/100BaseTX
Fa0/5                        notconnect   1            auto   auto 10/100BaseTX
```


Task 3: Mitigate a VLAN Hopping attack

You can mitigate a VLAN hopping attack by using the **switchport mode** command.

Activity Procedure

Complete these steps:

Step 1 Enter interface configuration mode.

```
switch(config)# interface FastEthernet 0/2
```

Step 2 Limit the port to access only.

```
switch(config-if)# switchport mode access
```

Activity Verification

You have completed this task when you attain these results:

- The output from the **show running-config** command shows the following:

```
!
interface FastEthernet0/2
switchport mode access
```

Task 4: Mitigate STP Manipulation

You can mitigate an STP manipulation attack using the **root guard** and **bpdu guard** commands.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
switch# configure terminal
```

Step 2 Enable BPDU guard by default on all PortFast ports on the switch.

```
switch(config)# spanning-tree portfast bpduguard default
```

Step 3 Enter interface configuration mode.

```
switch(config)# interface FastEthernet 0/3
```

Step 4 Enable the root guard feature on the interface.

```
switch(config-if)# spanning-tree guard root
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show spanning-tree** command should be similar to this:

```
witch# show spanning-tree summary totals
Switch is in pvst mode
Root bridge for: VLAN0011
```

```
EtherChannel misconfig guard is enabled
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is enabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Pathcost method used         is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
1 vlan                         0         0        0          2          2
```

Task 5: Mitigate a PVLAN Attack

You can use ACLs on a router to mitigate PVLAN attacks.

Note	You are using a router or other Layer 3 device to mitigate the PVLAN attack.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Enter interface configuration mode.

```
router(config)# ip access-list extended pvlan-attack
```

Step 3 Configure access control elements and exit.

```
router(config-ext-nacl)# deny ip 172.30.1.0 0.0.0.255
172.30.1.0 0.0.0.255
router(config-ext-nacl)# permit ip any any
router(config-ext-nacl)# exit
```

Step 4 Enter interface configuration mode.

```
router(config)# interface FastEthernet 0/0
```

Step 5 Apply the ACL to the interface.

```
router(config-if)# ip access-group pvlan-attack in
```


Activity Verification

You have completed this task when you attain these results:

- You can connect two computers on an isolated port of the same subnet (172.30.P.0) that you want to protect.
- You try to ping from one to the other.
- Your attempts should be unsuccessful.

Lab 1-2: Configure DHCP Snooping

Complete this lab activity to practice what you learned in the related module.

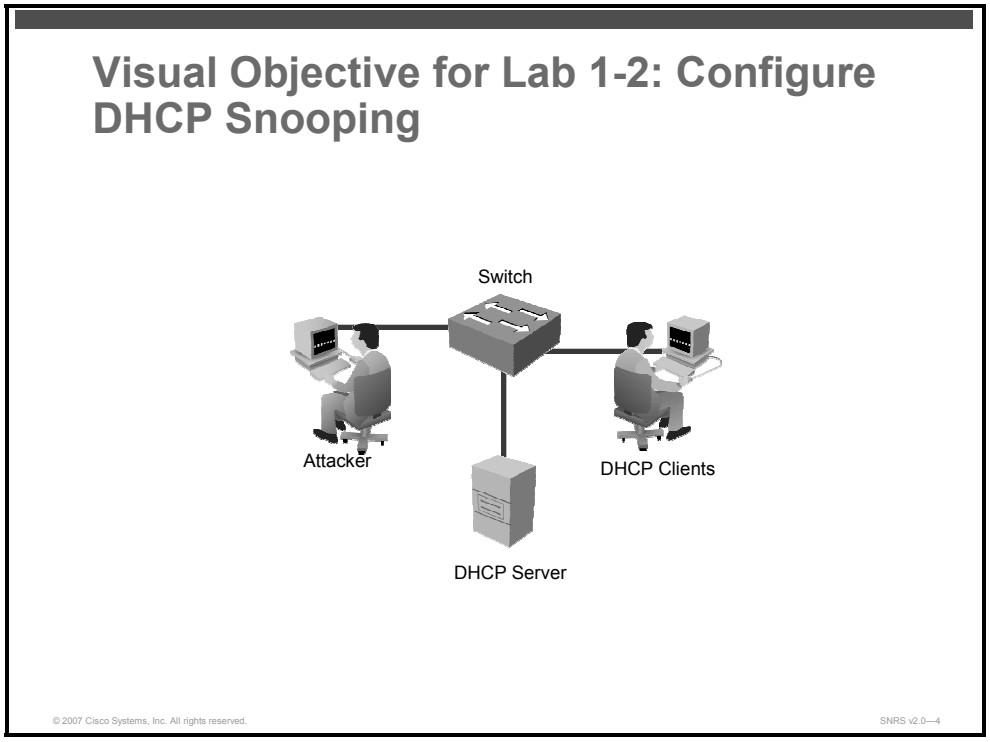
Activity Objective

In this activity, you will configure DHCP snooping on a Cisco Catalyst switch. After completing this activity, you will be able to meet these objectives:

- Enable DHCP snooping globally
- Apply DHCP snooping to a VLAN
- Configure ports as trusted or untrusted
- Verify DHCP snooping configuration

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod switches
- Pod routers

Command List

The table describes the commands that are used in this activity.

DHCP Snooping Commands

Command	Description
ip dhcp snooping	Globally enables DHCP snooping
ip dhcp snooping vlan *<vlan-id>*	Applies DHCP snooping to an active VLAN
ip dhcp snooping trust	Configures a switch port as trusted
show ip dhcp snooping	Displays information on DHCP snooping

Job Aids

There are no job aids for this activity.

Task 1: Globally Enable DHCP Snooping

In this task, you will globally enable DHCP snooping on the switch.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Globally enable DHCP snooping.

```
switch(config)# ip dhcp snooping
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show ip dhcp snooping command** should resemble the following:

```
switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
none
Insertion of option 82 is enabled
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
```

Task 2: Apply DHCP Snooping to an Active VLAN

In this task, you will apply DHCP snooping to an active VLAN.

Activity Procedure

Complete this step:

Step 1 Enable DHCP snooping on a VLAN or range of VLANs.

```
switch(config)# ip dhcp snooping vlan 11
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show ip dhcp snooping command** should resemble the following:.

```
switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
11
Insertion of option 82 is enabled
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
```

Task 3: Configure Trusted Ports

In this task, you will configure a port as trusted if it has a DHCP server connected.

Activity Procedure

Complete these steps:

Step 1 Enter interface configuration mode on the interface facing the DHCP server.

```
switch(config)# interface FastEthernet 0/2
```

Step 2 Configure the port as trusted.

```
switch(config-if)# ip dhcp snooping trust
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show ip dhcp snooping** command should resemble this:

```
switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
11
Insertion of option 82 is enabled
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
FastEthernet0/4              yes         unlimited
```


Task 4: Verify DHCP Snooping

In this task, you will verify the IP DHCP snooping configuration.

Activity Procedure

Complete these steps:

Step 1 Display the DHCP snooping configuration.

```
switch# show ip dhcp snooping
```

Step 2 Display only the dynamically configured bindings in the DHCP snooping binding database.

```
switch# show ip dhcp snooping binding
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show ip dhcp snooping** command should resemble this:

```
switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
11
Insertion of option 82 is enabled
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
FastEthernet0/4              yes         unlimited
```

Lab 2-1: Configure Cisco Secure ACS as a AAA Server

Complete this lab activity to practice what you learned in the related module.

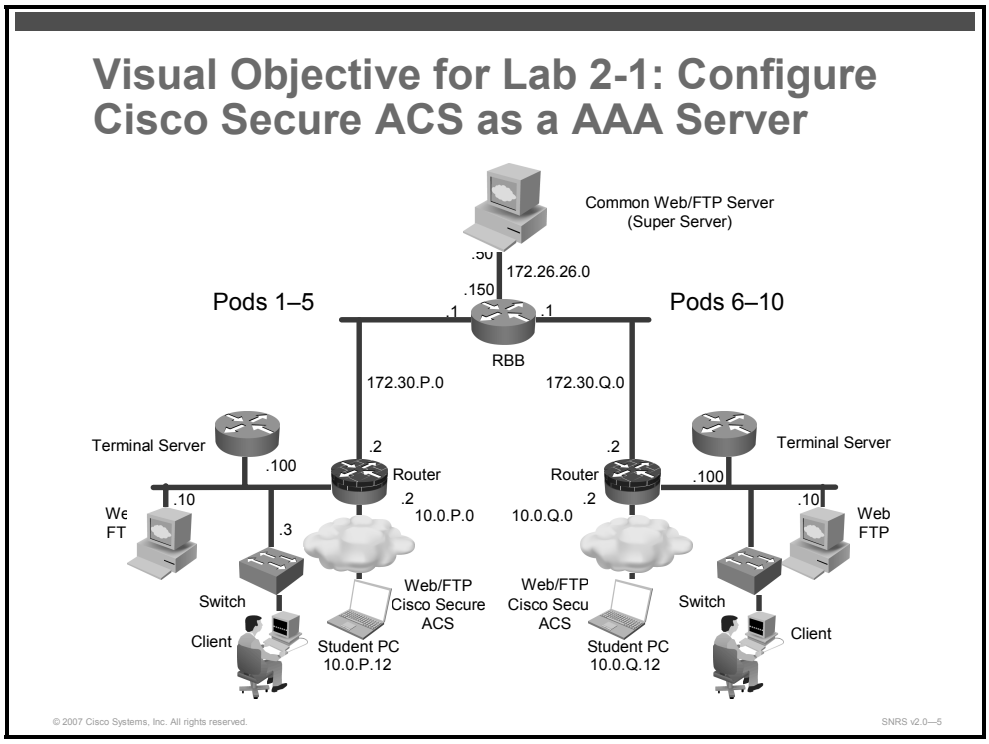
Activity Objective

In this activity, you will configure a Cisco Secure ACS for Windows to provide AAA services. After completing this activity, you will be able to meet these objectives:

- Install Cisco Secure ACS for Windows
- Add a Cisco IOS NAD as a AAA client
- Configure administrator interface settings
- Install a Cisco Secure ACS certificate
- Configure logging and reports
- Configure shared profile components
- Create a NAP for 802.1x authentication
- Define an authentication policy for a NAP
- Define an authorization policy for a NAP

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Intel-based server (laptop or desktop)
- Microsoft Windows 2000 Server with SP4
- Cisco Secure ACS 4.0
- Student laptops
- Pod devices

Command List

The table describes the commands that are used in this activity.

Cisco Secure ACS Commands

Command	Description
N/A	—

Job Aids

These job aids are available to help you complete the lab activity.

- The job aids shown in some of the tasks are available to help you complete the lab activity.

Task 1: Install Cisco Secure ACS for Windows

In this task, you will install Cisco Secure ACS 4.0 on a Microsoft Windows server machine.

Activity Procedure

Complete these steps:

Step 1 Open the **Cisco Secure ACS** folder.

Step 2 Double-click **Setup.exe**. The Cisco Secure ACS 4.0 Setup dialog box opens.

Step 3 Click **Accept** to acknowledge the terms of the Cisco Secure ACS license agreement. The Welcome window appears.

Step 4 Click **Next** in the Welcome window. The Before You Begin dialog box opens.

Step 5 Check all items listed in the Before You Begin window and click **Next**. The Choose Destination Location dialog box opens.

- End-user clients can successfully connect to AAA clients.
- This Microsoft Windows server can ping the AAA clients.
- Any Cisco IOS AAA clients are running Cisco IOS Release 11.1 or later.
- Microsoft Internet Explorer 6 SP1 or Netscape 8.0 is installed.

Step 6 Click **Next** to accept the default settings in the Choose Destination Location window. The Authentication Database Configuration dialog box opens.

Step 7 Choose **Check the Cisco Secure ACS database Only** and click **Next**. The files are installed on the server. The Advanced Options dialog box opens.

Step 8 Leave all of the Advanced Options selections unchecked at this time and click **Next**. The Active Service Monitoring dialog box opens.

Step 9 Accept the Active Service Monitoring defaults by clicking **Next**. The Cisco Secure ACS Service Initiation dialog box opens.

Step 10 Enter **cisco123** as the Cisco database encryption password. Click **Next**.

Step 11 Accept the default settings within the Cisco Secure ACS Service Initiation window by clicking **Next**. Setup then starts the Cisco Secure ACS service. The Setup Complete dialog box opens.

Step 12 Click **Finish**.

Activity Verification

You have completed this task when you attain these results:

- On the Microsoft Windows server, choose **Start > Administrative Tools > Services**. Check that all seven Cisco Secure ACS services are "Started."

Task 2: Add a Cisco IOS NAD as a AAA Client

In this task, you will configure the Cisco IOS NAD as a AAA client in the Cisco Secure ACS database.

Activity Procedure

Complete these steps:

Step 1 Click the **Network Configuration** button in the navigation bar.

Step 2 In the AAA Clients box, click **Add Entry**. The Add AAA Client window opens.

Step 3 Enter the hostname of your switch as **SwP** (where P = your pod number) in the AAA Client Hostname field.

Step 4 Enter an IP address of **10.0.P.3** (where P = your pod number) in the AAA Client IP Address field. This is the IP address of the switch (NAD) interface that will forward RADIUS packets to the Cisco Secure ACS.

Step 5 Enter a shared RADIUS key of **radiuskey** in the Key field.

Step 6 Choose **RADIUS (IETF)** from the Authenticate Using list.

Step 7 Click **Submit + Apply**.

Activity Verification

You have completed this task when you attain these results:

- You can view the new AAA client in the AAA Clients box.

Task 3: Configure Administrator Interface Settings

In this task, you will configure the Cisco Secure ACS administrator interface.

Activity Procedure

Complete these steps:

Step 1 Click the **Interface Configuration** button in the navigation bar. The Interface Configuration window opens.

Step 2 Choose **Advanced Options**. The Advanced Options window opens.

Step 3 Enable these advanced options by checking the check boxes in the Advanced Options list (uncheck any other items that are checked, for this lab only):

- **Group-Level Shared Network Access Restrictions**
- **Group-Level Network Access Restrictions**
- **Group-Level Downloadable ACLs**
- **Network Access Filtering**

Step 4 Click **Submit**.

Step 5 Choose **RADIUS (IETF)**. The RADIUS (IETF) options window opens.

Step 6 Check these items (uncheck any other items that are checked, for this lab only):

- **[027] Session-Timeout**
- **[029] Termination-Action**
- **[064] Tunnel-Type**
- **[065] Tunnel-Medium-Type**
- **[081] Tunnel-Private-Group-ID**

Step 7 Click **Submit**.

Activity Verification

You have completed this task when you attain these results:

- Review your settings by choosing **Interface Configuration > Advanced Options**.

Task 4: Add an Administrator

In this task, you will configure the Cisco Secure ACS administrator account.

Activity Procedure

Complete these steps:

Step 1 Click the **Administration Control** button in the navigation bar. The Administration Control window opens.

Step 2 Click the **Add Administrator** button. The Add Administrator window opens.

Step 3 Enter the administrator name **admin** in the Administrator Name field.

Step 4 Enter the password **cisco123** in the Password field.

Step 5 Re-enter the password **cisco123** in the Confirm Password field.

Step 6 Scroll down to the Administrator Privileges box and click **Grant All**.

Step 7 Click **Submit**.

Activity Verification

You have completed this task when you attain these results:

- Review your settings under **Administration Control**.

Task 5: Install a Cisco Secure ACS Certificate

In this task, you will install the required Cisco Secure ACS certificate.

Activity Procedure

Complete these steps:

Step 1 Click the **System Configuration** button in the navigation bar. The System Configuration window opens.

Step 2 Click **ACS Certificate Setup**. The Cisco Secure ACS Certificate Setup window opens.

Step 3 Choose **Install Cisco Secure ACS Certificate**. The Install Cisco Secure ACS Certificate window opens.

Step 4 Choose **Read Certificate from File**.

Step 5 Enter the full path to the certificate file as **c:\certs\server.cer** in the Certificate File field.

Step 6 Enter the full path to the private key file as **c:\certs\server.pvk** in the Private Key File field.

Step 7 Enter the private key password **1111** in the Private Key Password field.

Step 8 Click **Submit**. The Installed Certificate Information window opens, displaying "OK" on the Validity line. Do not restart the Cisco Secure ACS system as prompted.

Step 9 Click the **System Configuration** button in the navigation bar. The System Configuration window opens.

Step 10 Click **Cisco Secure ACS Certificate Setup**. The Cisco Secure ACS Certificate Setup window opens.

Step 11 Choose **Cisco Secure ACS Certification Authority Setup**. The Cisco Secure ACS Certification Authority Setup window opens.

Step 12 Enter the full path to the CA certificate file as **c:\certs\ca.cer** in the CA Certificate File field. A configuration change message is displayed. Do not restart Cisco Secure ACS as prompted.

Step 13 Click **Submit**.

Step 14 Click the **System Configuration** button in the navigation bar. The System Configuration window opens.

Step 15 Click **Cisco Secure ACS Certificate Setup**. The Cisco Secure ACS Certificate Setup window opens.

Step 16 Click **Edit Certificate Trust List**. The Edit Certificate Trust List window opens.

Step 17 Scroll down until you locate the Stress CA.

Step 18 Check the **Stress** check box.

Step 19 Click **Submit**.

Step 20 Choose **System Configuration > Service Control.**

Step 21 Click **Restart**. A progress bar in the lower-right corner of the window indicates the status of the restart. When the browser refreshes (blinks), this task is complete.

Activity Verification

You have completed this task when you attain these results:

- By choosing **System Configuration > Cisco Secure ACS Certificate Setup > Install Cisco Secure ACS Certificate,** you can view your certificate information.

Task 6: Configure Logging and Reports

In this task, you will configure Cisco Secure ACS service logging.

Job Aid

Use the values shown in this table to complete this task.

CSV Failed Attempts	**CSV Passed Authentications**
☑ Log to CSV Failed Attempts Report	☑ Log to CSV Passed Authentication Report
Logged Attribute	**Logged Attribute**
▪ Message-Type ▪ User-Name ▪ Group Name ▪ Caller-ID ▪ Authen-Failure-Code ▪ Author-Failure-Code ▪ Authen-Data ▪ NAS-Port ▪ NAS-IP-Address ▪ AAA Server ▪ Filter Information ▪ Access Device ▪ Network Access Profile Name ▪ Shared RAC ▪ Downloadable ACL ▪ Reason	▪ Message-Type ▪ User-Name ▪ Group Name ▪ Caller-ID ▪ NAS-Port ▪ NAS-IP-Address ▪ AAA Server ▪ Filter Information ▪ Access Device ▪ Network Access Profile Name ▪ Shared RAC ▪ Downloadable ACL ▪ Reason

Activity Procedure

Complete these steps:

Step 1 Click the **System Configuration** button in the navigation bar. The System Configuration window opens.

Step 2 Click **Service Control**.

Step 3 Scroll down to the Services Log File Configuration section and make these changes:

- Set the Level of Detail option to **Full**.
- Set the Generate New File option to **When Size Is Greater Than 2048KB**.

Step 4 Leave all other parameters at their default settings and click **Restart**. A progress bar in the lower-right corner of the window indicates the status of the restart. When the browser refreshes (blinks), this task is complete.

Step 5 Click the **System Configuration** button in the navigation bar. The System Configuration window opens.

Step 6 Click **Logging**. The Logging Configuration window opens.

Step 7 Click **CSV Passed Authentications**. The CSV Passed Authentications File Configuration window opens.

Step 8 Locate the **Enable Logging** area and check the **Log to CSV Passed Authentications Report** check box.

Step 9 Locate the **Select Columns to Log** area and click the **Right Arrow** button to move the NAC-specific attributes listed in the job aid for this task to the Logged Attributes column.

Step 10 Click **Submit**.

Step 11 Click **CSV Failed Attempts**.

Step 12 Repeat Step 9 for CSV Failed Attempts.

Step 13 Click **Submit**. The system returns you to the Logging Configuration window. The CSV Passed Authentications and CSV Failed Attempts logging configuration should now show a check (enabled) in the Use column.

Activity Verification

You have completed this task when you attain these results:

- Review your settings by choosing **System Configuration > Logging**.

Task 7: Configure Global Authentication

In this task, you will enable EAP for 802.1x authentication and set the various EAP session timeout values.

Note You usually enable all protocols globally so that you can choose a specific protocol from the protocols later on during the NAP configuration process. You can choose to enable one or all protocols here. Whatever you select here, will be available for selection when configuring a NAP.

Job Aid

Use the values shown in this table to complete this task.

EAP Configuration
PEAP
☑ Allow EAP-MSCHAPv2
☑ Allow EAP-GTC
☐ Allow Posture Validation
Cisco client initial message: **<empty>**
PEAP session timeout (minutes): **120**
Enable Fast Reconnect: ☑
EAP-FAST
EAP-FAST Configuration (see below)
EAP-TLS
☑ Allow EAP-TLS
Choose one or more of the following options:
☑ Certificate SAN comparison
☑ Certificate CN comparison
☑ Certificate Binary comparison
EAP-TLS Session Timeout (minutes): **120**
LEAP
☑ Allow LEAP (For Aironet only)
EAP-MD5
☑ Allow EAP-MD5
AP EAP request timeout (seconds): **20**

MS-CHAP Configuration
☑ Allow MS-CHAP Version 1 Authentication
☑ Allow MS-CHAP Version 2 Authentication

EAP-FAST Settings
EAP-FAST
☑ Allow EAP-FAST
Active master key TTL: **1 month**
Retired master key TTL: **3 month**
Tunnel PAC TTL: **1 week**
Client Initial Message: <empty>
Authority ID Info: **cisco**
☑ Allow anonymous in-band PAC provisioning
☑ Allow authenticated in-band PAC provisioning
☑ Accept client on authenticated provisioning
☐ Require client certificate for provisioning
☐ Allow Machine Authentication
Machine PAC TTL **1 week**
☑ Allow Stateless Session Resume
Authorization PAC TTL **1 hour**

Allow inner methods
- ☑ EAP-GTC
- ☑ EAP-MSCHAPv2
- ☑ EAP-TLS

Choose one or more of the following EAP-TLS comparison methods:
- ☑ Certificate SAN comparison
- ☑ Certificate CN comparison
- ☑ Certificate binary comparison

EAP-TLS session timeout (minutes): 120

☑ EAP-FAST master server

Actual EAP-FAST server status: **Master**

Note	You will not be authenticating to an external Active Directory server, so machine authentication is not enabled.

It is recommended that you enable all protocols globally. You will be able to configure specific protocols for specific NAPs later.

Activity Procedure

Complete these steps:

Step 1 Click the **System Configuration** button in the navigation bar. The System Configuration window opens.

Step 2 Choose **Global Authentication Setup**. The Global Authentication Setup window opens.

Step 3 Locate the EAP configuration sections.

Step 4 Configure the settings in accordance with the job aid for this task.

Step 5 Set the EAP session timeout values in accordance with the job aid.

Step 6 Click **Submit + Restart**.

Activity Verification

You have completed this task when you attain these results:

- Review your settings by choosing **System Configuration > Global Authentication Setup**.

Task 8: Create Groups and Users

In this task, you will configure Cisco Secure ACS groups and users to support 802.1x authentication.

Job Aid

Use the values shown in this table to complete this task.

Group	Name	Description
1	Corporate	Corporate users
2	Engineering	Engineering users
3	Guests	Guest users

Create Groups

This procedure describes how to create the groups for use with 802.1x.

Activity Procedure

Complete these steps:

Step 1 Click the **Group Setup** button in the navigation bar.

Step 2 Choose group number **1** from the Group list.

Step 3 Click **Rename Group**. Enter the group name **Corporate** in the Group field to replace the existing name.

Step 4 Click **Submit.**

Step 5 Repeat Step 2 through Step 4 to create the Engineering and Guest groups.

Create Users

This procedure describes how to create the usernames for use with 802.1x.

Job Aid

Use the values shown in this table to complete this task.

Username	Group
user1	Corporate
eng1	Engineering
guest1	Guest

Activity Procedure

Complete these steps:

Step 1 Click the **User Setup** button in the navigation bar. The User Setup window opens.

Step 2 Enter the new username **user1** in the User field.

Step 3 Click **Add/Edit**. The User: User1 (New User) window opens.

Step 4 Use the scroll bar to locate the User Setup section.

Step 5 Enter the password **cisco123** in the Password field.

Step 6 Re-enter the password **cisco123** in the Confirm Password field.

Step 7 Use the scroll bar to locate the Group to Which the User Is Assigned section.

Step 8 Choose the **Corporate** group from the list.

Step 9 Click **Submit**.

Step 10 Repeat Step 1 through Step 9 for the rest of the table.

Activity Verification

You have completed this task when you attain these results:

- Review your users and groups under **User Setup** and **Group Setup**.

Task 9: (Optional) Create a NAF

Sometimes, it is useful to filter devices by location or some other criteria. In this task, you will create a NAP to group your devices into a location.

Activity Procedure

Complete these steps:

Step 1 Click the **Shared Profile Components** button in the navigation bar. The Shared Profile Components window opens.

Step 2 Choose **Network Access Filtering**. The Network Access Filtering window opens.

Step 3 Click **Add**. The Network Access Filtering edit window opens.

Step 4 Enter the name **HQ** in the Name field.

Step 5 If you enabled NDGs, (Not Assigned) should appear in the Network Device Groups section. Click **(Not Assigned)**. Your AAA client should appear in the Network Devices section.

Step 6 Locate the Network Devices section and click the **Right Arrow** button to move your SwP (where P = your pod number) to the Selected Items column.

Step 7 Click **Submit + Restart**. The new NAC NAF is listed in the Network Access Filtering Name list.

Activity Verification

You have completed this task when you attain these results:

- The new HQ NAF is listed in the Network Access Filtering Name list.

Task 10: Define RADIUS Authorization Components

In this task, you will configure RADIUS attributes that will be downloaded and applied to the switch upon successful network authorizations.

Job Aid

Use the values shown in this table to complete this task.

RAC Name	Vendor	Assigned Attributes	Value
Corporate_802.1x_RAC	IETF	Session-Timeout (27)	3600
	IETF	Termination-Action (29)	RADIUS-Request (1)
	IETF	Tunnel-Type (64)	[T1] VLAN (13)
	IETF	Tunnel-Medium-Type (65)	[T1] 802 (6)
	IETF	Tunnel-Private-Group-ID (81)	[T1] corporate
Engineering_802.1x_RAC	IETF	Session-Timeout (27)	3600
	IETF	Termination-Action (29)	RADIUS-Request (1)
	IETF	Tunnel-Type (64)	[T1] VLAN (13)
	IETF	Tunnel-Medium-Type (65)	[T1] 802 (6)
	IETF	Tunnel-Private-Group-ID (81)	[T1] engineering
Guest_802.1x_RAC	IETF	Session-Timeout (27)	3600
	IETF	Termination-Action (29)	RADIUS-Request (1)
	IETF	Tunnel-Type (64)	[T1] VLAN (13)
	IETF	Tunnel-Medium-Type (65)	[T1] 802 (6)
	IETF	Tunnel-Private-Group-ID (81)	[T1] guest

Activity Procedure

Complete these steps:

Step 1 Click the **Shared Profile Components** button in the navigation bar. The Shared Profile Components window opens..

Step 2 Choose **RADIUS Authorization Components**. The RAC window opens.

Step 3 Click the **Add** button for each new RAC. Each RAC may contain one or more vendor RADIUS attributes, including Cisco IOS/PIX 6.0, IETF, and Ascend.

Step 4 Click the **Add** button next to whichever attribute you want to add in the Add New Attribute section. You may add specific attributes for Cisco IOS/PIX 6.0, IETF, and Ascend if you configured the Interface settings correctly as per Task 3.

Step 5 Use the table in the job aid for this step to create the appropriate RACs.

Step 6 Click **Submit**.

Step 7 Restart services by choosing **System Configuration** > **Service Control** > **Restart**.

Activity Verification

You have completed this task when you attain these results:

- The RACs that you created should appear in the RADIUS Authorization Components table.

Task 11: Create a NAP for Layer 2-802.1x Authentication (IBNS)

In this task, you will configure a NAP. There are actually three components to a NAP, two of which are used in this lab. Those two are authentication and authorization. The third, posture validation, is used when implementing Cisco NAC.

Activity Procedure

Complete these steps:

Step 1 Click the **Network Access Profiles** button in the navigation bar. The Network Access Profiles configuration window opens.

Step 2 Click **Add Template Profile**. The Create Profile from Template window appears.

Step 3 Enter the name **L2-802.1x** for this NAP.

Step 4 Choose **Microsoft IEEE 802.1x** from the Template drop-down menu.

Step 5 Check the **Active** check box.

Step 6 Click **Submit**. The prompt reads "The current configuration has been changed. Restart Cisco Secure ACS in 'System Configuration: Service Control' to adopt the new settings."

Step 7 Check the **Deny Access When No Profile Matches** check box.

Step 8 Click **Apply and Restart**.

Step 9 Click your **L2-802.1x** profile in the Network Access Profiles window. Choose **HQ** from the Network Access Filter section. You can also leave it as (Any).

Step 10 Click **Submit**.

Step 11 Click **Apply and Restart**.

Activity Verification

You have completed this task when you attain these results:

- Click the Network Access Profiles button in the navigation bar. The L2-802.1x profile should be listed.

Task 12: Define an Authentication Policy for a NAP

In this task, you will define an authentication policy for the 802.1x NAP.

Activity Procedure

Complete these steps:

Step 1 Click the **Network Access Profiles** button in the navigation bar. The Network Access Profiles configuration window opens.

Step 2 Click **Authentication** in your L2-802.1x profile.

Step 3 Choose **Allow MD-5**.

Step 4 Under Credential Validation Databases, choose **ACS Internal Database** and click the **Right Arrow** button to move it to the Selected Databases column.

Step 5 Click **Apply + Restart**.

Activity Verification

You have completed this task when you attain these results:

- Review your configuration by choosing **Network Access Profiles > L2-802.1x Authentication**.

Task 13: Define an Authorization Policy for a NAP

In this task, you will define an authorization policy for the 802.1x NAP.

Job Aid

Use the values shown in this table to complete this task.

User Groups	Assessment Result	Shared RAC	Downloadable ACL
Corporate	Any	Corporate_802.1x_RAC	
Engineering	Any	Engineering_802.1x_RAC	
Guest	Any	Guest_802.1x_RAC	
If a condition is not defined or there is no matched condition		Guest_802.1x_RAC	

Activity Procedure

Complete these steps:

Step 1 Click the **Network Access Profiles** button in the navigation bar. The Network Access Profiles configuration window opens.

Step 2 Click **Authorization** in your L2-802.1x profile.

Step 3 Click **Add Rule** and use the table to configure your authorization rules.

Step 4 Uncheck the **Include RADIUS Attributes from Group Records** and **Include RADIUS Attributes from User Records check boxes**.

Step 5 Click **Submit**.

Step 6 Click **Apply and Restart**.

Activity Verification

You have completed this task when you attain these results:

- Review your settings by choosing **Network Access Profiles > L2-802.1x Authorization**.

Task 14: Configure the Unknown User Policy

In this task, you will create an unknown user policy.

Activity Procedure

Complete these steps:

Step 1 Click the **External User Databases** button in the navigation bar. The External User Databases window opens.

Step 2 Choose **Unknown User Policy**. The Configure Unknown User Policy window opens.

Step 3 Select the **Fail the Attempt** radio button.

Step 4 Click **Submit**.

Step 5 Click the **System Configuration** button in the navigation bar.

Step 6 Choose **Service Control**.

Step 7 Click **Restart**.

Activity Verification

You have completed this task when you attain these results:

- Review your settings by choosing **External User Databases > Unknown User Policy**.

Lab 2-2: Configure 802.1x Port-Based Authentication

Complete this lab activity to practice what you learned in the related module.

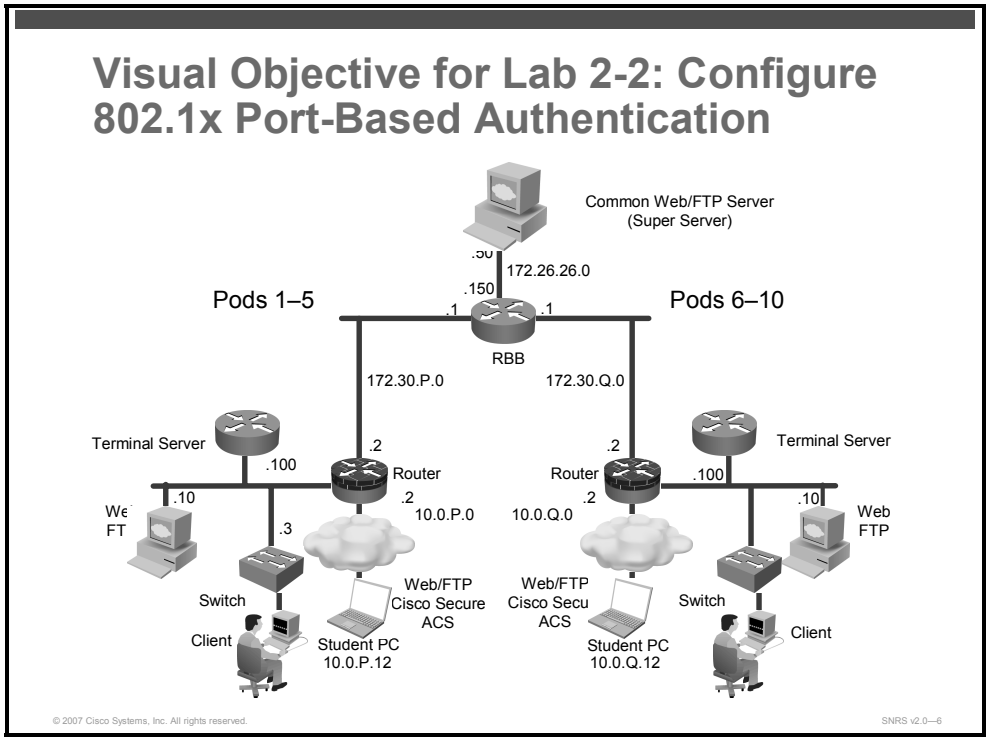
Activity Objective

In this activity, you will configure 802.1x port-based authentication on a Cisco Catalyst 2950 Series Switch. After completing this activity, you will be able to meet these objectives:

- Configure clients for dynamic addressing
- Create VLANs for segmentation according to a security policy
- Create DHCP pools for clients
- Configure the AAA service on a Cisco Catalyst switch
- Configure a port for 802.1x authentication with VLAN assignment
- Enable periodic reauthentication
- Configure 802.1x on a port with a guest VLAN
- Configure 802.1x on a port with a restricted VLAN
- Manually reauthenticate a client connected to a port
- Display 802.1x statistics and status

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops for Cisco Secure ACS
- Cisco Secure ACS 4.0.1
- Client laptops with 802.1x supplicant
- Pod switch

Command List

The table describes the commands that are used in this activity.

Switch IBNS Commands

Command	Description
aaa authentication dot1x default group radius	Creates an IEEE 802.1x authentication method list
aaa authorization network default group radius	Configures the switch for user RADIUS authorization for all network-related service requests, such as VLAN assignment
aaa accounting dot1x default start-stop group radius	Enables AAA accounting and creates method lists defining specific accounting methods on a per-line or per-interface basis for IEEE 802.1x sessions; sends a start accounting notice at the beginning of a process and a stop accounting notice at the end of a process
radius-server host *ip-address*	Specifies the IP address of a RADIUS server host
radius-server key *key*	Specifies the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon
ip radius source-interface *interface*	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets
ip dhcp pool *name*	Configures a DHCP address pool on a DHCP server and enters DHCP pool configuration mode
network *address netmask*	Configures the subnet number and subnet mask for a DHCP address pool on a Cisco IOS DHCP server
default-router *ip_address*	Defines a default router for DHCP clients
ip dhcp excluded-address *low-address* [*high-address*]	Specifies the IP addresses that a Cisco IOS DHCP server should not assign to DHCP clients
dot1x system-auth-control	Enables IEEE 802.1x authentication globally on the switch
dot1x guest-vlan supplicant	Allows clients to be put into a guest VLAN if they have an 802.1x supplicant but still fail authentication
dot1x port-control auto	Enables manual control of the authorization state of the port and causes the port to change to the authorized or unauthorized state based on the IEEE 802.1x authentication exchange between the switch and the client
dot1x timeout reauth-period server	Sets the number of seconds between reauthentication attempts The **server** keyword sets the number of seconds as the value of the session-timeout RADIUS attribute (attribute 27).
dot1x reauthentication	Enables periodic reauthentication of the client
dot1x guest-vlan *vlan-id*	Specifies an active VLAN as an IEEE 802.1x guest VLAN
dot1x host-mode multi-host	Allow multiple hosts (clients) on an IEEE 802.1x-authorized port
dot1x auth-fail vlan *vlan-id*	Specifies an active VLAN as an IEEE 802.1x restricted VLAN
show dot1x [**all** | **interface**]	Shows details for an identity profile
show interface status	Displays information about the status of an interface

Job Aids

These job aids are available to help you complete the lab activity.

- Job aids may be included in the tasks.

Task 1: Configure Client Addressing

In this task, you will configure a client for dynamic addressing. Make sure that the client is plugged into interface Fa0/1 on the pod switch.

Activity Procedure

Complete these steps on the client:

Step 1 On the PC, under the Authentication tab of Local Area Network Connection Properties, check the following:

- Ensure that the Enable Network Access Control Using IEEE 802.1x check box is checked.
- Ensure that the EAP type is MD5-Challenge.

Step 2 Right-click **My Network Places**.

Step 3 Click **Properties**. The Network Connections window opens.

Step 4 Right-click **Local Area Connection**.

Step 5 Click **Properties**. The Local Area Connection Properties window opens.

Step 6 In the This Connection Uses the Following Items window, choose **Internet Protocol (TCP/IP)**.

Step 7 Click **Properties**.

Step 8 Click the **Obtain an IP Address Automatically** radio button and click **OK**.

Step 9 Click **OK**.

Activity Verification

You have completed this task when you attain these results:

- **Obtain an IP Address Automatically** is checked when you review your TCP/IP properties.

Task 2: Create VLANs on the Switch

In this task, you will create VLANs to assign to different clients according to their identity.

Job Aid

Use the values shown in this table to complete this task.

VLAN	Name
20	guest
30	corporate
40	engineering
50	restricted
90	unauthenticated

Activity Procedure

Complete these steps:

Step 1 Create the VLAN named "guest" using the **vlan** command.

```
switch(config)# vlan 20
switch(config-VLAN)# name guest
switch(config-VLAN)# exit
```

Step 2 Repeat Step 1 and Step 2 for the rest of the VLANs.

Activity Verification

You have completed this task when you attain these results:

- The output of the **show vlan** command should resemble this:

```
switch# show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
-
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Gi0/1
                                                Gi0/2

20   guest                            active
30   corporate                        active
40   engineering                      active
50   restricted                       active
90   unauthenticated                  active
```

```
101  network_devices                  active    Fa0/24
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

Task 3: (Optional) Create DHCP Pools on the Switch or Router

In this task, you will create and configure DHCP pools for addressing clients after they are authenticated or put into the “guest” or “restricted” VLANs.

Job Aid

Use the values shown in this table to complete this task.

Name	Network	Default Router	Excluded Address
guest	10.0.20.0/24	10.0.20.2	10.0.20.1 to 10.0.20.5
corporate	10.0.30.0/24	10.0..30.2	10.0.30.2 to 10.0..30.5
engineering	10.0.40.0/24	10.0.40.2	10.0.40.2 to 10.0.40.5
restricted	10.0.50.0/24	10.0.50.2	10.0.50.2 to 10.0.50.5
unauthenticated	10.0.90.0/24	10.0.90.2	10.0.90.2 to 10.0.90.5

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
switch# configure terminal
```

Step 2 Create a DHCP pool for “guest” clients.

```
switch(config)# ip dhcp pool guest
```

Step 3 Define the subnet for this pool.

```
switch(dhcp-config)# network 10.0.20.0 255.255.255.0
```

Step 4 Define the default gateway for DHCP clients on this subnet.

```
switch(dhcp-config)# default-router 10.0.20.2
```

Step 5 Return to global configuration mode.

```
switch(dhcp-config)# exit
```

Step 6 Exclude the router interface address from the DHCP pools.

```
switch(config)# ip dhcp excluded-address 10.0.20.1 10.0.20.5
```

Step 7 Repeat Step 2 through Step 6 for the rest of the DHCP pools.

Activity Verification

You have completed this task when you attain these results:

- The output of the **show running-config** command should resemble the following:

```
switch# show running-config
!
ip dhcp excluded-address 10.0.20.2
ip dhcp excluded-address 10.0.30.2
ip dhcp excluded-address 10.0.40.2
ip dhcp excluded-address 10.0.90.2
!
ip dhcp pool guest
   network 10.0.20.0 255.255.255.0
   default-router 10.0.20.2
!
ip dhcp pool corporate
   network 10.0.30.0 255.255.255.0
   default-router 10.0..30.2
!
ip dhcp pool engineering
   network 10.0.40.0 255.255.255.0
   default-router 10.0.40.2
!
ip dhcp pool restricted
   network 10.0.50.0 255.255.255.0
   default-router 10.0.50.2
!
ip dhcp pool unauthenticated
   network 10.0.90.0 255.255.255.0
   default-router 10.0.90.2
!
```

Task 4: Configure the AAA Service

In this task, you will configure the switch for 802.1x authentication and configure the switch-to-RADIUS-server communications.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
switch# configure terminal
```

Step 2 Create a local username and password.

```
switch(config)# username cisco password 0 cisco
```

Step 3 Enable AAA.

```
switch(config)# aaa new-model
```

Step 4 Create an IEEE 802.1x authentication method list.

```
switch(config)# aaa authentication dot1x default group radius
```

To create a default list that is used when a named list is *not* specified in the **authentication** command, use the **default** keyword followed by the method that is to be used in default situations. The default method list is automatically applied to all ports.

You will enter the **group radius** keyword to use the list of all RADIUS servers for authentication.

Note	Though other keywords are visible in the command-line help string, only the **default** and **group radius** keywords are supported.

Step 5 Enable IEEE 802.1x authentication globally on the switch.

```
switch(config)# dot1x system-auth-control
```

Step 6 Configure the switch for user RADIUS authorization for all network-related service requests.

```
switch(config)# aaa authorization network default group radius
```

Note	To allow VLAN assignment, you must enable AAA authorization to configure the switch for all network-related service requests.

Step 7 Specify the IP address of the RADIUS server.

```
switch(config)# radius-server host 10.0.P.12
```

Step 8 Specify the authentication and encryption key.

```
switch(config)# radius-server key radiuskey
```


Note	Using the previous example, you are specifying RADIUS servers separately that use the same key (**radiuskey**). You can also list RADIUS servers separately with their own specific keys by using the **radius-server host** {*hostname* | *ip-address*} **auth-port** *port-number* **key** *string* command.

Step 9 Assign the device VLAN interface as the RADIUS source interface.

```
switch(config)# ip radius source-interface vlan 30P
```

Activity Verification

You have completed this task when you attain these results:

- Review your configuration using the **show running-config** command.

```
switch# show running-config
!
aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
!
dot1x system-auth-control
!
ip radius source-interface Vlan101
radius-server host 10.0.1.12 auth-port 1812 acct-port 1813
radius-server retransmit 3
radius-server key radiuskey
!
```

Task 5: Configure Port for 802.1x Authentication with VLAN Assignment and Reauthentication

In this task, you will configure a port for 802.1x authentication with VLAN assignment.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
switch# configure terminal
```

Step 2 Enter interface configuration mode.

```
switch(config)# interface FastEthernet 0/1
```

Step 3 Set the port to access mode only.

```
switch(config-if)# switchport mode access
```

Step 4 Set the port to the initial (unauthenticated) VLAN.

```
switch(config-if)# switchport access vlan 90
```

Step 5 Enable IEEE 802.1x authentication on the interface.

```
switch(config-if)# dot1x port-control auto
```

Step 6 Enable periodic reauthentication of the client.

```
switch(config-if)# dot1x reauthentication
```

Step 7 Set the number of seconds based on the value of the Session-Timeout RADIUS attribute (attribute 27) and Termination-Action RADIUS attribute (attribute 29).

```
switch(config-if)# dot1x timeout reauth-period server
```

Step 8 Specify an active VLAN as an IEEE 802.1x guest VLAN.

```
switch(config-if)# dot1x guest-vlan 20
```

Step 9 Specify an active VLAN as an IEEE 802.1x restricted VLAN.

```
switch(config-if)# dot1x auth-fail vlan 50
```

Step 10 (Optional) Specify a number of authentication attempts to allow before a port moves to the restricted VLAN.

```
switch(config-if)# dot1x auth-fail max-attempts 2
```

Note The range is 1 to 3, and the default is 3.

Step 11 Return to privileged EXEC mode.

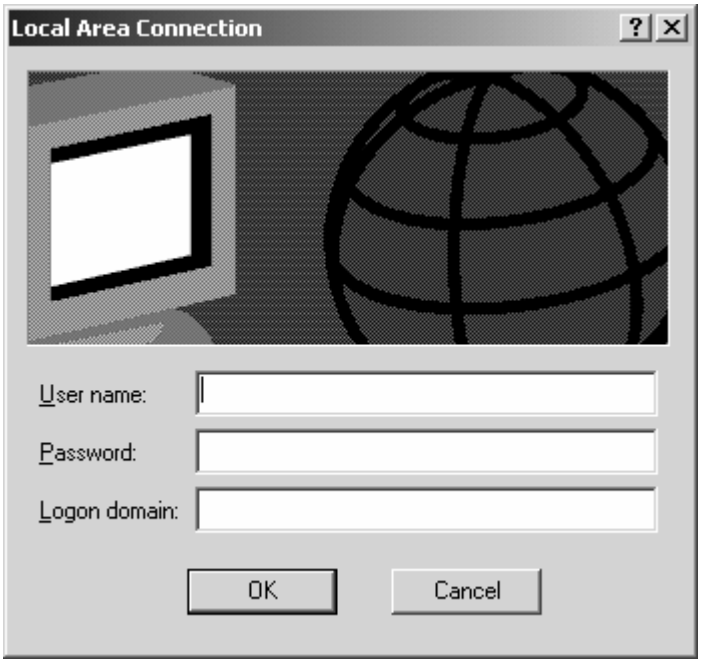
```
switch(config-if)# end
```


Activity Verification

You have completed this task when you attain these results:

- Connect a client into the switch that has an 802.1x supplicant.

You should get a prompt for your user credentials as follows:



Input a valid username and password. Authentication will then take place and you will be put into the proper VLAN.

Note If you are using a Microsoft Windows XP client and you do not see this dialog box, check your registry settings under HKEY_LOCAL_MACHINE > Software > Microsoft > EAPOL > Parameters > General > Global > AuthMode=0. Sometimes, the AuthMode default setting is set to 2. AuthMode = 2 will not ever do user authentication. It will only attempt machine authentication. This will produce an “unknown cs_user” error in the failed attempts report in Cisco Secure ACS.

The output of the **show dot1x** command should resemble the following:

```
switch# show dot1x all
Dot1x Info for interface FastEthernet0/1
-----------------------------------------------------
Supplicant MAC 0050.daeb.43d4
AuthSM State          = AUTHENTICATED
BendSM State          = IDLE
Posture               = N/A
   ReAuthPeriod      = 3600 Seconds (From Authentication Server)
   ReAuthAction      = Reauthenticate
   TimeToNextReauth  = 3112 Seconds
PortStatus            = AUTHORIZED
```

```
        MaxReq                = 2
        MaxAuthReq            = 2
        HostMode              = Single
        Port Control          = Auto
        ControlDirection      = Both
        QuietPeriod           = 60 Seconds
        Re-authentication     = Enabled
        ReAuthPeriod          = From Authentication Server
        ServerTimeout         = 30 Seconds
        SuppTimeout           = 30 Seconds
        TxPeriod              = 30 Seconds
        Guest-Vlan            = 0
        AuthFail-Vlan         = 0
        AuthFail-Max-Attempts = 3
```

```
switch# show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
--
1    default                          active    Fa0/3, Fa0/5, Fa0/6, Fa0/7
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15,
Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20,
Fa0/21
                                                Fa0/22, Gi0/1, Gi0/2
10   server                           active    Fa0/23
20   guest                            active
30   corporate                        active    Fa0/1
40   engineering                      active
50   restricted                       active
90   unauthenticated                   active    Fa0/2, Fa0/8
101  network_devices                  active    Fa0/4, Fa0/16
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup

switch# show interfaces status

Port      Name               Status       Vlan       Duplex  Speed Type
```

```
Fa0/1     Client             connected    30         a-full  a-100
10/100BaseTX
Fa0/2     Client             notconnect   90           auto   auto
10/100BaseTX
```

- Connect a client into the switch that *does not have* the 802.1x supplicant. You will not get a prompt for credentials. The output of the **show dot1x** command should resemble the following:

```
switch# show dot1x
Sysauthcontrol                    = Enabled
Supplicant Allowed In Guest Vlan  = Disabled
Dot1x Protocol Version            = 1


switch# show dot1x all
Dot1x Info for interface FastEthernet0/1
--------------------------------------------------
Supplicant MAC <Not Applicable>
AuthSM State          = AUTHENTICATED(GUEST_VLAN)
BendSM State          = IDLE
Posture               = N/A
   ReAuthPeriod      = None (From Authentication Server)
   ReAuthAction      = N/A
   TimeToNextReauth  = N/A
PortStatus            = AUTHORIZED(GUEST-VLAN)
MaxReq                = 2
MaxAuthReq            = 2
HostMode              = Single
Port Control          = Auto
ControlDirection      = Both
QuietPeriod           = 60 Seconds
Re-authentication     = Enabled
ReAuthPeriod          = From Authentication Server
ServerTimeout         = 30 Seconds
SuppTimeout           = 30 Seconds
TxPeriod              = 30 Seconds
Guest-Vlan            = 20
AuthFail-Vlan         = 50
AuthFail-Max-Attempts = 3
```

```
router# show ip dhcp binding
Bindings from all pools not associated with VRF:
```

```
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.0.20.6           0100.1125.8709.75       Jun 20 2006 02:09 PM    Automatic
```

- Connect a client that has an 802.1x supplicant but enter a bad username or password. The output of the **show dot1x** command should resemble the following:

```
switch# show dot1x all
Dot1x Info for interface FastEthernet0/1
--------------------------------------------------
Supplicant MAC 0011.2587.0975
AuthSM State          = AUTHENTICATED(AUTH-FAIL-VLAN)
BendSM State          = IDLE
Posture               = N/A
   ReAuthPeriod      = None (From Authentication Server)
   ReAuthAction      = N/A
   TimeToNextReauth  = N/A
PortStatus            = AUTHORIZED(AUTH-FAIL-VLAN)
MaxReq                = 2
MaxAuthReq            = 2
HostMode              = Single
Port Control          = Auto
ControlDirection      = Both
QuietPeriod           = 60 Seconds
Re-authentication     = Enabled
ReAuthPeriod          = From Authentication Server
ServerTimeout         = 30 Seconds
SuppTimeout           = 30 Seconds
TxPeriod              = 30 Seconds
Guest-Vlan            = 20
AuthFail-Vlan         = 50
AuthFail-Max-Attempts = 3
```

```
router# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.0.50.6           0100.1125.8709.75       Jun 20 2006 02:09 PM    Automatic
```


Task 8: Display 802.1x Statistics and Status

In this task, you will use some commands to view 802.1x status and statistics.

Activity Procedure

Complete these steps:

Step 1 Display IEEE 802.1x statistics for a specific interface.

```
switch# show dot1x statistics interface FastEthernet 0/1
```

Step 2 Display the IEEE 802.1x administrative and operational status for the switch.

```
switch# show dot1x all
```

Step 3 Display the IEEE 802.1x administrative and operational status for a specific interface.

```
switch# show dot1x interface FastEthernet 0/1
```

Activity Verification

You have completed this task when you attain these results:

- Use various options of the **show dot1x** command to view various settings.

```
switch# show dot1x statistics interface fa0/1
PortStatistics Parameters for Dot1x
---------------------------------------------
TxReqId = 3     TxReq = 3       TxTotal = 5
RxStart = 0     RxLogoff = 0    RxRespId = 0    RxResp = 0
RxInvalid = 0   RxLenErr = 0    RxTotal= 0
RxVersion = 0   LastRxSrcMac 0000.0000.0000

switch# show dot1x all
Dot1x Info for interface FastEthernet0/1
---------------------------------------------------
Supplicant MAC 0050.daeb.43d4
AuthSM State          = AUTHENTICATED
BendSM State          = IDLE
Posture               = N/A
   ReAuthPeriod      = 3600 Seconds (From Authentication Server)
   ReAuthAction      = Reauthenticate
   TimeToNextReauth  = 3593 Seconds
PortStatus            = AUTHORIZED
MaxReq                = 2
MaxAuthReq            = 2
HostMode              = Single
Port Control          = Auto
```

```
ControlDirection      = Both
QuietPeriod           = 60 Seconds
Re-authentication     = Enabled
ReAuthPeriod          = From Authentication Server
ServerTimeout         = 30 Seconds
SuppTimeout           = 30 Seconds
TxPeriod              = 30 Seconds
Guest-Vlan            = 20
AuthFail-Vlan         = 50
AuthFail-Max-Attempts = 3

switch# show dot1x interface FastEthernet 0/1
Supplicant MAC 0011.2587.0975
AuthSM State          = AUTHENTICATED
BendSM State          = IDLE
Posture               = N/A
   ReAuthPeriod      = 3600
   ReAuthAction      = N/A
   TimeToNextReauth  = 2439
PortStatus            = AUTHORIZED
MaxReq                = 2
MaxAuthReq            = 2
HostMode              = Single
Port Control          = Auto
ControlDirection      = Both
QuietPeriod           = 60 Seconds
Re-authentication     = Enabled
ReAuthPeriod          = From Authentication Server
ServerTimeout         = 30 Seconds
SuppTimeout           = 30 Seconds
TxPeriod              = 30 Seconds
Guest-Vlan            = 20
AuthFail-Vlan         = 50
AuthFail-Max-Attempts = 3
```


Lab 3-1: Configure Cisco NFP

Complete this lab activity to practice what you learned in the related module.

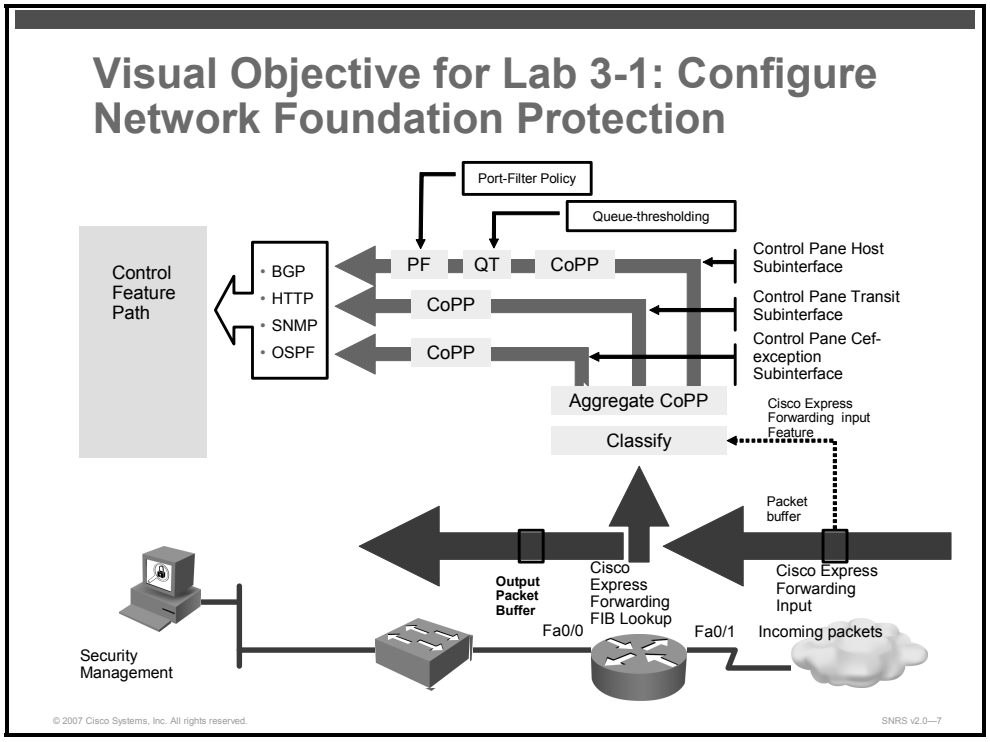
Activity Objective

In this activity, you will configure control, management, and data plane protection from the command line on a Cisco router. After completing this activity, you will be able to meet these objectives:

- Define packet classification criteria for CoPP
- Define a CoPP service policy
- Enter control plane configuration mode
- Apply a CoPP service policy
- Configure a port-filter policy
- Configure a queue-threshold policy
- Use **show** commands to verify CPPr
- Enter MPP configuration mode
- Designate one or more interfaces as a management interface and configure the management protocols that will be allowed on the management interfaces
- Load a PHDF
- Create a traffic class for FPM
- Create a traffic policy for FPM
- Apply an FPM filter policy to an interface

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Pod routers
- Student laptops

Command List

The table describes the commands that are used in this activity.

Network Foundation Protection Commands

Command	Description
class-map [**match-any** | **match-all**] *class-map-name*	Matches packets to a specified class
match {**access-group** | **name** *access-group-name*}	Specifies the match criteria for the class map
ip access list extended *access-group-name*	Creates an extended ACL
policy-map *policy-map-name*	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy
class *class-name*	Specifies the name of the class whose policy you want to create or change
police rate [burst-normal] [burst-max] [**pps**] **conform-action** action **exceed-action** action [**violate-action** action]	Configures traffic policing
control-plane [**host**|**transit**|**cef-exception**]	Enters control plane configuration mode and applies a CoPP, port-filter policy, or queue-threshold policy to police traffic destined for the control plane
service-policy {**input** | **output**} *policy-map-name*	Attaches a QoS service policy to the control plane **Note** This command is used in aggregate control plane configuration mode.
class-map type port-filter [**match-all** | **match-any**] *class-name*	Creates a class map used to match packets to a specified class and enables the port-filter class-map configuration mode
match {**closed-ports**|**not**|**port**} {**TCP**|**UDP**} **0-65535**	Specifies the TCP/UDP match criteria for the class map
policy-map type port-filter *policy-map-name*	Creates a port-filter service policy and enters the policy-map configuration mode
drop	Applies the port-filter service policy drop action on the class
service-policy type port-filter {**input**} *port-filter-policy-map-name*	Attaches a port-filter service policy to the control plane host subinterface
class-map type queue-threshold [**match-all** | **match-any**] *class-name*	Enables queue thresholding that limits the total number of packets for a specified protocol that is allowed in the control plane IP input queue
match protocol [**bgp** | **dns** | **ftp** | **http** | **igmp** | **snmp** | **ssh** | **syslog** | **telnet**| **tftp** | **host-protocols**]	Specifies the ULP match criteria for the class map
policy-map type queue- threshold *policy-name*	Enables the queue-threshold service policy configuration mode

queue-limit *number*	Applies the queue-threshold service policy action on the class
service-policy type queue-threshold {**input**} *queue-threshold-policy-map-name*	Attaches a queue-threshold service policy to the control plane
management-interface *interface* **allow** *protocols*	Configures an interface to be a management interface
load protocol *location:filename*	Loads a PHDF onto a router
class-map type stack [**match-all** | **match-any**] *class-name*	Enables FPM to determine the correct protocol stack in which to examine
match field *protocol protocol-field* {**eq** [*mask*] | **neq** [*mask*] | **gt** | **lt** | **range** *range* | **regex** *string*} *value* [**next** *next-protocol*]	Configures the match criteria for a class map on the basis of the fields defined in the protocol header
class-map type access-control [**match-all** | **match-any**] *class-map-name*	Determines the exact pattern to look for in the protocol stack of interest
match start {**l2-start** | **l3-start**} **offset** *number* **size** *number* {**eq** | **neq** | **gt** | **lt** | **range** *range* | **regex** *string*} {*value* [*value2*] | [*string*]}	Configures the match criteria for a class map on the basis of the datagram header (Layer 2) or the network header (Layer 3)
policy-map type access-control *policy-map-name*	Creates or modifies a policy map that can determine the exact pattern to look for in the protocol stack of interest
service-policy type access-control {**input** | **output**} *policy-map-name*	Attaches a policy map to an input interface
show class-map	Displays all class maps and their matching criteria
show policy-map	Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps
show policy-map interface	Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface
show policy-map control-plane	Displays the configuration either of a class or of all classes for the policy map of a control plane
show management-interface [*interface* | **protocol** *protocol-name*]	Displays all management interface configurations and activity on a device and filters the output by interface or protocol
show class-map type stack	Displays class maps that are configured to determine the correct protocol stack in which to examine via FPM
show class-map type access-control	Displays class maps that are configured to determine the exact pattern to look for in the protocol stack of interest

Job Aids

There are no job aids for this activity.

Configuring CPPr

Task 1: Define Packet Classification Criteria for CoPP

In this task, you will create a class map and define criteria for the class map.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Define an ACL for trusted hosts using specific protocols to access the router.

```
router(config)# ip access list extended IP access list CP-acl
router(config-ext-nacl)# deny tcp host 10.0.P.12 any eq telnet
router(config-ext-nacl)# deny tcp host 10.0.P.12 any eq www
router(config-ext-nacl)# permit tcp any any eq telnet
router(config-ext-nacl)# permit tcp any any eq www
```

Step 3 Exit back to global configuration mode.

```
router(config-ext-nacl)# exit
```

Step 4 Enable class map global configuration command mode.

```
router(config)# class-map match-any CP-class
```

Step 5 Specify the criteria to match. In this case, you will match to an ACL.

```
router(config-cmap)# match access-group name CP-acl
```

Step 6 Exit back to global configuration mode.

```
router(config-cmap)# exit
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show class-map** and **show ip access-lists** commands should resemble the following:

```
router# show class-map
 Class Map match-any class-default (id 0)
   Match any

 Class Map match-any CP-class (id 2)
   Match access-group name CP-acl

router# show ip access-lists
Extended IP access list CP-acl
    10 deny tcp host 10.0.1.12 any eq telnet
```

```
20 deny tcp host 10.0.1.12 any eq www
30 permit tcp any any eq telnet
40 permit tcp any any eq www
```

Task 2: Define a CoPP Service Policy

In this task, you will define a CoPP service policy using a policy map.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Enter policy map configuration mode to define a policy.

```
router(config)# policy-map CP-policy
```

Step 3 Enter class map configuration mode within the policy map mode.

```
router(config-pmap)# class CP-class
```

Step 4 Configure traffic policing.

```
router(config-pmap-c)# police rate 50000 pps conform-action
transmit exceed-action drop
```

Step 5 Return to privileged EXEC mode.

```
router(config-pmap-c)# end
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show policy-map** command should resemble the following:

```
router# show policy-map
  Policy Map CP-policy
    Class CP-class
     police rate 50000 pps burst 12207 packets
       conform-action transmit
       exceed-action drop

router# show policy-map CP-policy
  Policy Map CP-policy
    Class CP-class
     police rate 50000 pps burst 12207 packets
       conform-action transmit
       exceed-action drop
```


Task 3: Apply CoPP Service Policy to the Control Plane Host Subinterface

In this task, you will enter the control plane configuration mode.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Enter aggregate control plane configuration mode to attach a QoS policy that manages control plane traffic to a specified control plane subinterface.

```
router(config)# control-plane host
```

Step 3 Attach your QoS service policy to the control plane.

```
router(config-cp)# service-policy input CP-policy
```

Step 4 Exit back to privileged EXEC mode.

```
router(config-cp)# end
```

Activity Verification

You have completed this task when you attain these results:

- Telnet to 10.0.P.2 to generate traffic to the control plane.
- The output of the **show policy-map control-plane host** command should resemble the following:

```
router# show policy-map control-plane host
 Control Plane Host
  Service-policy input: CP-policy
    Class-map: CP-class (match-any)
      1704 packets, 102240 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group name CP-acl
        1704 packets, 102240 bytes
        5 minute rate 0 bps
      police:
          rate 50000 pps, burst 12207 packets
        conformed 3400 packets; actions:
          transmit
        exceeded 0 packets; actions:
          drop
        conformed 2 pps, exceed 0 pps


    Class-map: class-default (match-any)
```

```
2202 packets, 213406 bytes
5 minute offered rate 2000 bps, drop rate 0 bps
Match: any
```

Task 4: Configure a Port-Filter Policy

In this task, you will configure a port-filter policy on the host subinterface of the control plane.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Create a class map of type “port-filter” and specify the criteria to match.

```
router(config)# class-map type port-filter match-all PF-class
```

Step 3 Specify the TCP/UDP match criteria for the class map. In this lab, you will match all closed ports.

```
router(config-cmap)# match closed-ports
```

Step 4 Exit to global configuration mode.

```
router(config-cmap)# exit
```

Step 5 Create a service policy of type “port-filter” and enter the policy map configuration mode.

```
router(config)# policy-map type port-filter PF-policy
```

Step 6 Associate a service policy with a class and enter class map configuration mode.

```
router(config-pmap)# class PF-class
```

Step 7 Apply the port-filter service policy action on the class.

```
router(config-pmap-c)# drop
```

Step 8 Return to policy map configuration mode.

```
router(config-pmap-c)# exit
```

Step 9 Return to global configuration mode.

```
router(config-pmap)# exit
```

Step 10 Enter the control plane host subinterface configuration mode.

```
router(config)# control-plane host
```

Step 11 Attach a service policy of type “port-filter” to the control plane host subinterface.

```
router(config-cp-host)# service-policy type port-filter input
PF-policy
```

Step 12 Return to privileged EXEC mode.

```
router(config-cp-host)# end
```


Activity Verification

You have completed this task when you attain these results:

- The output of the **show class-map type port-filter** and **show policy-map type port-filter** commands should resemble the following:

```
router# show class-map type port-filter
 Class Map type port-filter match-all PF-class (id 3)
   Match  closed-ports

router# show policy-map type port-filter
  Policy Map type port-filter PF-policy
    Class PF-class
      drop

router# show policy-map type port-filter control-plane host
      drop
 Control Plane Host
  Service-policy port-filter input: PF-policy
    Class-map: PF-class (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:  closed-ports

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 5: Configure a Queue-Threshold Policy

In this task, you will create a queue-threshold policy on the host subinterface of the control plane.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Create a class map of type “queue-threshold” and specify the criteria to match.

```
router(config)# class-map type queue-threshold match-all QT-
class
```

Step 3 Specify the ULP match criteria for the class map. In this lab, the ULP will be BGP.

```
router(config-cmap)# match protocol bgp
```

Step 4 Return to global configuration mode.

```
router(config-cmap)# exit
```

Step 5 Create a service policy of type “queue-threshold” and enter the policy map configuration mode.

```
router(config)# policy-map type queue-threshold QT-policy
```

Step 6 Enter class map configuration mode.

```
router(config-pmap)# class QT-class
```

Step 7 Apply the queue-threshold service policy action on the class.

```
router(config-pmap-c)# queue-limit 100
```

Step 8 Return to global configuration mode.

```
router(config-pmap-c)# exit
```

Step 9 Enter the control plane host subinterface configuration mode.

```
router(config)# control-plane host
```

Step 10 Attach the service policy to the control plane.

```
router(config-cp-host)# service-policy type queue-threshold
input QT-policy
```

Step 11 Return to privileged EXEC mode.

```
router(config-cp-host)# end
```


Activity Verification

You have completed this task when you attain these results:

- The output of the **show class-map type queue-threshold** and **show policy-map type queue-threshold** commands should resemble the following:

```
router# show class-map type queue-threshold
 Class Map type queue-threshold match-all QT-class (id 1)
   Match  protocol bgp

router# show policy-map type queue-threshold
  Policy Map type queue-threshold QT-policy
    Class QT-class
      queue-limit 100

router# show policy-map type queue-threshold control-plane host
      queue-limit 100
      queue-count 0     packets allowed/dropped 0/0
 Control Plane Host

  Service-policy queue-threshold input: QT-policy

    Class-map: QT-class (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:  protocol bgp

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Configuring MPP

Task 6: Enter Control Plane Host Configuration Mode

In this task, you will configure management plane protection.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Enter control plane host configuration mode.

```
router(config)# control-plane host
```

Activity Verification

You have completed this task when you attain these results:

- You will verify this activity after the next task.

Task 7: Specify Management Interface and Protocols

In this task, you will specify the management interface and allowed protocols.

Activity Procedure

Complete these steps:

Step 1 Configure an interface to be a management interface and specify which management protocols are allowed.

```
router(config-cp-host)# management-interface Fa0/0 allow ssh
SNMP
```

Step 2 Return to privileged EXEC mode.

```
router(config-cp-host)# end
```

Activity Verification

You have completed this task when you attain these results:

1. Try to telnet to 10.0.P.2. You should fail unless you entered telnet as an “allowed” management protocol.
2. Now use SSH to connect to 10.0.P.2. You should be able to connect using SSH.

- The output of the **show management-interface** command should resemble the following:

```
router# show management-interface
Management interface FastEthernet0/1
        Protocol        Packets processed
             ssh               43
            snmp                0
```


Configuring FPM

Task 8: Load a PHDF

In this task, you will load two PHDFs.

Note	Make sure that the PHDFs are stored in flash memory for use in this lab.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Load the PHDFs on the router.

```
router(config)# load protocol flash:ip.phdf
router(config)# load protocol flash:udp.phdf
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show protocols phdf** command should resemble this:

```
router# show protocols phdf ip
Protocol ID: 1
Protocol name: IP
Description: IP-Protocol
Original file name: flash:ip.phdf
Header length: 20
Constraint(s):
        Protocol ID: 1
        Field ID: 0
        Match Value: 4
        Operator is eq

        Protocol ID: 1
        Field ID: 1
        Match Value: 5
        Operator is eq

Total number of fields: 13

        Field id: 0, version, IP-Version
```

```
Fixed offset. offset 0
Constant length. Length: 4

Field id: 1, ihl, IP-Header-Length
Fixed offset. offset 4
Constant length. Length: 4

Field id: 2, tos, IP-Type-Of-Service
Fixed offset. offset 8
Constant length. Length: 8

Field id: 3, length, IP-Packet-Length
Fixed offset. offset 16
Constant length. Length: 16

Field id: 4, identification, IP-Identification
Fixed offset. offset 32
Constant length. Length: 16

Field id: 5, flags, IP-Fragmentation-Flags
Fixed offset. offset 48
Constant length. Length: 3

Field id: 6, fragment-offset, IP-Fragmentation-Offset
Fixed offset. offset 51
Constant length. Length: 13

Field id: 7, ttl, IP-TTL
Fixed offset. offset 64
Constant length. Length: 8

Field id: 8, protocol, IP-Protocol
Fixed offset. offset 72
Constant length. Length: 8

Field id: 9, checksum, IP-Header-Checksum
Fixed offset. offset 80
Constant length. Length: 16

Field id: 10, source-addr, IP-Source-Address
```



```
Fixed offset. offset 96
Constant length. Length: 32

Field id: 11, dest-addr, IP-Destination-Address
Fixed offset. offset 128
Constant length. Length: 32

Field id: 12, payload-start, IP-Payload-Start
Fixed offset. offset 160
Constant length. Length: 0
```

Task 9: Create a Traffic Class

In this task, you will create two types of class maps. One of type “stack” used to define a stack of protocol headers and another of type “access-control” used to classify packets.

Activity Procedure

Complete these steps:

Step 1 Create a class map of type “stack” to define the sequence of headers as IP first, then UDP.

```
router(config)# class-map type stack match-all ip-udp
```

Step 2 Add a description to the class map.

```
router(config-cmap)# description match UDP over IP packets
```

Step 3 Create the match criteria.

```
router(config-cmap)# match field ip protocol eq 0x11 next udp
```

Note UDP is protocol 0x11 in hexadecimal format, which is 17 in decimal format.

Step 4 Return to global configuration mode.

```
router(config-cmap)# exit
```

Step 5 Create a class map of type “access-control” for classifying packets.

```
router(config)# class-map type access-control match-all
slammer
```

Step 6 Add a description to this class map.

```
router(config-cmap)# description match on slammer packets
```

Step 7 Create match criteria.

```
router(config-cmap)# match field udp dest-port eq 0x59A
```

Note Port 0x59A in hexadecimal format is port 1434 in decimal format—a known slammer port also used in monitoring Microsoft SQL databases.

```
router(config-cmap)# match field ip length eq 0x194
router(config-cmap)# match start l3-start offset 224 size 4 eq
0x4011010
```

Step 8 Return to privileged EXEC mode.

```
router(config-cmap)# end
```


Activity Verification

You have completed this task when you attain these results:

- The output of the **show class-map type stack** command should resemble this:

```
router# show class-map type stack
 Class Map type stack match-all ip-udp (id 4)
  Description: match UDP over IP packets
   Match field IP protocol eq 0x11 next UDP

router# show class-map type access-control
 Class Map type access-control match-all slammer (id 5)
  Description: match on slammer packets
   Match field UDP dest-port eq 0x59A
   Match field IP length eq 0x194
   Match start l3-start offset 224 size 4 eq 0x4011010
```

Task 10: Create a Traffic Policy

In this task, you will create a policy map to define the traffic policy for an interface.

Activity Procedure

Complete these steps:

Step 1 Specify the policy map that associates the class defined with an action.

```
router(config)# policy-map type access-control fpm-udp-policy
```

Step 2 Give the policy a description.

```
router(config-pmap)# description policy for UDP based attacks
```

Step 3 Specify the associated class map.

```
router(config-pmap)# class slammer
```

Step 4 Specify the action to be taken.

```
router(config-pmap-c)# drop
```

Step 5 Exit to policy map configuration mode.

```
router(config-pmap-c)# exit
```

Step 6 Exit to global configuration mode.

```
router(config-pmap)# exit
```

Step 7 Within the final policy definition, you will first specify the “ip-udp” class so that only UDP packets are inspected by the policy defined in Step 1 above. Then, specify the “fpm-udp-policy” policy map to complete the classification and drop action.

```
router(config)# policy-map type access-control fpm-policy
router(config-pmap)# description drop worms and malicious
attacks
router(config-pmap)# class ip-udp
router(config-pmap-c)# service-policy fpm-udp-policy
```

Step 8 Return to privileged EXEC mode.

```
router(config-pmap-c)# end
```


Activity Verification

You have completed this task when you attain these results:

- The output of the **show policy-map type access-control** command should resemble this:

```
router# show policy-map type access-control
 Policy Map type access-control fpm-udp-policy
   Description: policy for UDP based attacks
  Class slammer
   drop

 Policy Map type access-control fpm-policy
   Description: drop worms and malicious attacks
  Class ip-udp
   service-policy fpm-udp-policy
```

Task 11: Apply Service Policy to an Interface

In this task, you will apply the policy to the perimeter interface of your network.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode.

```
router# configure terminal
```

Step 2 Enter interface configuration mode on your external interface.

```
router(config)# interface FastEthernet 0/0
```

Step 3 Apply the policy to this interface.

```
router(config-if)# service-policy type access-control input
fpm-policy
```

Step 4 Return to privileged EXEC mode.

```
router(config-if)# end
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show policy-map type access-control interface <*int*>** command should resemble this:

```
router# show policy-map type access-control interface FastEthernet 0/0
FastEthernet0/1

  Service-policy access-control input: fpm-policy

    Class-map: ip-udp (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: field IP version eq 4
      Match: field IP ihl eq 5
      Match: field IP protocol eq 0x11 next UDP

      Service-policy access-control : fpm-udp-policy

        Class-map: slammer (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: field UDP dest-port eq 0x59A
          Match: field IP length eq 0x194
          Match: start l3-start offset 224 size 4 eq 0x4011010

        Class-map: class-default (match-any)
          0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
          Match: any

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```


Lab 4-1: Configure a Site-to-Site VPN using Pre-Shared Keys

Complete this lab activity to practice what you learned in the related module.

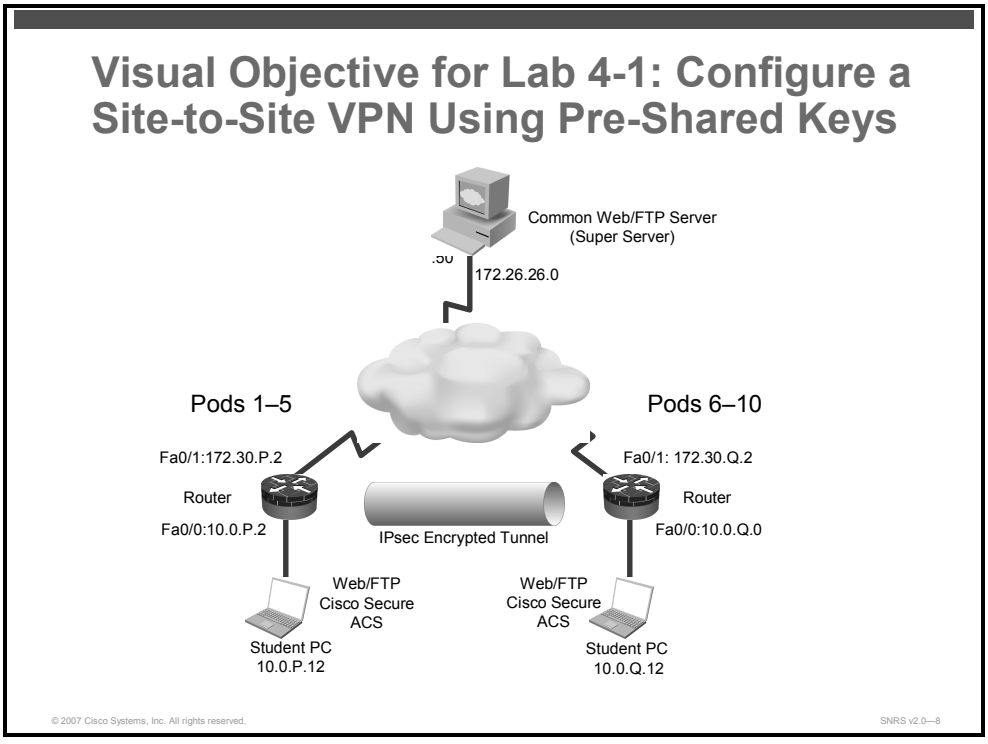
Activity Objective

In this activity, you will configure a perimeter router for site-to-site VPNs using pre-shared keys. After completing this activity, you will be able to meet these objectives:

- Set up lab devices
- Prepare for configuring IPsec
- Create an ISAKMP policy to use pre-shared keys
- Configure transform sets
- Configure a crypto ACL
- Configure a crypto map
- Apply the crypto map to an interface
- Ensure that encryption is working between routers

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers

Command List

The table describes the commands that are used in this activity.

IPsec Commands

Command	Description
access-list *access-list-number*	Creates a numbered ACL
authentication {**rsa-sig** | **rsa-encr** | **pre-share**}	Specifies the authentication method within an IKE policy
clear crypto sa	Deletes IPsec SAs
crypto ipsec transform-set *transform-set-name transform1 transform2 ..*	Defines an IPsec transform set
crypto isakmp enable	Globally enables IKE
crypto isakmp identity {**address** | **hostname**}	Defines the identity used by the router when participating in the IKE protocol
crypto isakmp key *key-string* **address** *peer-address* [*mask*] [**no-xauth**]	Configures a pre-shared authentication key
crypto isakmp policy *priority*	Defines an IKE policy
encryption {**des** | **3des** | **aes** | **aes 192** | **aes 256**}	Specifies the encryption algorithm within an IKE policy
group {**1** | **2**}	Specifies the DH group identifier within an IKE policy
hash {**sha** | **md5**}	Specifies the hash algorithm within an IKE policy
lifetime *seconds*	Specifies the lifetime of an IKE SA
crypto map *map-name seq-num* [**ipsec-isakmp**]	(Global IPsec) Enters crypto map configuration mode and specifies that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry
crypto map *map-name* [**redundancy** *standby-group-name*[**stateful**]]	(Interface IPsec) Applies a previously defined crypto map set to an interface
match address [*access-list-id* | *name*]	Specifies a crypto ACL for a crypto map entry
mode [**tunnel** | **transport**]	Changes the mode for a transform set
set peer {*host-name* | *ip-address*}	Specifies an IPsec peer in a crypto map entry
set transform-set *transform-set-name* [*transform-set-name2...transform-set-name6*]	Specifies which transform sets can be used with the crypto map entry
ping *ip-address*	Diagnoses basic network connectivity
show crypto ipsec transform-set [**tag** *transform-set-name*]	Displays the configured transform sets
show crypto isakmp policy	Displays the parameters for each IKE policy
show crypto isakmp sa	Displays all current IKE SAs
show crypto ipsec sa	Displays all current IPsec SAs
show crypto map [**interface** *interface* | **tag** *map-name*]	Displays the crypto map configuration

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will complete the lab setup exercise by ensuring connectivity with other routers in the lab.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student laptop is operating with the correct date and time.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2**. (where P = pod number).

Step 3 Restore the original course router configuration. Your instructor will explain how to do this.

Step 4 Verify that you have connectivity with the peer pod router.

```
router# ping 172.30.Q.2
(where Q = peer pod number)
```

Activity Verification

You have completed this task when you attain these results:

- Ping the peer pod outside interface. Your output should resemble the following:

```
router# ping 172.30.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```


Task 2: Prepare for IPsec

In this task, you will prepare for configuring IPsec by determining the ISAKMP and IPsec policy and by creating an ACL to allow IPsec traffic.

Activity Procedure

Complete these steps:

Step 1 Determine the ISAKMP and IPsec policy. In this lab exercise, you will use default values except when you are directed to enter a specific value.

- The ISAKMP policy is to use pre-shared keys.
- The IPsec policy is to use ESP mode with 3DES encryption.
- The IPsec policy is to encrypt all traffic between the specified subnetworks.

Step 2 Create an ACL to allow IPsec protocols on the outside interface.

```
router# configure terminal
router(config)# ip access-list extended 102
router(config-ext-nacl)# permit ahp host 172.30.P.2 host
172.30.Q.2
router(config-ext-nacl)# permit esp host 172.30.P.2 host
172.30.Q.2
router(config-ext-nacl)# permit udp host 172.30.P.2 host
172.30.Q.2 eq isakmp
router(config-ext-nacl)# permit udp host 172.30.P.2 host
172.30.Q.2 eq 4500
```

Step 3 Exit to privileged EXEC mode.

```
router(config-ext-nacl)# end
```

Activity Verification

You have completed this task when you attain these results:

- Perform a **show ip access-lists** command. The output should be similar to this:

```
router# show ip access-lists
Extended IP access list 102
    10 permit ahp host 172.30.1.2 host 172.30.6.2
    20 permit esp host 172.30.1.2 host 172.30.6.2
    30 permit udp host 172.30.1.2 host 172.30.6.2 eq isakmp
    40 permit udp host 172.30.1.2 host 172.30.6.2 eq non500-isakmp
```

Task 3: Configure an ISAKMP Policy to Use Pre-Shared Keys

In this task, you will enable IKE/ISAKMP on the router and configure authentication using pre-shared keys.

Activity Procedure

Complete these steps:

Step 1 Verify that ISAKMP is enabled. You should see a default policy.

```
router# show crypto isakmp policy
```

Note If you see the message "ISAKMP is turned off," complete Step 2, then complete the rest of the steps. If ISAKMP is already enabled, skip Step 2.

```
R1# show crypto isakmp policy
Global IKE policy
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 2 Enable ISAKMP on the router.

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

Step 3 Set the policy priority and enter ISAKMP policy configuration mode.

```
router(config)# crypto isakmp policy 110
```

Step 4 Set authentication to use pre-shared keys.

```
router(config-isakmp)# authentication pre-share
```

Step 5 Set IKE encryption.

```
router(config-isakmp)# encryption 3des
```

Step 6 Set the DH group.

```
router(config-isakmp)# group 2
```

Step 7 Set the hash algorithm.

```
router(config-isakmp)# hash md5
```

Step 8 Set the ISAKMP SA lifetime.

```
router(config-isakmp)# lifetime 36000
```

Step 9 Exit the ISAKMP policy configuration mode.

```
router(config-isakmp)# exit
```


Step 10 Configure the pre-shared key and peer address.

```
router(config)# crypto isakmp key 0 cisco1234 address
172.30.Q.2
(where Q = peer pod number)
```

Step 11 Exit configuration mode.

```
router(config)# end
```

Step 12 Examine the crypto policy suite.

Activity Verification

You have completed this task when you attain these results:

- Your output is similar to this:

```
R1# show crypto isakmp policy
Global IKE policy
Protection suite of priority 110
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               36000 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
```

Task 4: Configure an IPsec Transform Set

In this task, you will configure an IPsec transform set.

Activity Procedure

Complete these steps:

Step 1 Define a transform set that includes the following:

- Transform name: **SNRS**
- ESP protocols: **esp-des**
- Mode: **tunnel**

```
router(config)# crypto ipsec transform-set SNRS esp-des
```

Step 2 Set the mode to tunnel.

```
router(cfg-crypto-trans)# mode tunnel
```

Step 3 Exit the configuration mode.

```
router(cfg-crypto-trans)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto ipsec transform-set** command. Your output should be similar to the following:

```
R1# show crypto ipsec transform-set
Transform set SNRS: { esp-des  }
   will negotiate = { Tunnel,  },
```

Task 5: Configure an IPsec Crypto ACL

In this task, you will create an ACL that "defines" traffic to protect. The ACL should encrypt traffic between the subnetworks that you specify. Use the following parameters:

- Traffic encrypted: Traffic between **10.0.P.0** and **10.0.Q.0**
- ACL number: **101**
- Protocol: **IP**

Activity Procedure

Complete these steps:

Step 1 Configure the crypto ACL.

```
router(config)# ip access-list extended 101
router(config-ext-nacl)# permit ip 10.0.P.0 0.0.0.255 10.0.Q.0
0.0.0.255
(where P = pod number, and Q = peer pod number)
```

Step 2 Exit to privileged EXEC mode.

```
router(config-ext-nacl)# end
```


Activity Verification

You have completed this task when you attain these results:

- Issue a **show access-list** command. The output should be similar to this:

```
R1# show ip access-lists
Extended IP access list 101
    10 permit ip 10.0.1.0 0.0.0.255 10.0.6.0 0.0.0.255
Extended IP access list 102
    10 permit ahp host 172.30.1.2 host 172.30.6.2
    20 permit esp host 172.30.1.2 host 172.30.6.2
    30 permit udp host 172.30.1.2 host 172.30.6.2 eq isakmp
    40 permit udp host 172.30.1.2 host 172.30.6.2 eq non500-isakmp
```

Task 6: Configure an IPsec Crypto Map

In this task, you will configure a crypto map. Use the following parameters:

- Name of map: **SNRS-MAP**
- Number of map: **10**
- Key exchange type: **isakmp**
- Peer: **172.30.Q.2**
- Transform set: **SNRS**
- Match address: **101**

Activity Procedure

Complete these steps:

Step 1 Set the name of the map, the map number, and the type of key exchange to be used.

```
router(config)# crypto map SNRS-MAP 10 ipsec-isakmp
```

You should see the following:

```
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
```

Step 2 Specify the extended ACL to use with this map.

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

Step 3 Specify the transform set that you defined earlier.

```
router(config-crypto-map)# set transform-set SNRS
```

Step 4 Assign the VPN peer using the hostname or IP address of the peer.

```
router(config-crypto-map)# set peer 172.30.Q.2
(where Q = peer pod number)
```

Step 5 Exit back to privileged EXEC mode.

```
router(config-crypto-map)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue the **show crypto map** command. The output should be similar to this:

```
R1# show crypto map
Crypto Map "SNRS-MAP" 10 ipsec-isakmp
        Peer = 172.30.6.2
        Extended IP access list 101
            access-list 101 permit ip 10.0.1.0 0.0.0.255 10.0.6.0
0.0.0.255
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                SNRS,
        }
        Interfaces using crypto map SNRS-MAP:
```


Task 7: Apply the Crypto Map to an Interface

In this task, you will apply the crypto map to an interface. Use the following parameters:

- Interface to configure: **FastEthernet 0/1**
- Crypto map to use: **SNRS-MAP**

Activity Procedure

Complete these steps:

Step 1 Access interface configuration mode.

```
router(config)# interface fastEthernet 0/1
```

Step 2 Assign the crypto map to the interface.

```
router(config-if)# crypto map SNRS-MAP
```

You should see the following message:

```
Jul 26 16:19:05.123: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

Step 3 Exit interface configuration mode.

```
router(config-if)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue the **show crypto map interface fa0/1** command. The output should be similar to this:

```
R1# show crypto map interface fastEthernet 0/1
Crypto Map "SNRS-MAP" 10 ipsec-isakmp
        Peer = 172.30.6.2
        Extended IP access list 101
            access-list 101 permit ip 10.0.1.0 0.0.0.255 10.0.6.0
0.0.0.255
        Current peer: 172.30.6.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                SNRS,
        }
        Interfaces using crypto map SNRS-MAP:
                FastEthernet0/1
```

Task 8: Ensure That Encryption Is Working Between Routers

In this task, you will generate traffic from your internal subnet to your peer pod internal subnet to ensure that encryption is working between the routers.

Activity Procedure

Complete these steps:

Step 1 Generate interesting traffic using an extended ping. You will ping from the inside interface of your pod router to the inside interface of your peer pod router. You can also ping from your laptop to the laptop of your peer pod.

```
R1# ping
Protocol [ip]:
Target IP address: 10.0.6.2
Repeat count [5]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: yes
Source address or interface: 10.0.1.2
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 10.0.6.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4
ms
```

Step 2 Display your ISAKMP SAs.

Step 3 Display your IPsec SAs

Activity Verification

You have completed this task when you attain these results:

- Verify that the IKE and IPsec SAs have been established.

```
R1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.6.2      172.30.1.2      QM_IDLE           1001    0 ACTIVE

IPv6 Crypto ISAKMP SA
R1# show crypto ipsec sa
interface: FastEthernet0/1
    Crypto map tag: SNRS-MAP, local addr 172.30.1.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (10.0.6.0/255.255.255.0/0/0)
   current_peer 172.30.6.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 6657, #pkts encrypt: 6657, #pkts digest: 6657
    #pkts decaps: 6656, #pkts decrypt: 6656, #pkts verify: 6656
    #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.: 172.30.1.2, remote crypto endpt.: 172.30.6.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x1B029B45(453155653)

     inbound esp sas:
      spi: 0xD74582A5(3611656869)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: FPGA:1, crypto map: SNRS-MAP
        sa timing: remaining key lifetime (k/sec): (4565588/2901)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE
```

```
inbound ah sas:

inbound pcp sas:

outbound esp sas:
 spi: 0x1B029B45(453155653)
   transform: esp-des ,
   in use settings ={Tunnel, }
   conn id: 2002, flow_id: FPGA:2, crypto map: SNRS-MAP
   sa timing: remaining key lifetime (k/sec): (4565588/2871)
   IV size: 8 bytes
   replay detection support: N
   Status: ACTIVE

outbound ah sas:

outbound pcp sas:
```


Lab 4-2: Configure a Site-to-Site VPN Using Certificates

Complete this lab activity to practice what you learned in the related module.

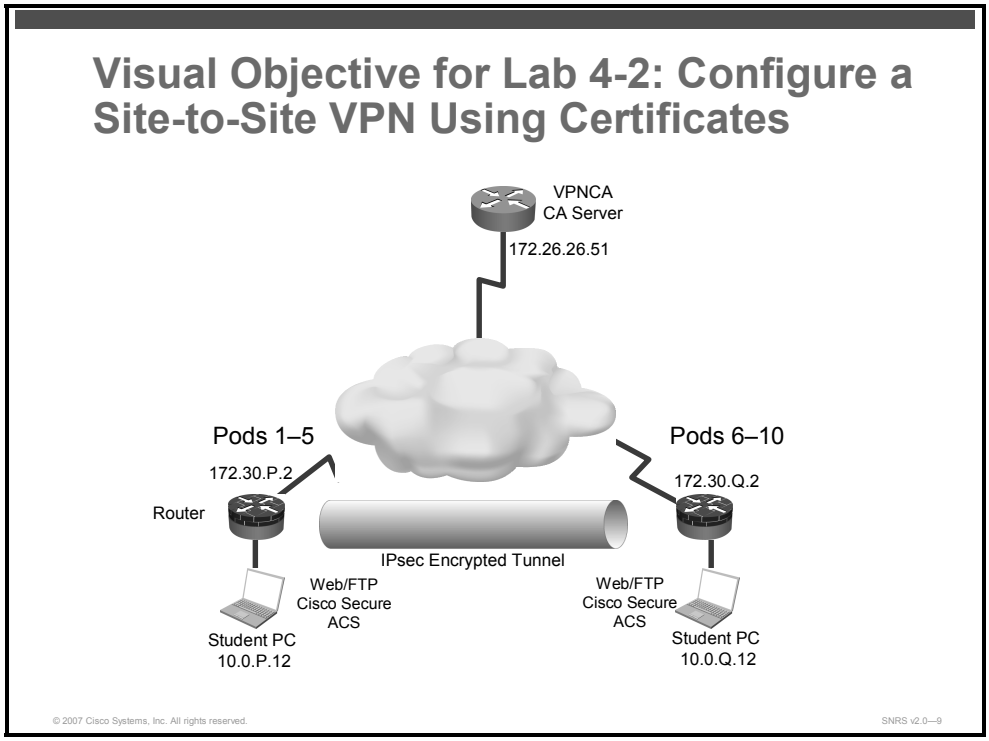
Activity Objective

In this activity, you will configure a perimeter router for site-to-site VPNs using a CA. After completing this activity, you will be able to meet these objectives:

- Set up lab devices
- Set the router date and time
- Define the domain name of the router
- Define the static hostname-to-IP address mapping of the CA server
- Generate RSA keys
- Configure the CA server trustpoint
- Create an IKE policy to use RSA signatures
- Configure transform sets and SA parameters
- Configure crypto ACLs
- Configure crypto maps
- Apply the crypto map to an interface
- Ensure that encryption is working

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers
- CA server

Command List

The table describes the commands that are used in this activity.

PKI Commands

Command	Description
ping [*protocol*] [**tag**] {*host-name* | *system-address*}	Diagnoses basic network connectivity on AppleTalk, ATM, CLNS, DECnet, IP, Novell IPX, or source-route bridging (SRB) networks
ip route *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [*name*] [**permanent**] [**tag** *tag*]	Establishes a static route
clock timezone *zone hours-offset* [*minutes-offset*]	Sets the time zone for display purposes
hostname *<name>*	Configures a hostname for the router (for RSA key pairs and certificates)
ip domain-name *<name>*	Configures a domain for the router (for RSA key pairs and certificates)
ip host {***name*** | **t***modem-telephone-number*} [*tcp-port-number*] {*address1* [*address2...address8*]}	Defines a static hostname-to-address mapping in the host cache
crypto key generate rsa	Generates RSA key pairs
crypto pki trustpoint	Declares the CA that your router should use
enrollment [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]	Specifies the enrollment parameters of a CA
crypto pki authenticate *<name>*	Authenticates the CA (by acquiring the certificate of the CA)
crypto pki enroll *<name>*	Obtains the certificate or certificates for your router from the CA
crypto isakmp enable	Globally enables IKE on a Cisco router
crypto isakmp policy *priority*	Defines an ISAKMP policy
authentication {**rsa-sig** | **rsa-encr** | **pre-share**}	Specifies the authentication method within an ISAKMP policy
encryption {**des** | **3des** | **aes** | **aes 192** | **aes 256**}	Specifies the encryption algorithm within an ISAKMP policy
group {**1** | **2**}	Specifies the DH group identifier within an IKE policy
hash {**sha** | **md5**}	Specifies the hash algorithm within an IKE policy
crypto ipsec transform-set *<name>* **esp-des**	Creates a transform set and specifies an ESP protocol
mode tunnel	Specifies tunnel mode
ip access-list extended *<name>*	Creates an extended ACL used to protect traffic

permit ip host *ip-address* **host** *ip-address*	Defines the traffic to be protected
crypto map *<name> priority* **ipsec-isakmp**	Creates crypto map, assigns a priority, and specifies that IKE will be used to establish the IPsec SAs
match address *<crypto-acl>*	Specifies an extended ACL for a crypto map entry Note: The ACL defines the traffic to encrypt.
set transform-set *<name>*	Specifies which transform sets can be used with the crypto map entry
set peer *ip-address*	Specifies an IPsec peer in a crypto map entry
crypto map *<map-name>*	Specifies interface configuration mode; assigns crypto map to the interface
show crypto isakmp policy	Displays the parameters for each IKE policy
show crypto ipsec transform-set	Displays the configured transform sets
show crypto key mypubkey rsa	Displays the RSA public keys of a router
show crypto pki certificates	Displays information about your certificate, the CA certificate, and any RA certificates
show crypto map [**interface** *interface* | **tag** *map-name*]	Displays the crypto map configuration
show crypto isakmp sa	Displays the current IKE SAs
show crypto ipsec sa	Displays the settings used by the current SAs
show ip access-lists	Displays IP ACL entries
debug crypto ipsec	Displays IP IPsec events
debug crypto isakmp	Displays messages about IKE events

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will complete the lab exercise setup by resetting router defaults, ensuring connectivity with other routers in the lab, and establishing connectivity to the CA server.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student laptop is operating with the correct date and time.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2**. (where P = pod number).

Step 3 Restore the original course router configuration.

Step 4 Verify that you have connectivity with the peer pod router.

```
router# ping 172.30.Q.2
(where Q = peer pod number)
```

Step 5 Build a static route to the 172.26.26.0/24 network where the CA server is located.

```
router(config)# ip route 172.26.26.0 255.255.255.0 172.30.P.1
(where P = pod number)
```

Step 6 Ensure that you can connect to the CA server from your router.

```
router# ping 172.26.26.51
```

Step 7 Ensure that you can establish an HTTP session to the CA server. Test this capability from your Microsoft Windows 2000 Server by opening a web browser and entering the location: **http://172.26.26.51/**.

Activity Verification

You have completed this task when you attain these results:

- You can successfully ping the 172.26.26.51 address (CA server) and your peer pod router.

Task 2: Prepare for IPsec

In this task, you will prepare for configuring IPsec by determining the ISAKMP and IPsec policy, creating an ACL to allow IPsec traffic and verifying the time zone, date, and time on the router.

Activity Procedure

Complete these steps:

Step 1 Determine the ISAKMP and IPsec policy. In this lab exercise, you will use default values except when you are directed to enter a specific value.

- The ISAKMP policy is to use RSA signature keys.
- The IPsec policy is to use ESP mode with DES.
- The IPsec policy is to encrypt all traffic between specified subnetworks.

Step 2 Create an ACL to allow IPsec protocols on the outside interface.

```
router# configure terminal
router(config)# ip access-list extended 102
router(config-ext-nacl)# permit ahp host 172.30.P.2 host
172.30.Q.2
router(config-ext-nacl)# permit esp host 172.30.P.2 host
172.30.Q.2
router(config-ext-nacl)# permit udp host 172.30.P.2 host
172.30.Q.2 eq isakmp
router(config-ext-nacl)# permit udp host 172.30.P.2 host
172.30.Q.2 eq 4500
```

Step 3 Set the router time zone.

```
router(config)# clock timezone CST -6
```

Step 4 Set the router date and time.

```
router# clock set hh:mm:ss day month year
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show clock** and a **show ip access-lists** command. The output should be similar to this:

```
R1# show clock
23:21:24.007 CST Fri Sept 8 2006
R1# show ip access-lists
Extended IP access list 102
    10 permit ahp host 172.30.1.2 host 172.30.6.2
    20 permit esp host 172.30.1.2 host 172.30.6.2
    30 permit udp host 172.30.1.2 host 172.30.6.2 eq isakmp
    40 permit udp host 172.30.1.2 host 172.30.6.2 eq non500-isakmp
```


Task 3: Define the Router Host and Domain Name

In this task, you will give the router a hostname and define the router domain name. These will be used when generating your RSA key pairs and certificates.

Activity Procedure

Complete these steps:

Step 1 Give the router a hostname.

```
router(config)# hostname RP
(where P = pod number)
```

Step 2 Define the router domain name.

```
router(config)# ip domain-name cisco.com
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show run** command. The output should contain the following:

```
!
hostname R<P>
ip domain name cisco.com
!
```

Task 4: Define Hostname-to-IP Address Mapping

In this task, you will define the CA server static hostname-to-IP address mapping.

Activity Procedure

Complete these steps:

Step 1 Define the CA server static hostname-to-IP address mapping.

```
router(config)# ip host vpnca 172.26.26.51
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show run** command. The output should contain the following:

```
!
hostname R1
ip domain name cisco.com
ip host VPNCA 172.26.26.51
!
```

Task 5: Generate RSA Key Pairs

In this task, you will generate RSA keys.

Activity Procedure

Complete this step:

Step 1 Generate RSA keys.

```
router(config)# crypto key generate rsa
```

Note	Follow the router prompts to complete the task. Use **512** for the number of bits for the modulus.

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto key mypubkey rsa** command. The output should be similar to this:

```
R2# show crypto key mypubkey rsa
% Key pair was generated at: 08:27:16 CST Mar 8 2005
Key name: R2.cisco.com
 Usage: Signature Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D589C9 E077B874
  4E659CA9 8AFB7BCB 1AFB5534 6AFF4207 0B575271 543AC147 C34383AC F68FA0B0
  65153A9F 56725C8E D0BD5AA4 BB38A91D 3F10EC8D 8209FCB3 71020301 0001
% Key pair was generated at: 08:27:18 CST Mar 8 2005
Key name: R2.cisco.com
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00B732F0 6AE5F0A5
  0DAA23D7 86595EE0 A2ECDCB9 EEF0079E 8878DEC7 6F12F304 0F1D0FA8 E3313317
  ECD5521C F82962F5 41903C39 BC26A362 C03D8221 CEE2A7A6 A1020301 0001
% Key pair was generated at: 08:27:27 CST Mar 8 2005
Key name: R2.cisco.com.server
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00AFBE5F 651AE624
  F220E6BD 473A6643 9D24644E 5034F6EF D9B1DB4F E96DCB48 727997ED 46DFC45E
  2FAE67C0 78A82788 D4A27D12 A96E472B D178A7A9 9A23E3E8 60275C72 56603867
      0DF75F9E A682F959 14AA0E1E EB4D49BA 41A2D002 33CA2A1C AD020301 0001
```


Task 6: Configure the CA Server Trustpoint

In this task, you will configure the CA server trustpoint.

Activity Procedure

Complete these steps:

Step 1 Create a name for the CA and enter CA trustpoint mode.

```
router(config)# crypto pki trustpoint vpnca
```

Step 2 Specify the URL of the CA.

```
router(ca-trustpoint)# enrollment url http://vpnca
```

Step 3 Exit CA configuration mode.

```
router(ca-trustpoint)# exit
```

Step 4 Authenticate the CA server.

```
router(config)# crypto pki authenticate vpnca
```

```
You should see the following:
Certificate has the following attributes:
Fingerprint: 527D8DCA 4D52A047 C8DA1DAD D5368629
% Do you accept this certificate? [yes/no]: y
```

Step 5 Request your own certificate.

```
router(config)# crypto pki enroll vpnca
```

You should see the following:

```
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
      password to the CA Administrator in order to revoke your
certificate.
      For security reasons your password will not be saved in the
configuration.
      Please make a note of it.
Password: <mypassword>
Re-enter password: <mypassword>
% The subject name in the certificate will include: router1.cisco.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate vpnca verbose' command will show the
fingerprint.
*Jul 24 17:07:15.403: CRYPTO_PKI:  Certificate Request Fingerprint
MD5: D35C6688
 E6EBADEF 504EE6F2 BEC8FA13
```

```
*Jul 24 17:07:15.407: CRYPTO_PKI:  Certificate Request Fingerprint
SHA1: 1A45EA0

A 6725B055 E84018FB 9DE5DD88 4E1C2CF5

*Jul 24 17:07:19.915: %PKI-6-CERTRET: Certificate received from
Certificate Authority
```

Step 6 Save the keys and certificates to NVRAM.

```
router# copy system:running-config nvram:startup-config
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto pki certificates** command. The output should be similar to this:

```
router1# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 02
  Certificate Usage: General Purpose
  Issuer:
    cn=vpnca
  Subject:
    Name: router1.cisco.com
    hostname=router1.cisco.com
  Validity Date:
    start date: 10:06:21 CST Jul 24 2006
    end   date: 10:06:21 CST Jul 24 2007
  Associated Trustpoints: vpnca
  Storage: nvram:vpnca#6102.cer

  Certificate
    Status: Available
    Certificate Serial Number: 01
    Certificate Usage: Signature
    Issuer:
      cn=vpnca
    Subject:
      cn=vpnca
    Validity Date:
      start date: 09:33:21 CST Jul 24 2006
      end   date: 09:33:21 CST Jul 23 2009
    Associated Trustpoints: vpnca
    Storage: nvram:vpnca#6101CA.cer
```


Task 7: Configure an ISAKMP Policy to Use RSA Signatures

In this task, you will configure an ISAKMP policy to use RSA signatures.

Activity Procedure

Complete these steps:

Step 1 Verify that ISAKMP is enabled. You should see a default policy.

```
router# show crypto isakmp policy
```

Note	If you see the message "ISAKMP is turned off," complete Step 2, then complete the rest of the steps. If ISAKMP is already enabled, skip Step 2.

```
R1# show crypto isakmp policy
Global IKE policy
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 2 Enable IKE/ISAKMP on your router.

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

Step 3 Create the policy and specify the policy priority.

```
router(config)# crypto isakmp policy 110
```

Step 4 Specify authentication to use RSA signatures.

```
router(config-isakmp)# authentication rsa-sig
```

Step 5 Specify the IKE encryption.

```
router(config-isakmp)# encryption 3des
```

Step 6 Specify the DH group.

```
router(config-isakmp)# group 2
```

Step 7 Specify the hash algorithm.

```
router(config-isakmp)# hash md5
```

Step 8 Set the ISAKMP SA lifetime.

```
router(config-isakmp)# lifetime 36000
```

Step 9 Exit ISAKMP policy configuration mode.

```
router(config-isakmp)# exit
```

Step 10 Configure the pre-shared key and peer address.

```
router(config)# crypto isakmp key 0 cisco1234 address
172.30.Q.2
(where Q = peer pod number)
```

Step 11 Exit configuration mode.

```
router(config)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto isakmp policy** command. The output should be similar to this:

```
R1# show crypto isakmp policy
Global IKE policy
Protection suite of priority 110
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               36000 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
```


Task 8: Configure an IPsec Transform Set

In this task, you will configure a transform set.

Activity Procedure

Complete these steps:

Step 1 Define a transform set. Use the following parameters:

- Transform name = **SNRS**
- ESP protocols = **esp-des**
- Mode = **tunnel**

```
router(config)# crypto ipsec transform-set SNRS esp-des
```

Step 2 Set the mode to tunnel.

```
router(cfg-crypto-trans)# mode tunnel
```

Step 3 Exit crypto transform configuration mode.

```
router(cfg-crypto-trans)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto ipsec transform-set** command. The output should be similar to this:

```
router# show crypto ipsec transform-set
Transform set SNRS: { esp-des  }
   will negotiate = { Tunnel,  },
```

Task 9: Configure an IPsec Crypto ACL

In this task, you will create an ACL that "defines" traffic to protect. The ACL should encrypt traffic between the subnetworks that you specify. Use the following parameters:

- Traffic encrypted: Traffic between **10.0.P.0** and **10.0.Q.0**
- ACL number: **101**
- Protocol: **IP**

Activity Procedure

Complete these steps:

Step 1 Configure the crypto ACL.

```
router(config)# ip access-list extended 101
router(config-ext-nacl)# permit ip 10.0.P.0 0.0.0.255 10.0.Q.0
0.0.0.255
(where P = pod number, and Q = peer pod number)
```

Step 2 Exit ACL configuration mode.

```
router(config-ext-nacl)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show access-list** command. The output should be similar to this:

```
R1# show ip access-lists
Extended IP access list 101
    10 permit ip 10.0.1.0 0.0.0.255 10.0.6.0 0.0.0.255
Extended IP access list 102
    10 permit ahp host 172.30.1.2 host 172.30.6.2
    20 permit esp host 172.30.1.2 host 172.30.6.2
    30 permit udp host 172.30.1.2 host 172.30.6.2 eq isakmp
    40 permit udp host 172.30.1.2 host 172.30.6.2 eq non500-isakmp
```

Task 10: Configure an IPsec Crypto Map

In this task, you will configure a crypto map. Use the following parameters:

- Name of map: **SNRS-MAP**
- Priority of map: **10**
- Key exchange type: **isakmp**
- Peer: **172.30.Q.2**
- Transform set: **SNRS**
- Match address: **101**

Activity Procedure

Complete these steps:

Step 1 Set the name of the map, the map priority, and the type of key exchange to be used.

```
router(config)# crypto map SNRS-MAP 10 ipsec-isakmp
```

Step 2 Specify the extended ACL to use with this map.

```
router1(config-crypto-map)# match address 101
```

Step 3 Specify the transform set that you defined earlier.

```
router1(config-crypto-map)# set transform-set SNRS
```

Step 4 Specify the VPN peer using the hostname or IP address of the peer.

```
router(config-crypto-map)# set peer 172.30.Q.2
(where Q = peer pod number)
```

Step 5 Exit crypto map configuration mode.

```
router(config-crypto-map)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto map** command. The output should be similar to this:

```
R1# show crypto map
Crypto Map "SNRS-MAP" 10 ipsec-isakmp
        Peer = 172.30.6.2
        Extended IP access list 101
            access-list 101 permit ip 10.0.1.0 0.0.0.255 10.0.6.0
0.0.0.255
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                SNRS,
        }
        Interfaces using crypto map SNRS-MAP:
```

Task 11: Apply the Crypto Map to an Interface

In this task, you will apply the crypto map to an interface. Use the following parameters:

- Interface to configure: **FastEthernet 0/1**
- Crypto map to use: **SNRS-MAP**

Activity Procedure

Complete these steps:

Step 1 Access interface configuration mode.

```
router(config)# interface FastEthernet 0/1
```

Step 2 Assign a crypto map to the interface.

```
router(config-if)# crypto map SNRS-MAP
```

You should see the following message:

```
Jul 26 16:19:05.123: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

Step 3 Exit interface configuration mode.

```
router(config-if)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto map** or **show crypto map interface** command. The output should be similar to this:

```
R1# show crypto map interface fastEthernet 0/1
Crypto Map "SNRS-MAP" 10 ipsec-isakmp
        Peer = 172.30.6.2
        Extended IP access list 101
            access-list 101 permit ip 10.0.1.0 0.0.0.255 10.0.6.0
0.0.0.255
        Current peer: 172.30.6.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                SNRS,
        }
        Interfaces using crypto map SNRS-MAP:
                FastEthernet0/1
```


Task 12: Ensure That Encryption Is Working Between Routers

In this task, you will generate traffic from your internal subnet to your peer pod internal subnet to ensure that encryption is working between the routers.

Activity Procedure

Complete these steps:

Step 1 Generate interesting traffic using an extended ping. You will ping from the inside interface of your pod router to the inside interface of your peer pod router. You can also ping from your laptop to the laptop of your peer pod.

```
R1# ping
Protocol [ip]:
Target IP address: 10.0.6.2
Repeat count [5]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: yes
Source address or interface: 10.0.1.2
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 10.0.6.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Step 2 Display your ISAKMP SAs.

Step 3 Display your IPsec SAs

Activity Verification

You have completed this task when you attain these results:

- Verify that the IKE and IPsec SAs have been established.

```
R1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.6.2      172.30.1.2      QM_IDLE           1001    0 ACTIVE
```

```
IPv6 Crypto ISAKMP SA
R1# show crypto ipsec sa
interface: FastEthernet0/1
    Crypto map tag: SNRS-MAP, local addr 172.30.1.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (10.0.6.0/255.255.255.0/0/0)
   current_peer 172.30.6.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 6657, #pkts encrypt: 6657, #pkts digest: 6657
    #pkts decaps: 6656, #pkts decrypt: 6656, #pkts verify: 6656
    #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.: 172.30.1.2, remote crypto endpt.: 172.30.6.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x1B029B45(453155653)

     inbound esp sas:
      spi: 0xD74582A5(3611656869)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: FPGA:1, crypto map: SNRS-MAP
        sa timing: remaining key lifetime (k/sec): (4565588/2901)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x1B029B45(453155653)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: FPGA:2, crypto map: SNRS-MAP
```



```
            sa timing: remaining key lifetime (k/sec): (4565588/2871)
            IV size: 8 bytes
            replay detection support: N
            Status: ACTIVE

      outbound ah sas:

outbound pcp sas:
```

Lab 4-3: Configure a GRE Tunnel to a Remote Site

Complete this lab activity to practice what you learned in the related module.

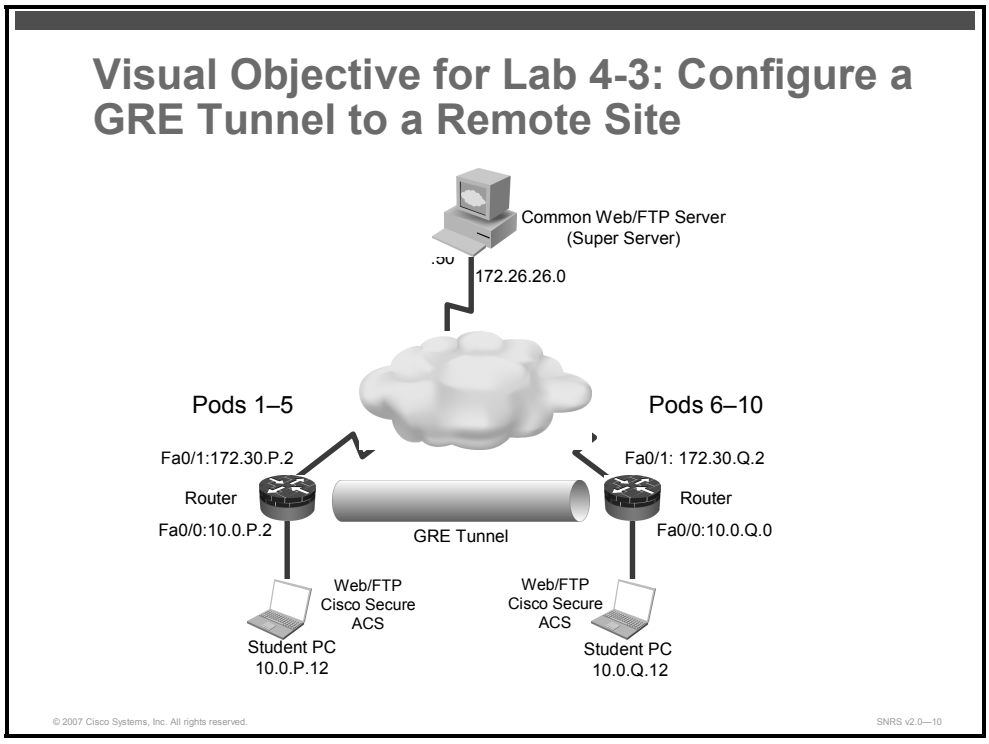
Activity Objective

In this activity, you will configure s Cisco perimeter router to use GRE tunnels. After completing this activity, you will be able to meet these objectives:

- Create a GRE tunnel and configure the source and destination addresses
- Configure GRE as the tunnel mode and bring up the interface
- Configure static routes
- Verify connectivity to a remote site

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers

Command List

The table describes the commands that are used in this activity.

GRE Commands

Command	Description
interface tunnel 0	Creates a tunnel and enters interface configuration mode
ip address *ip-address netmask*	Assigns an IP address to an interface
tunnel source *source-ip source-net-mask*	Specifies the tunnel interface source address and subnet mask
tunnel destination *dest-ip dest-net-mask*	Specifies the tunnel interface destination address
no shutdown	Brings up the tunnel interface
ip route *remote-network remote-mask* **tunnel** *number*	Configures a static route to a remote subnet through the tunnel
show ip interface brief	Views IP interface summary
show ip route	Displays routing information for a host or network
show interfaces tunnel *number*	Displays tunnel configuration
ping *ip-address*	Checks network connectivity

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will complete the lab exercise setup by resetting the router defaults and ensuring connectivity with the other routers in the lab.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student laptop is operating with the correct date and time.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2**. (where P = pod number).

Step 3 Remove the crypto map from the interface.

Step 4 Verify that you have connectivity with the peer pod router.

```
router# ping 172.30.Q.2
(where Q = peer pod number)
```

Activity Verification

You have completed this task when you attain these results:

- Your output should resemble the following:

```
router# ping 172.30.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Task 2: Configure the Tunnel Interface, Source, and Destination

In this task, you will create the tunnel and configure the GRE tunnel source and destination addresses.

Activity Procedure

Complete these steps:

Step 1 Specify a tunnel interface number and enter interface configuration mode.

```
router(config)# interface tunnel 0
```

Step 2 Configure an IP address and subnet mask on the tunnel interface.

Note	Both tunnel interfaces must be on the same subnet.

```
router(config-if)# ip address 172.PQ.1.P 255.255.255.0
(Where P = your pod, Q = remote pod)
```

Other Pod

```
router(config-if)# ip address 172.QP.1.Q 255.255.255.0
(Where P = your pod, Q = remote pod)
```

Step 3 Specify the tunnel interface source address and subnet mask.

```
router(config-if)# tunnel source 172.30.P.2
```

Note	This is your local outside interface.

Step 4 Specify the tunnel interface destination address.

```
router(config-if)# tunnel destination 172.30.Q.2 255.255.255.0
```

Activity Verification

You have completed this task when you attain these results:

- You will verify this activity after the next task.

Task 3: Bring Up the Tunnel Interface

In this task, you will bring up the tunnel interface.

Activity Procedure

Complete these steps:

Step 1 Bring up the tunnel interface.

```
router(config-if)# no shutdown
```

Step 2 Exit back to global configuration mode.

```
router(config-if)# exit
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show** commands should be similar to this:

```
router# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            10.0.1.2        YES NVRAM  up
up

FastEthernet0/1            172.30.1.2      YES NVRAM  up
up

Tunnel0                    172.16.1.1      YES manual up
up
```

```
router# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.1/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.30.1.2, destination 172.30.2.2
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
```

```
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Task 4: Configure a Route to a Remote Network Through a Tunnel

In this task, you will configure static routes to the remote site.

Activity Procedure

Complete these steps:

Step 1 Configure a static route to the remote site subnets.

```
router(config)# ip route 10.0.Q.0 255.255.255.0 Tunnel 0
```

Step 2 Exit to EXEC mode.

```
router(config)# exit
```

Activity Verification

You have completed this task when you attain these results:

- The output of the **show ip route** command should be similar to this.

```
router2# show ip route 10.0.6.0
Routing entry for 10.0.6.0/24
  Known via "static", distance 1, metric 0 (connected)
  Redistributing via eigrp 1
  Advertised by eigrp 1
  Routing Descriptor Blocks:
  * directly connected, via Tunnel0
      Route metric is 0, traffic share count is 1
```


Task 5: Verify the Tunnel

In this task, you will verify connectivity to the remote site.

Activity Procedure

Complete these steps:

Step 1 Ping the other side of the tunnel.

```
R1# ping 172.16.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.6, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4
ms
```

Step 2 Ping the remote subnet.

```
R1# ping 10.0.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4
ms.
```

Activity Verification

You have completed this task when you attain these results:

- Verify traffic on the tunnel by using the **show interfaces tunnel** command and checking if the counters increase.

```
R1# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.1/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.30.1.2, destination 172.30.6.2
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
```

```
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:03:34, output 00:03:34, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     145 packets input, 11500 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     50 packets output, 6200 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```


Lab 4-4: Configure a DMVPN

Complete this lab activity to practice what you learned in the related module.

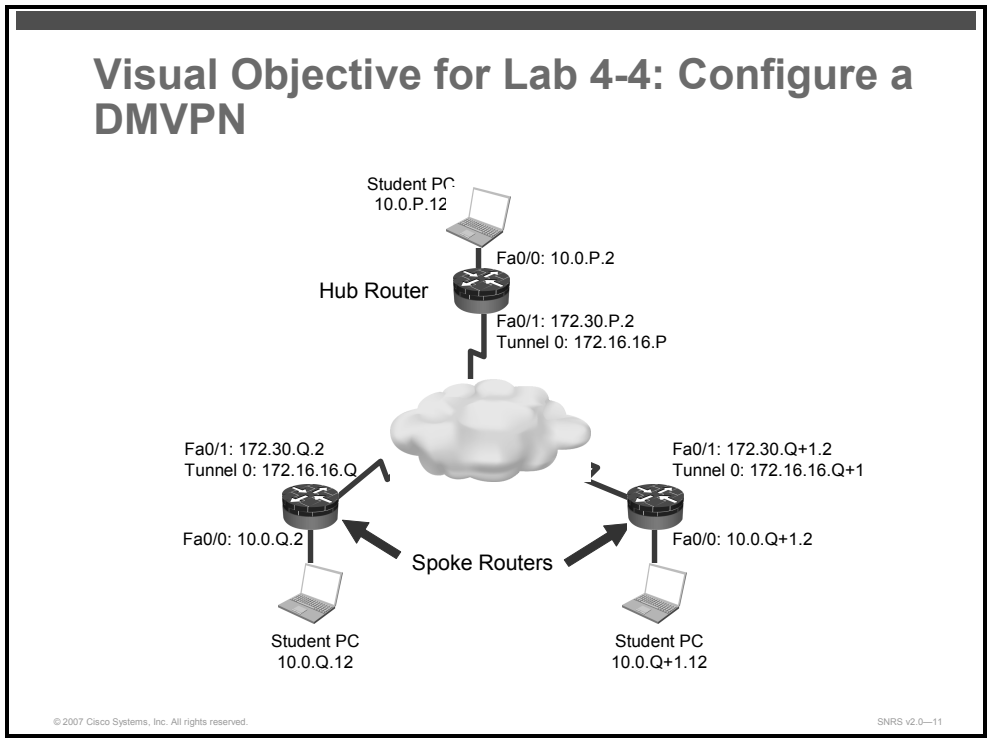
Activity Objective

In this activity, you will set up a DMVPN. After completing this activity, you will be able to meet these objectives:

- Set up lab devices
- Configure ISAKMP and IPsec policies to support a DMVPN
- Configure an IPsec profile
- Configure the hub router for mGRE and IPsec integration
- Configure the spoke routers for mGRE and IPsec integration
- Verify DMVPN operation

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers

Command List

The table describes the commands that are used in this activity.

DMVPN Commands

Command	Description
crypto ipsec profile *name*	Specifies the name of the IPsec profile and enters IPsec profile configuration mode
set transform-set *transform-set-name*	Specifies which transform sets can be used with the IPsec profile
interface tunnel *number*	Configures a tunnel interface and enters interface configuration mode
ip address *ip-address mask*	Sets a primary or secondary IP address for an interface
ip mtu *bytes*	Sets the MTU size, in bytes, of IP packets sent on an interface
ip nhrp authentication *string*	Configures the authentication string for an interface using NHRP
ip nhrp map multicast dynamic	Allows NHRP to automatically add spoke routers to the multicast NHRP mappings
ip nhrp network-id *number*	Enables NHRP on an interface
tunnel source {*ip-address* | *type number*}	Sets the source address for a tunnel interface
tunnel key *key-number*	Enables an ID key for a tunnel interface
tunnel mode gre multipoint	Sets the encapsulation mode to mGRE for the tunnel interface
tunnel protection ipsec profile *name*	Associates a tunnel interface with an IPsec profile
ip nhrp map *hub-tunnel-ip-address hub-physical-ip-address*	Statically configures the IP-to-NBMA address mapping of IP destinations connected to an NBMA network
ip nhrp map multicast *hub-physical-ip-address*	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub router
ip nhrp nhs *hub-tunnel-ip-address*	Configures the hub router as the NHRP next-hop server
show ip nhrp	Displays the NHRP cache
show crypto isakmp sa	Displays all current IKE SAs
show crypto ipsec sa	Displays the settings used by current SAs
show crypto map	Displays the crypto map configuration

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will complete the lab exercise setup by resetting the router defaults and ensuring connectivity with the other routers in the lab.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student laptop is operating with the correct date and time.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2**. (where P = pod number).

Step 3 Remove the crypto map from the interface.

Step 4 Verify that you have connectivity with the peer pod routers.

```
router# ping 172.30.Q.2
router# ping 172.30.Q+1.2
(where Q = peer pod number)
```

Activity Verification

You have completed this task when you attain these results:

- You can successfully ping the spoke routers.

Task 2: Configure ISAKMP and IPsec Policies on Routers

In this task, you will create ISAKMP and IPsec policies on all routers. You will configure your ISAKMP and IPsec policies just as you did with an IPsec site-to-site VPN using pre-shared keys.

Activity Procedure

Complete these steps:

Step 1 Set the policy priority and enter ISAKMP policy configuration mode.

```
router(config)# crypto isakmp policy 20
```

Step 2 Set authentication to use pre-shared keys.

```
router(config-isakmp)# authentication pre-share
```

Step 3 Set the hash algorithm.

```
router(config-isakmp)# hash md5
```

Step 4 Exit the ISAKMP policy configuration mode.

```
router(config-isakmp)# exit
```

Step 5 Exit configuration mode

Step 6 Create a transform set to use with the IPsec profile.

```
router(config)# crypto ipsec transform-set DMVPN-Transform
esp-des
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto isakmp policy** command and a **show crypto ipsec transform** command. Your output should be similar to this:

```
R1# show crypto isakmp policy
Global IKE policy
Protection suite of priority 20
        encryption algorithm:   DES - Data Encryption Standard (56
bit keys)
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
       Diffie-Hellman group:    #1 (768 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56
bit keys)
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit


router# show crypto ipsec transform-set
Transform set DMVPN-Transform: { esp-des  }
   will negotiate = { Tunnel,  },
```

Task 3: Configure an IPsec Profile

In this task, you will create an IPsec profile.

Activity Procedure

Complete these steps:

Step 1 Create a profile and enter IPsec profile configuration mode.

```
router(config)# crypto ipsec profile DMVPN
```

Step 2 Specify which transform sets can be used with the IPSec profile.

```
router(ipsec-profile)# set transform-set DMVPN-Transform
```


Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto ipsec profile** command. Your output should be similar to this:

```
router# show crypto ipsec profile
IPSEC profile DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
```

Task 4: Configure the Hub for DMVPN

In this task, you will configure the hub router for mGRE and IPsec integration.

Activity Procedure

Complete these steps:

Step 1 Configure the ISAKMP pre-shared key to accept multiple addresses.

```
router_hub(config)# crypto isakmp key 0 cisco123 address
0.0.0.0 0.0.0.0
```

Step 2 Configure a tunnel interface and enter interface configuration mode.

```
router_hub(config)# interface Tunnel 1
```

```
You should see the following:
```

```
*Jul 27 20:34:17.203: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Tunnel1, changed state to down
```

Step 3 Set a primary or secondary IP address for the tunnel interface.

```
router_hub(config-if)# ip address 172.16.H.H 255.255.255.0
```

```
(where H = hub pod number)
```

Step 4 (Optional) Set the MTU size, in bytes, of IP packets.

```
router_hub(config-if)# ip mtu 1416
```

Step 5 Change the EIGRP maximum hold time. It should not to exceed 7 times the EIGRP hello timer (35 seconds).

```
router_hub(config-if)# ip hold-time eigrp 1 35
```

Step 6 Disable eigrp next-hop-self.

```
router_hub(config-if)# no ip next-hop-self eigrp 1
```

Step 7 Turn off split horizon on the mGRE tunnel interface.

```
router_hub(config-if)# no ip split-horizon eigrp 1
```

Note	Otherwise, EIGRP will not advertise routes that are learned via the mGRE interface back out that interface.

Step 8 Configure the authentication string for an interface using NHRP.

```
router_hub(config-if)# ip nhrp authentication cisco123
```

Step 9 Allow NHRP to automatically add spoke routers to the multicast NHRP mappings.

```
router_hub(config-if)# ip nhrp map multicast dynamic
```

Step 10 Enable NHRP on the tunnel interface.

```
router_hub(config-if)# ip nhrp network-id 99
```

Step 11 Set a source address for the tunnel interface.

```
router_hub(config-if)# tunnel source FastEthernet 0/1
```


Step 12 Enable an ID key for the tunnel interface.

```
router_hub(config-if)# tunnel key 999
```

Step 13 Set the encapsulation mode to mGRE for the tunnel interface.

```
router_hub(config-if)# tunnel mode gre multipoint
```

You should see the following:

```
*Jul 27 20:45:27.199: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Tunnel0, changed state to up
```

Step 14 Associate the tunnel interface with an IPsec profile.

```
router_hub(config-if)# tunnel protection ipsec profile DMVPN
```

You should see the following:

```
*Jul 27 20:46:20.079: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

Step 15 Return to global configuration mode.

```
router_hub(config-if)# exit
```

Step 16 Enter EIGRP configuration mode.

```
router_hub(config)# router eigrp 1
```

Step 17 Specify networks to advertise.

```
router_hub(config-router)# network 10.0.P.0
router_hub(config-router)# network 172.16.0.0
router_hub(config-router)# no network 172.30.0.0
```

Step 18 Disable auto summarization.

```
router_hub(config-router)# no auto-summary
```

Step 19 Return to privileged EXEC mode.

```
router_hub(config-router)# exit
```

Step 20 Remove any static routes to spoke internal networks.

```
router_hub(config)# no ip route 10.0.Q.0 FastEthernet 0/1
router_hub(config)# no ip route 10.0.Q+1.0 FastEthernet 0/1
```

Step 21 Add static routes to spokes.

```
router_hub(config)# ip route 172.30.6.0 255.255.255.0
172.30.P.1
router_hub(config)# ip route 172.30.7.0 255.255.255.0
172.30.P.1
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto map** command. Your output should look like this:

```
router_hub# show crypto map
Crypto Map "MYMAP" 10 ipsec-isakmp
        Peer = 172.30.6.2
        Extended IP access list vpn
            access-list vpn permit ip host 172.30.1.2 host 172.30.6.2
        Current peer: 172.30.6.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
        Interfaces using crypto map MYMAP:

Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                DMVPN,
        }
        Interfaces using crypto map Tunnel0-head-0:
               Tunnel0
```


Task 5: Configure the Spokes for DMVPN

In this task, you will configure spoke routers for mGRE and IPsec integration.

Activity Procedure

Complete these steps:

Step 1 Configure the ISAKMP pre-shared key.

```
router_spoke(config)# crypto isakmp key 0 cisco123 address
0.0.0.0 0.0.0.0
(where H = hub pod number)
```

Step 2 Configure a tunnel interface and enter interface configuration mode.

```
router_spoke(config)# interface Tunnel 0
```

Step 3 Set a primary or secondary IP address for the tunnel interface.

```
router_spoke(config-if)# ip address 172.16.H.2 255.255.255.0
(where H = hub pod number)
```

Step 4 (Optional) Set the MTU size, in bytes, of IP packets.

```
router_spoke(config-if)# ip mtu 1416
```

Step 5 Change the EIGRP maximum hold time.

```
router_spoke(config-if)# ip hold-time eigrp 1 35
```

Step 6 Disable eigrp next-hop-self.

```
router_spoke(config-if)# no ip next-hop-self eigrp 1
```

Step 7 Disable split horizon.

```
router_spoke(config-if)# no ip split-horizon eigrp 1
```

Step 8 Configure the authentication string for an interface using NHRP.

```
router_spoke(config-if)# ip nhrp authentication cisco123
```

Step 9 Statically configure the IP-to-NBMA address mapping of an IP destination connected to an NBMA network.

```
router_spoke(config-if)# ip nhrp map 172.16.H.H 172.30.H.2
(where H = hub pod number)
```

Step 10 Enable the use of a dynamic routing protocol between the spoke and hub, and send multicast packets to the hub router.

```
router_spoke(config-if)# ip nhrp map multicast 172.30.H.2
(where H = hub pod number)
```

Step 11 Configure the hub router as the NHRP next-hop server.

```
router_spoke(config-if)# ip nhrp nhs 172.16.H.H
(where H = hub pod number)
```

Step 12 Enable NHRP on the interface.

```
router_spoke(config-if)# ip nhrp network-id 99
```

Step 13 Set the source address for the tunnel interface.

```
router_spoke(config-if)# tunnel source FastEthernet 0/1
```

Step 14 Enable an ID key for the tunnel interface.

```
router_spoke(config-if)# tunnel key 999
```

Step 15 Set the encapsulation mode to mGRE for the tunnel interface.

```
router_spoke(config-if)# tunnel mode gre multipoint
```

Step 16 Associates a tunnel interface with an IPsec profile.

```
router_spoke(config-if)# tunnel protection ipsec profile DMVPN
```

Step 17 Return to global configuration mode.

```
router_spoke(config-if)# exit
```

Step 18 Enter EIGRP configuration mode.

```
router_hub(config)# router eigrp 1
```

Step 19 Specify networks to advertise.

```
router_spoke(config-router)# network 10.0.Q.0
router_spoke(config-router)# network 172.16.0.0
router_spoke(config-router)# no network 172.30.0.0
```

Step 20 Disable auto summarization.

```
router_spoke(config-router)# no auto-summary
```

Step 21 Configure the router as a stub and to advertise connected networks.

```
router_spoke(config-router)# eigrp stub connected
```

Step 22 Return to privileged EXEC mode.

```
router_spoke(config-router)# exit
```

Step 23 Remove any static routes to other spokes or hubs.

```
router_spoke(config)# no ip route 10.0.Q.0
router_spoke(config)# no ip route 10.0.P+1.0
```

Step 24 Configure static routes to other pods.

```
router_spoke(config)# ip route 172.30.Q.0 255.255.255.0
172.30.P.1
router_spoke(config)# ip route 172.30.P+1.0 255.255.255.0
172.30.P.1
```


Activity Verification

You have completed this task when you attain these results:

- Issue a **show crypto map** command. Your output should look like this:

```
router_spoke# show crypto map
Crypto Map "MYMAP" 10 ipsec-isakmp
        Peer = 172.30.1.2
        Extended IP access list vpn
            access-list vpn permit ip host 172.30.1.2 host 172.30.6.2
        Current peer: 172.30.6.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                DMVPN,
        }
        Interfaces using crypto map MYMAP:

Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.6.2
        Extended IP access list
            access-list  permit gre host 172.30.1.2 host 172.30.6.2
        Current peer: 172.30.1.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                DMVPN,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
```

Task 5: Test and Verify

In this task, you will verify that the DMVPN feature is working.

Activity Procedure

Complete these steps:

Step 1 Perform an extended ping from the internal interface of one spoke router to the internal interface of the other spoke router.

```
R6#ping
Protocol [ip]:
Target IP address: 10.0.7.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.0.6.2
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 10.0.7.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Step 2 Display the crypto map configuration.

```
router# show crypto map
```

Step 3 Display the current IKE SAs.

Step 4

```
router# show crypto isakmp sa
```

Step 5 Display the settings used by the current SAs.

```
router# show crypto ipsec sa
```

Step 6 Display the NHRP cache.

```
router# show ip nhrp
```


Activity Verification

You have completed this task when you attain these results:

- Issue the commands listed in the Activity Procedure section. Your results should be similar to what follows.

On the Hub Router

Before pinging the spoke routers, your output should look like this:

```
hub# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65539 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.1.5
        Extended IP access list
            access-list  permit gre host 172.30.1.2 host 172.30.6.2
        Current peer: 172.30.1.5
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65540 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.6.2
        Extended IP access list
            access-list  permit gre host 172.30.1.2 host 172.30.6.2
        Current peer: 172.30.6.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
```

```
hub# show ip nhrp
172.16.16.6/32 via 172.16.16.6, Tunnel0 created 01:12:15, expire
01:27:44
  Type: dynamic, Flags: unique nat registered
  NBMA address: 172.30.1.5
172.16.16.7/32 via 172.16.16.7, Tunnel0 created 00:55:34, expire
01:44:25
  Type: dynamic, Flags: unique registered
  NBMA address: 172.30.1.6

hub# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.1.2      172.30.6.2      QM_IDLE           1003    0 ACTIVE
172.30.1.2      172.30.7.2      QM_IDLE           1004    0 ACTIVE

IPv6 Crypto ISAKMP SA

hub# show crypto ipsec sa
interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 172.30.1.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.2/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.6/255.255.255.255/47/0)
   current_peer 172.30.1.6 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest: 3
    #pkts decaps: 3, #pkts decrypt: 3, #pkts verify: 3
    #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.: 172.30.1.2, remote crypto endpt.: 172.30.1.6
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x6B4D9B3F(1800248127)

     inbound esp sas:
```



```
      spi: 0xBDBA0F87(3183087495)
        transform: esp-des ,
```

On the Spoke1 Router

```
spoke1# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.1.2
        Extended IP access list
            access-list  permit gre host 172.30.1.5 host 172.30.1.2
        Current peer: 172.30.1.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0

spoke1# show ip nhrp
172.16.16.1/32 via 172.16.16.1, Tunnel0 created 01:18:26, never expire
  Type: static, Flags: nat used
  NBMA address: 172.30.1.2

spoke1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.1.2      172.30.1.5      QM_IDLE           1003    0 ACTIVE

IPv6 Crypto ISAKMP SA

spoke1# show crypto ipsec sa
```

```
Interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 172.30.1.5


   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.5/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.2/255.255.255.255/47/0)
   current_peer 172.30.1.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 23, #pkts encrypt: 23, #pkts digest: 23
    #pkts decaps: 21, #pkts decrypt: 21, #pkts verify: 21
    #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.: 172.30.1.5, remote crypto endpt.: 172.30.1.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x26E1DFA(40771066)

     inbound esp sas:
      spi: 0x13F1E21C(334619164)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2011, flow_id: FPGA:11, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4554551/2336)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x26E1DFA(40771066)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2012, flow_id: FPGA:12, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4554551/2311)
```

```
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

    outbound ah sas:

    outbound pcp sas:
```

After Ping from Spoke2

```
spoke1# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.1.2
        Extended IP access list
            access-list  permit gre host 172.30.1.5 host 172.30.1.2
        Current peer: 172.30.1.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65538 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.1.6
        Extended IP access list
            access-list  permit gre host 172.30.1.5 host 172.30.1.6
        Current peer: 172.30.1.6
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
```

```
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0


spoke1# show ip nhrp
172.16.16.1/32 via 172.16.16.1, Tunnel0 created 01:32:20, never expire
  Type: static, Flags: nat used
  NBMA address: 172.30.1.2
172.16.16.6/32 via 172.16.16.6, Tunnel0 created 00:06:52, expire
01:53:07
  Type: dynamic, Flags: router unique nat local
  NBMA address: 172.30.1.5
    (no-socket)
172.16.16.7/32 via 172.16.16.7, Tunnel0 created 00:06:53, expire
01:53:07
  Type: dynamic, Flags: router implicit
  NBMA address: 172.30.1.6


spoke1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.1.6      172.30.1.5      QM_IDLE           1005    0 ACTIVE
172.30.1.2      172.30.1.5      QM_IDLE           1003    0 ACTIVE


IPv6 Crypto ISAKMP SA


spoke1# show crypto ipsec sa
interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 172.30.1.5


   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.5/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.6/255.255.255.255/47/0)
   current_peer 172.30.1.6 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 1, #pkts encrypt: 1, #pkts digest: 1
    #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.: 172.30.1.5, remote crypto endpt.: 172.30.1.6
      path mtu 1500, ip mtu 1500
      current outbound spi: 0xE937D794(3912750996)

      inbound esp sas:
       spi: 0x42C40F9B(1120145307)
         transform: esp-des ,
         in use settings ={Tunnel, }
         conn id: 2013, flow_id: FPGA:13, crypto map: Tunnel0-head-0
         sa timing: remaining key lifetime (k/sec): (4579214/3120)
         IV size: 8 bytes
         replay detection support: N
         Status: ACTIVE

      inbound ah sas:

      inbound pcp sas:

      outbound esp sas:
       spi: 0xE937D794(3912750996)
         transform: esp-des ,
         in use settings ={Tunnel, }
         conn id: 2014, flow_id: FPGA:14, crypto map: Tunnel0-head-0
         sa timing: remaining key lifetime (k/sec): (4579213/3109)
         IV size: 8 bytes
         replay detection support: N
         Status: ACTIVE

      outbound ah sas:
      outbound pcp sas:

   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.5/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.2/255.255.255.255/47/0)
   current_peer 172.30.1.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 29, #pkts encrypt: 29, #pkts digest: 29
    #pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
```

```
    #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.: 172.30.1.5, remote crypto endpt.: 172.30.1.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x26E1DFA(40771066)

     inbound esp sas:
      spi: 0x13F1E21C(334619164)
        transform: esp-des ,
        in use settings ={Tunnel, }
                conn id: 2011, flow_id: FPGA:11, crypto map: Tunnel0-
head-0
        sa timing: remaining key lifetime (k/sec): (4554549/1467)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x26E1DFA(40771066)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2012, flow_id: FPGA:12, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4554549/1459)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:

spoke1# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
```

```
  Internet address is 172.16.16.6/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.30.1.5 (FastEthernet0/1), destination UNKNOWN
  Tunnel protocol/transport multi-GRE/IP
    Key 0x3E7, sequencing disabled
    Checksumming of packets disabled

  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "DMVPN")
  Last input 00:09:16, output 00:09:15, output hang never
  Last clearing of "show interface" counters 00:14:02
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
       6 packets input, 776 bytes, 0 no buffer
       Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
       0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       6 packets output, 804 bytes, 0 underruns
       0 output errors, 0 collisions, 0 interface resets
       0 output buffer failures, 0 output buffers swapped out
```

On the Spoke2 Router

Before pinging other the pods, your output should look like this:

```
spoke2# show crypto map
spoke2#show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.1.2
        Extended IP access list
            access-list  permit gre host 172.30.1.6 host 172.30.1.2
        Current peer: 172.30.1.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0

spoke2# show ip nhrp
172.16.16.1/32 via 172.16.16.1, Tunnel0 created 00:03:26, never expire
  Type: static, Flags: authoritative used
  NBMA address: 172.30.1.2

spoke2# show crypto isakmp sa
spoke2#show crypto isakmp sa
dst             src             state          conn-id slot status
172.30.1.2      172.30.1.6      QM_IDLE              3    0 ACTIVE

spoke2# show crypto ipsec sa
interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 172.30.1.6
```



```
   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.6/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.2/255.255.255.255/47/0)
   current_peer 172.30.1.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest: 3
    #pkts decaps: 3, #pkts decrypt: 3, #pkts verify: 3
    #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.: 172.30.1.6, remote crypto endpt.: 172.30.1.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0xBDBA0F87(3183087495)

     inbound esp sas:
      spi: 0x6B4D9B3F(1800248127)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 3002, flow_id: FPGA:2, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4585714/964)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xBDBA0F87(3183087495)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 3003, flow_id: FPGA:3, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4585714/946)
        IV size: 8 bytes
        replay detection support: N
```

```
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:

spoke2# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.16.7/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.30.1.6 (FastEthernet0/1), destination UNKNOWN
  Tunnel protocol/transport multi-GRE/IP, key 0x3E7, sequencing
disabled
  Checksumming of packets disabled,  fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "DMVPN")
  Last input 00:06:09, output 00:06:09, output hang never
  Last clearing of "show interface" counters 00:00:10
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```


After Pings to Spoke1

```
spoke2# ping 172.16.16.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.16.6, timeout is 2 seconds:
!!!!!

spoke2# show ip nhrp
172.16.16.1/32 via 172.16.16.1, Tunnel0 created 01:08:52, never expire
  Type: static, Flags: authoritative used
  NBMA address: 172.30.1.2
172.16.16.6/32 via 172.16.16.6, Tunnel0 created 00:00:06, expire
01:59:54
  Type: dynamic, Flags: router
  NBMA address: 172.30.1.5

spoke2# show crypto isakmp sa
dst             src             state          conn-id slot status
172.30.1.2      172.30.1.6      QM_IDLE              3    0 ACTIVE
172.30.1.6      172.30.1.5      QM_IDLE              4    0 ACTIVE

spoke2# show crypto ipsec sa
interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 172.30.1.6

   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.6/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.2/255.255.255.255/47/0)
   current_peer 172.30.1.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 9, #pkts encrypt: 9, #pkts digest: 9
    #pkts decaps: 8, #pkts decrypt: 8, #pkts verify: 8
    #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.: 172.30.1.6, remote crypto endpt.: 172.30.1.2
     path mtu 1500, ip mtu 1500
```

```
      current outbound spi: 0x14077AE8(336034536)

      inbound esp sas:
       spi: 0x304A295A(810166618)
         transform: esp-des ,
         in use settings ={Tunnel, }
         conn id: 3004, flow_id: FPGA:4, crypto map: Tunnel0-head-0
         sa timing: remaining key lifetime (k/sec): (4397274/2869)
         IV size: 8 bytes
         replay detection support: N
         Status: ACTIVE

      inbound ah sas:

      inbound pcp sas:

      outbound esp sas:
       spi: 0x14077AE8(336034536)
         transform: esp-des ,
         in use settings ={Tunnel, }
         conn id: 3001, flow_id: FPGA:1, crypto map: Tunnel0-head-0
         sa timing: remaining key lifetime (k/sec): (4397274/2843)
         IV size: 8 bytes
         replay detection support: N
         Status: ACTIVE

      outbound ah sas:
      outbound pcp sas:

   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.6/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.5/255.255.255.255/47/0)
   current_peer 172.30.1.5 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 1, #pkts decrypt: 1, #pkts verify: 1
    #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.: 172.30.1.6, remote crypto endpt.: 172.30.1.5
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x42C40F9B(1120145307)

     inbound esp sas:
      spi: 0xE937D794(3912750996)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 3003, flow_id: FPGA:3, crypto map: Tunnel0-head-0
        conn id: 3003, flow_id: FPGA:3, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4402655/3483)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x42C40F9B(1120145307)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 3002, flow_id: FPGA:2, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4402656/3473)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:

spoke2# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.16.7/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
```

```
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.30.1.6 (FastEthernet0/1), destination UNKNOWN
  Tunnel protocol/transport multi-GRE/IP, key 0x3E7, sequencing
disabled
  Checksumming of packets disabled,  fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "DMVPN")
  Last input 00:02:11, output 00:02:11, output hang never
  Last clearing of "show interface" counters 00:36:12
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     7 packets input, 940 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     7 packets output, 864 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```


Lab 4-5: Configure a Cisco IOS SSL VPN (WebVPN)

Complete this lab activity to practice what you learned in the related module.

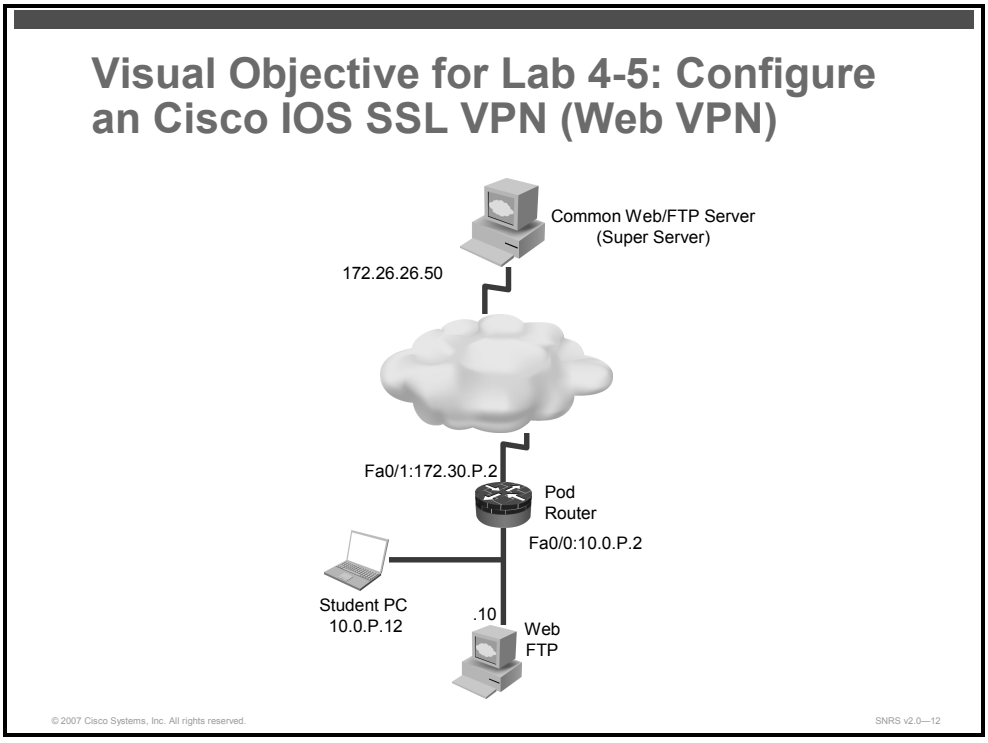
Activity Objective

In this activity, you will configure a Cisco router for Cisco IOS SSL VPN clientless access. After completing this activity, you will be able to meet these objectives:

- Set up lab devices
- Configure AAA for WebVPN
- Configure DNS for WebVPN
- Configure certificates and trustpoints for WebVPN
- Configure a WebVPN gateway
- Configure a WebVPN context
- Verify WebVPN operation

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers
- External web server (Super Server)

Command List

The table describes the commands that are used in this activity.

WebVPN Commands

Command	Description
username *name* **password 0** *password*	Create a user and password in the local database.
aaa new-model	Enable AAA
aaa authentication login default local	Specifies the default authentication method.
ip domain name *name*	Specifies a domain name to be used with its certificate
ip host *host-name ip-address*	Defines static hostname-to-address mappings
webvpn gateway *gateway-name*	Creates the WebVPN gateway and enter SSLVPN gateway configuration mode
hostname *name*	Specifies the hostname for the WebVPN gateway
http-redirect	Configures HTTP traffic to be carried over secure HTTPS
ip address *ip-address* **port** *port-number*	Configures a proxy address and port number for HTTPS
ssl trustpoint *trustpoint-name*	Specifies a trust point
inservice	Puts the WebVPN gateway into service
webvpn context *context-name*	Creates a webvpn context and enters context configuration mode.
gateway *gateway-name*	Associates a WebVPN gateway with this WebVPN context.
login-message *"string"*	Configures a message for the user login text box displayed on the login page.
title "*title***"**	Configures the HTML title string.
url-list "*list-name***"**	Creates a URL list and enters URL list configuration mode.
heading "*string***"**	Configures the heading that is displayed above URLs listed on the Portal page.
url-text "*string***" url-value "***url***"**	Adds an entry to the URL list.
port-forward *port-list-name*	Names a port- forwarding list and enter Cisco IOS SSL VPN port-forward list configuration mode.

local-port *port-number* **remote-server** *FQDN* **remote-port** *port-number* **description "***string***"**	Remaps (forwards) application port numbers in the port-forwarding list.
policy group *group-name*	Entesr Group Policy Configuration mode
url-list *string*	Attaches a URL list to this policy group configuration
port-forward *port-list-name*	Attaches a port- forwarding list to this policy group configuration
banner "*string***"**	Configures a banner to be displayed after a successful login.
timeout idle *seconds*	Configures remote user session idle time.
timeout session *seconds*	Configures the total length of time that a session can remain connected.
default-group-policy *policy-name*	Associates a group policy with the WebVPN context configuration.
inservice	Puts the WebVPN context into service.
show webvpn gateway *<name>*	Displays WebVPN gateway information.
show webvpn context *<name>*	Displays WebVPN context information.
show webvpn session context *context-name*	Displays WebVPN session information
show webvpn session user *username* **context all**	Displays WebVPN user session information.

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will set up the lab devices.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student laptop is operating with the correct date and time.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2**. (where P = pod number).

Step 3 Check connectivity to router.

```
C:>\ping 10.0.P.2
(Where P = Pod number)
```

Step 4 Check connectivity to Super Server.

```
C:>\ping 172.26.26.50
```

Activity Verification

You have completed this task when you attain these results:

- You have a successful ping to the router and to the Super Server.

```
C:\>ping 10.0.1.2

Pinging 10.0.1.2 with 32 bytes of data:

Reply from 10.0.1.2: bytes=32 time<1ms TTL=255
Reply from 10.0.1.2: bytes=32 time<1ms TTL=255
Reply from 10.0.1.2: bytes=32 time<1ms TTL=255
Reply from 10.0.1.2: bytes=32 time<1ms TTL=255

Ping statistics for 10.0.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```


Task 2: Configure AAA

In this task, you will configure AAA parameters to work with WebVPN.

Activity Procedure

Complete these steps:

Step 1 Populate the local user database.

```
router(config)# username user1 password 0 user1
```

Step 2 Enable AAA.

```
router(config)# aaa new-model
```

Step 3 Specify local AAA authentication.

```
router(config)#aaa authentication login default local
```

Activity Verification

You have completed this task when you attain these results:

- Execute a **show running-config** command. The output should include these statements:

```
Router#show running-config
!
aaa new-model
!
aaa authentication login default local
!
username user1 password 0 user1
```

Task 3: Configure DNS

In this task, you will configure DNS parameters to work with WebVPN.

Use the table to populate the router host table.

Host	Domain	IP Address
home	Cisco.com	10.0.P.12
superserver	Cisco.com	172.26.26.50

Activity Procedure

Complete these steps:

Step 1 Make sure that the router has a hostname.

Step 2 Define a default domain name.

```
router(config)#ip domain name cisco.com
```

Step 3 Define the static hostname-to-address mappings on the router.

```
router(config)# ip host home.cisco.com 10.0.P.12
router(config)# ip host superserver.cisco.com 172.26.26.50
```

Activity Verification

You have completed this task when you attain these results:

- Execute a **show running-config** command. The output should include these statements:

```
router#show running-config
!
ip domain name cisco.com
ip host vpnca 172.30.1.5
ip host home.cisco.com 10.0.1.12
ip host superserver.cisco.com 172.26.26.50
!
```


Task 4: Verify a Self-Signed Certificate

In this task, you will ensure that the router has a self-signed certificate.

Note	A self-signed certificate is automatically generated when a WebVPN gateway is put in service.

Activity Procedure

Complete this step:

Step 1 Check to see if the self-signed certificate is already on the router.

```
router#show running-config
```

If a certificate exists, the output should look like this:

```
!
crypto pki trustpoint TP-self-signed-1898720763
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1898720763
 revocation-check none
 rsakeypair TP-self-signed-1898720763
!
!
crypto pki certificate chain TP-self-signed-1898720763
 certificate self-signed 01 nvram:IOS-Self-Sig#3301.cer
```

Activity Verification

You have completed this task when you attain these results:

- You should be able to see the self-signed certificate in Step 1 above.

Task 5: Configure a WebVPN Gateway

In this task, you will configure the WebVPN virtual gateway.

Activity Procedure

Complete these steps:

Step 1 Name the gateway and enter Cisco IOS SSL VPN gateway configuration mode.

```
router(config)#webvpn gateway SNRS-GW
```

Step 2 Specify the hostname for the WebVPN gateway.

```
router(config-webvpn-gateway)#hostname GW-1
```

Step 3 Configure HTTP traffic to be carried over HTTPS.

```
router(config-webvpn-gateway)#http-redirect
```

Step 4 Configure a proxy IP address for the WebVPN gateway.

```
router(config-webvpn-gateway)#ip address 10.0.P.2 port 443
```

Step 5 (Optional) Configure the certificate trustpoint for the WebVPN gateway.

```
router(config-webvpn-gateway)# ssl trustpoint TP-self-signed-
1898720763
```

Note The name of the self-signed certificate is automatically inserted into the configuration file when the gateway is put in service.

Step 6 Put the WebVPN virtual gateway into service.

```
router(config-webvpn-gateway)#inservice
```

Activity Verification

You have completed this task when you attain these results:

- Execute a **show webvpn gateway** command and a **show webvpn gateway <*name*>** command. The output should resemble the following:

```
router#show webvpn gateway
Gateway Name                        Admin  Operation
------------                        -----  ---------
SNRS-GW                             up     up


router#show webvpn gateway SNRS-GW
Admin Status: up
Operation Status: up
IP: 10.0.1.2, port: 443
HTTP Redirect port: 80
SSL Trustpoint: TP-self-signed-1898720763
Mangling Hostame: GW-1
```


Task 6: Configure a WebVPN Context

In this task, you will configure a WebVPN context.

Activity Procedure

Complete these steps:

Step 1 Name the context and enter Cisco IOS SSL VPN configuration mode.

```
router(config)#webvpn context SSLVPN
```

Step 2 Associate a WebVPN gateway with this WebVPN context.

```
router(config-webvpn-context)#gateway SNRS-GW
```

Step 3 Configure a message for the User Login text box displayed on the Login page.

```
router(config-webvpn-context)#login-message "Please enter your
credentials"
```

Step 4 Configure the HTML title string.

```
router(config-webvpn-context)#title "SNRS WebVPN Page"
```

Configure a URL List

Complete these steps to create a URL list:

Step 1 Enter URL list configuration mode.

```
router(config-webvpn-context)# url-list "MYLINKS"
```

Step 2 Configure the heading that is displayed above URLs listed on the Portal page.

```
router(config-webvpn-url)#heading "Quicklinks"
```

Step 3 Add an entry to the URL list.

```
router(config-webvpn-url)#url-text "Pod Homepage" url-value
“home.cisco.com”
router(config-webvpn-url)#url-text "Super Server" url-value
"superserver.cisco.com"
```

Step 4 Exit back to WebVPN context configuration mode.

```
router(config-webvpn-url)#exit
```

Configure Thin-Client Mode

Complete these steps to configure the thin-client mode of operation:

Step 1 Enter Cisco IOS SSL VPN configuration mode.

```
router(config)#webvpn context SSLVPN
```

Step 2 Name a port-forwarding list and enter Cisco IOS SSL VPN port-forward list configuration mode.

```
router(config-webvpn-context)# port-forward Portlist
```

Step 3 Remap (forward) application port numbers in the port-forwarding list.

```
router(config-webvpn-port-fwd)# local-port 30020 remote-server
mail.corporate.com remote-port 25 description "SMTP"
router(config-webvpn-port-fwd)# local-port 30021 remote-server
mail.corporate.com remote-port 110 description "POP3"
router(config-webvpn-port-fwd)# local-port 30022 remote-server
mail.corporate.com remote-port 143 description "IMAP"
```

Step 4 Exit Cisco IOS SSL VPN port-forward list configuration mode.

```
router(config-webvpn-port-fwd)# exit
```

Configure a Policy Group

Complete these steps to configure a policy group:

Step 1 Enter group policy configuration mode.

```
router(config-webvpn-context)# policy group SSL-Policy
```

Step 2 Attach a URL list to this policy group configuration.

```
router(config-webvpn-group)# url-list MYLINKS
```

Step 3 Attach a port-forwarding list to this policy group configuration.

```
router(config-webvpn-group)# port-forward Portlist
```

Step 4 Configure a banner to be displayed after a successful login.

```
router(config-webvpn-group)#banner "Login Successful"
```

Step 5 Configure remote user session idle time and the total length of time that a session can remain connected.

```
router(config-webvpn-group)# timeout idle 1800
router(config-webvpn-group)# timeout session 36000
```

Step 6 Exit back to WebVPN context configuration mode.

```
router(config-webvpn-group)#exit
```

Step 7 Associate a group policy with the WebVPN context configuration.

```
router(config-webvpn-context)# default-group-policy SSL-Policy
```

Step 8 Put the WebVPN context into service.

```
router(config-webvpn-context)# inservice
```

Activity Verification

You have completed this task when you attain these results:

- Execute a **show webvpn context** command and a **show webvpn context <*name*>** command. The output should resemble the following:

```
router#show webvpn context
Codes: AS - Admin Status, OS - Operation Status
       VHost - Virtual Host


Context Name        Gateway  Domain/VHost      VRF      AS    OS
```



```
------------        -------  ------------      -------  ----  --------
Default_context     n/a      n/a               n/a      down  down
SSLVPN              SNRS-GW  -                 -        up    up

router#show webvpn context SSLVPN
Admin Status: up
Operation Status: up
CSD Status: Disabled
Certificate authentication type: All attributes (like CRL) are
verified
AAA Authentication List not configured
AAA Authentication Domain not configured
Default Group Policy: SSL-Policy
Associated WebVPN Gateway: SNRS-GW
Domain Name and Virtual Host not configured
Maximum Users Allowed: 10000 (default)
NAT Address not configured
VRF Name not configured
```

Task 7: Verify WebVPN

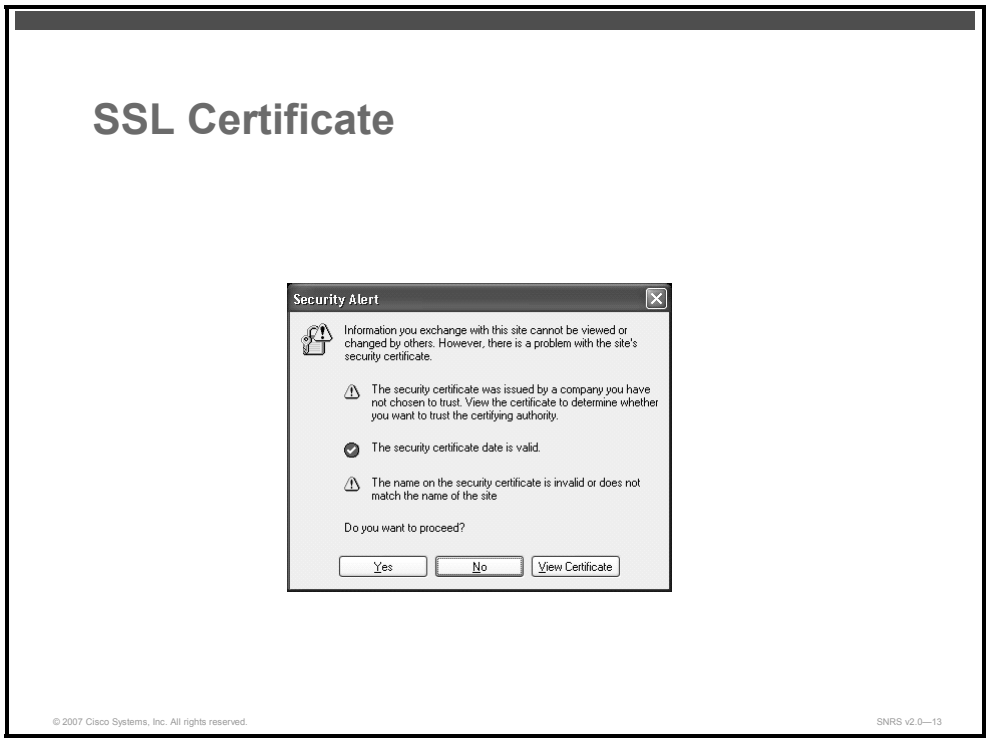
In this task, you will verify WebVPN configuration and operation.

Activity Procedure

Complete these steps:

Step 1 Point your browser to the address that you assigned the virtual gateway. The HTTP session should be redirected to HTTPS and the certificate dialog box should appear.

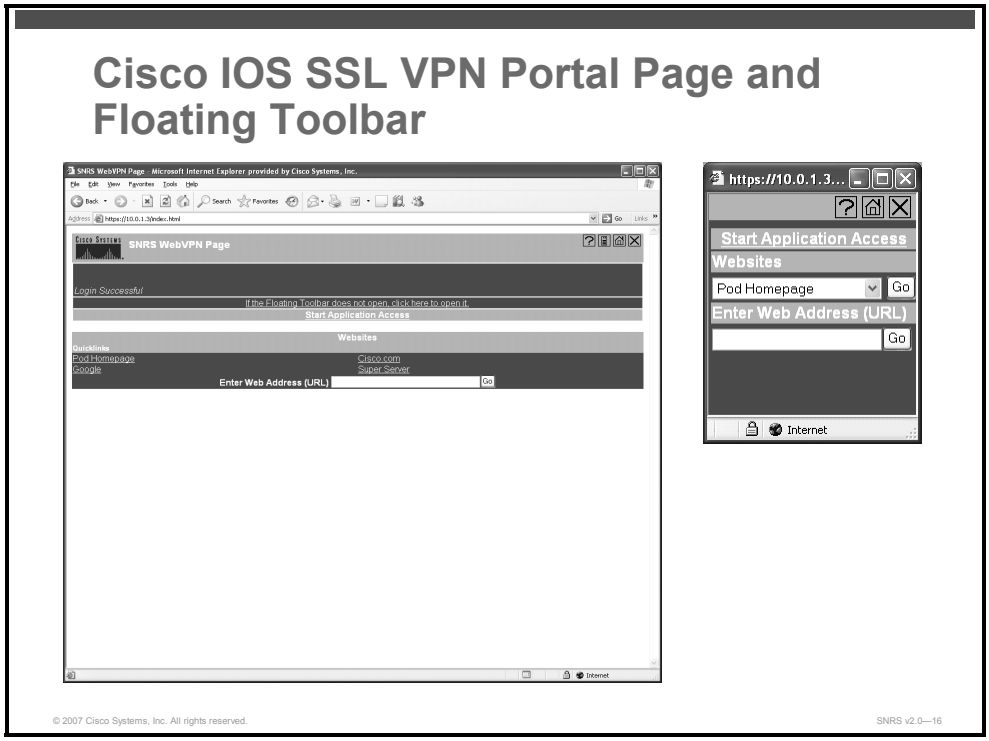
```
http:\\10.0.P.2
```



Step 2 Click **Yes** to proceed. The user login screen should appear.

Step 3 Input a valid username and password. The Login Successful dialog box should appear.

Step 4 Click **OK**. The main portal page and floating toolbar should appear.



Step 5 Click the **Pod Homepage** or **Super Server** links under the Websites section. The web pages should appear.

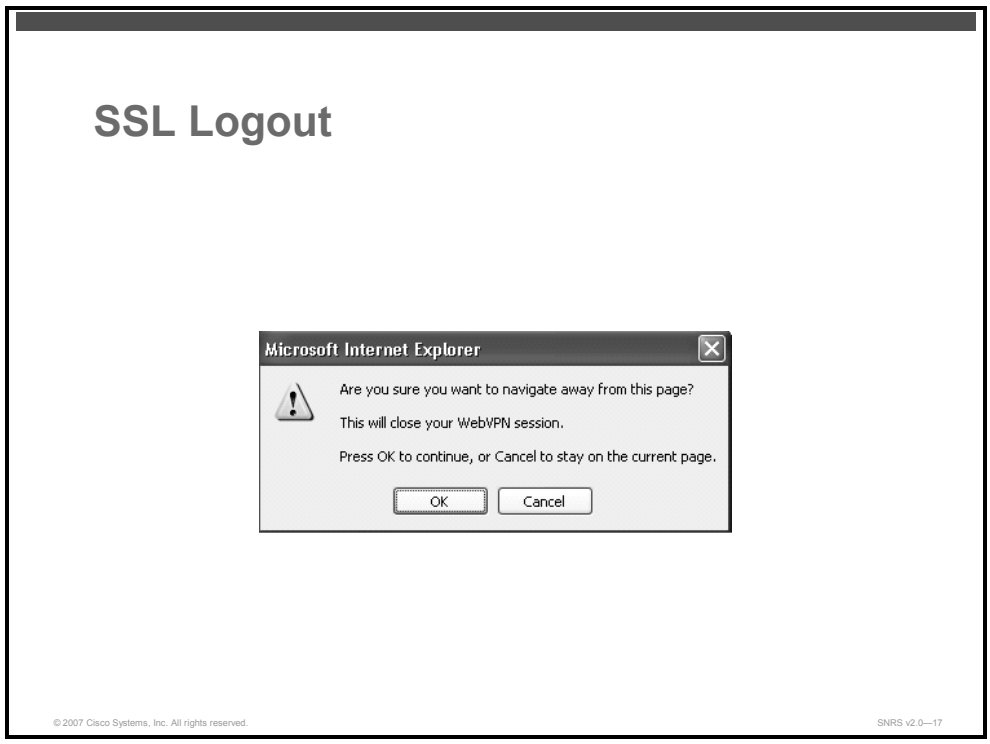
Step 6 Display session context information on the router.

```
router#show webvpn session context SSLVPN
```

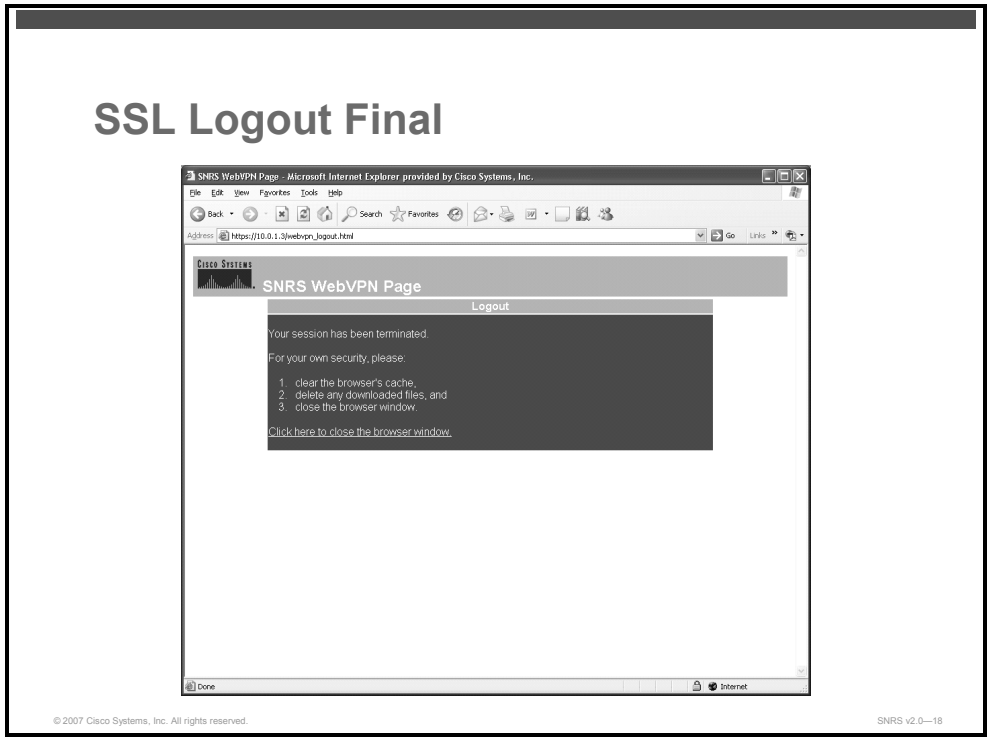
Step 7 Display session user information.

```
router#show webvpn session user user1 context all
```

Step 8 Click the Close icon of either the main portal page or the floating toolbar. You should see a prompt to make sure that you want to close the session.



Step 9 Click **OK**. The WebVPN logout page should appear.



Step 10 Click **Click Here to Close the Browser Window**. The browser window should close.

Activity Verification

You have completed this task when you attain these results:

- You should be able to do the following:
  - Log into the portal
  - Browse to different sites under the Websites section
  - Log out of the portal
- When you execute the **show webvpn session** commands, the output should be similar to this:

```
router#show webvpn session context SSLVPN
WebVPN context name: SSLVPN
Client_Login_Name  Client_IP_Address  No_of_Connections  Created
Last_Used
user1              10.0.1.5                   2         00:00:43  00:00:41
```

```
router#show webvpn session user user1 context all
WebVPN user name = user1 ; IP address = 10.0.1.5 ; context = SSLVPN
    No of connections: 1
    Created 00:01:27, Last-used 00:01:25
    Client Port: 1042
    User Policy Parameters
      Group name = SSL-Policy
    Group Policy Parameters
      banner = "Login Successful"
      url list name = "ACCESS"
      idle timeout = 2100 sec
      session timeout = 43200 sec
      port forward name =
      functions =
      citrix disabled
      dpd client timeout = 300 sec
      dpd gateway timeout = 300 sec
      keep sslvpn client installed = disabled
      rekey interval = 3600 sec
      rekey method =
      lease duration = 43200 sec
```


Lab 4-6: Configure Cisco Easy VPN Remote Access

Complete this lab activity to practice what you learned in the related module.

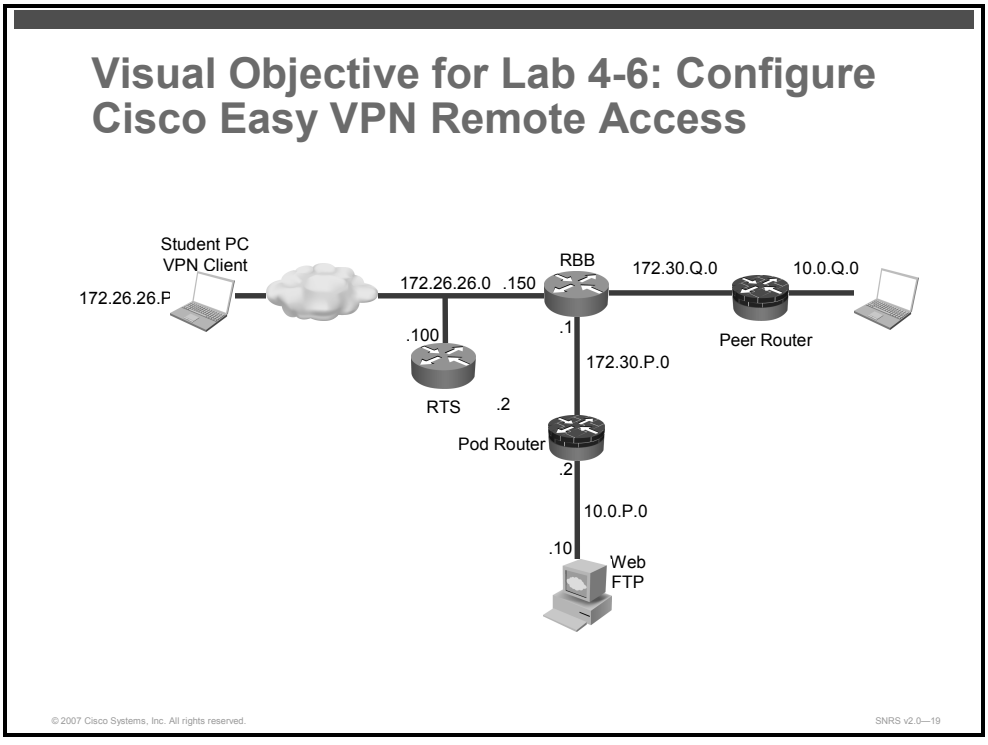
Activity Objective

In this activity, you will configure a Cisco router for Cisco Easy VPN Remote access. After completing this activity, you will be able to meet these objectives:

- Configure a router as a Cisco Easy VPN Server
- Configure Cisco Easy VPN Client on a laptop
- Configure a router as a Cisco Easy VPN Client
- Verify Cisco Easy VPN operation

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Pod routers
- Student laptops

Command List

The table describes the commands that are used in this activity.

Cisco Easy VPN Commands

Command	Description
username *cisco* **password 0** *cisco*	Creates a username and password in the local database.
aaa new-model	Enables AAA.
aaa authorization {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*]	To set parameters that restrict user access to a network.
authentication {rsa-sig | **rsa-encr** | **pre-share}**	Specifies the authentication method within an IKE policy.
crypto dynamic-map *dynamic-map-name dynamic-seq-num*	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
crypto isakmp client configuration group {*group-name* | **default**}	Specifies which group's policy profile will be defined.
crypto isakmp enable	Globally enables IKE.
crypto isakmp keepalive *secs* [*retries*]	Allows the gateway to send DPD messages to the peer.
crypto isakmp key *key-string* **address** *peer-address* [*mask*] [**no-xauth**]	Configures a pre-shared authentication key.
crypto isakmp policy *priority*	Defines an IKE policy.
domain *name*	Specify the DNS domain to which a group belongs.
encryption {**des** | **3des** | **aes** | **aes 192** | **aes 256**}	Specify the encryption algorithm within an IKE policy.
group {**1** | **2**}	Specifies the Diffie-Hellman group identifier within an IKE policy.
hash {**sha** | **md5**}	Specifies the hash algorithm within an IKE policy.
ip local pool {**default** | *poolname*} [*low-ip-address* [*high-ip-address*]] [**group** *group-name*] [**cache-size** ***size***]	Configures a local pool of IP addresses to be used when a remote peer connects to a point-to-point interface.
key *name*	Specifies the IKE pre-shared key for group policy attribute definition.
lifetime *seconds*	Specifies the lifetime of an IKE SA.
pool *name*	Defines a local pool address.

reverse-route [**remote-peer** [*ip-address*]]	Creates a source proxy information for a crypto map entry.
set transform-set *transform-set-name* [*transform-set-name2...transform-set-name6*]	Specifies which transform sets can be used with the crypto map entry.
crypto isakmp xauth timeout *sec*	Specifies the amount of time, in seconds, that the user has to enter the appropriate username and password to authenticate the VPN session.
ip dhcp pool *name*	Creates a name for the DHCP server address pool and places you in DHCP pool configuration mode.
network *network-number* [*mask* | */prefix-length*]	Specifies the subnet network number and mask of the DHCP address pool.
default-router *address* [*address2 ... address8*]	Specifies the IP address of the default router for a DHCP client.
ip dhcp excluded-address *low-address* [*high-address*]	Specifies the IP addresses that the DHCP server should not assign to DHCP clients.
crypto ipsec client ezvpn *name*	Creates a Cisco Easy VPN Remote configuration and then enters the Cisco Easy VPN Remote configuration mode.
group *group-name* **key** *group-key*	Specifies the group name and key value for the VPN connection.
peer {*ipaddress* | *hostname*}	Sets the peer IP address or host name for the VPN connection. A host name can be specified only when the router has a DNS server available for host name resolution.
mode {**client** | **network-extension**}	Specifies the mode of operation of the VPN of the router.
crypto ipsec client ezvpn xauth *name*	Responds to a pending VPN authorization request.
show crypto ipsec client ezvpn	Display the Cisco Easy VPN Remote configuration.

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will complete the lab exercise setup by resetting the router defaults and ensuring connectivity with the other routers in the lab.

In this task, you will assign the student laptop an IP address of 172.26.26.X to act as an XAUTH client for authentication.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student laptop is operating with the correct date and time.

Step 2 Configure your student PC for IP address **172.26.26.12** with a default gateway of **172.26.26.150**.

Step 3 Verify that you have connectivity with the peer pod routers.

```
C:\> ping 172.30.Q.2
```

Activity Verification

You have completed this task when you attain these results:

- You can successfully ping your 172.26.26.150 gateway.

Task 2: Configure a Router as a Cisco Easy VPN Server

In this task, you will configure a router to act as a Cisco Easy VPN Server.

Activity Procedure

Complete these steps:

Step 1 Create a local IP address pool named Remote-Pool with an IP address range of **10.0.P.32** to **10.0.P.64**.

```
router(config)# ip local pool Remote-Pool 10.0.P.100
10.0.P.150
```

Step 2 Configure a local username of **cisco**, and a password of **cisco** for an account accessing the perimeter router.

```
router(config)# username cisco password 0 cisco123
```

Note The **aaa new-model** command (used in Task 3) causes the local username and password on the router to be used in the absence of other AAA statements. It is important to create a known local username and password combination to prevent you from being locked out of the router.

Enable Policy Lookup

Step 3 Enable AAA.

```
router(config)# aaa new-model
```


Step 4 Create a group called "vpn-group" to be used for local AAA authorization and policy lookup for remote clients.

```
router(config)# aaa authorization network vpn-group local
```

Create an ISAKMP Policy for Remote Client Access

Step 5 Enable ISAKMP.

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

Step 6 Create ISAKMP policy 10.

```
router(config)# crypto isakmp policy 10
```

Step 7 Configure ISAKMP policy 10 to use pre-shared keys for authentication.

```
(config-isakmp)# authentication pre-share
```

Step 8 Configure ISAKMP policy 10 to use 3DES encryption.

```
router(config-isakmp)# encryption 3des
```

Step 9 Configure ISAKMP policy 10 to use DH group 2.

```
router(config-isakmp)# group 2
```

Step 10 Return to privileged EXEC mode.

```
router(config-isakmp)# end
```

Step 11 Verify your ISAKMP policy.

```
R1# show crypto isakmp policy
```

```
R1# show crypto isakmp policy

Global IKE policy
Protection suite of priority 10
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56 bit
keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit
```

Define Group Policy Information for a Mode Configuration Push

Step 12 Specify which group policy profile will be defined and enter ISAKMP group configuration mode. If no specific group matches and if a default group is defined, users will automatically be given the policy of the default group. For this lab exercise, use a group name of R6.

```
router(config)# crypto isakmp client configuration group R6
```

Step 13 Specify the ISAKMP pre-shared key for group policy attribute definition. Note that this command must be enabled if the VPN client identifies itself with a pre-shared key. For this lab exercise, use a key name of VPNKEY.

```
router(config-isakmp-group)# key VPNKEY
```

Step 14 Specify the domain name to be pushed to the client. For this lab exercise, use a domain name of **cisco.com.**

```
router(config-isakmp-group)# domain cisco.com
```

Step 15 Choose a local IP address pool. Note that this command must refer to a valid local IP address pool or the VPN client connection will fail. For this lab exercise, use the Remote-Pool pool name you created earlier.

```
router(config-isakmp-group)# pool Remote-Pool
```

Step 16 Return to global configuration mode.

```
router(config-isakmp-group)# exit
```

Create a Transform Set

Step 17 Create a transform set.

```
router(config)# crypto ipsec transform-set VPNTRANSFORM esp-
3des esp-sha-hmac
```

Step 18 Return to privileged EXEC mode.

```
router(cfg-crypto-trans)# end
```

Step 19 Verify your transform set configuration.

```
router# show crypto ipsec transform-set
R1# show crypto ipsec transform-set
Transform set VPNTRANSFORM: { esp-3des esp-sha-hmac  }
   will negotiate = { Tunnel,  },
```

Create a Dynamic Crypto Map

You will create a dynamic crypto map to handle remote-access traffic for the perimeter router.

Step 20 Create dynamic crypto map, Dynamic-Map, and enter the crypto map configuration mode.

```
router(config)# crypto dynamic-map Dynamic-Map 10
```

Step 21 Assign a transform set to Dynamic-Map.

```
router(config-crypto-map)# set transform-set VPNTRANSFORM
```


Step 22 Enable RRI.

```
router(config-crypto-map)# reverse-route
```

Step 23 Return to privileged EXEC mode.

```
router(config-crypto-map)# end
```

Step 24 Verify your dynamic map.

```
router# show crypto dynamic-map
```

```
R1# show crypto dynamic-map
Crypto Map Template"Dynamic-Map" 10
        No matching address list set.
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                VPNTRANSFORM,
        }
```

Apply Mode Configuration

You will apply mode configuration to a crypto map. Mode configuration must be applied to a crypto map to be enforced. Use the commands shown to apply mode configuration to a crypto map.

Step 25 Configure the router to initiate or reply to mode configuration requests.

```
router(config)# crypto map ClientMap client configuration
address respond
```

Step 26 Enable ISAKMP querying for group policy when requested by the VPN client.

```
router(config)# crypto map ClientMap isakmp authorization list
vpn-group
```

Step 27 Apply the dynamic crypto map to this crypto map.

```
router(config)# crypto map ClientMap 65535 ipsec-isakmp
dynamic Dynamic-Map
```

Apply Crypto Map to Interface

Step 28 Enter interface configuration mode.

```
router(config)# interface fastEthernet 0/1
```

Step 29 Assign the ClientMap crypto map to the interface.

```
router(config-if)# crypto map ClientMap
```

Step 30 Return to privileged EXEC mode.

```
router(config-if)# end
```

Step 31 Verify your crypto map configuration.

```
router# show crypto map
```

```
R1# show crypto map
Crypto Map "ClientMap" 65535 ipsec-isakmp
```

```
Dynamic map template tag: Dynamic-Map
Interfaces using crypto map ClientMap:
        FastEthernet0/1
```

Enable DPD

Step 32 Enable keepalives for DPD. The *20* value specifies the number of seconds between DPD messages (the range is between 10 and 3600 seconds); the *10* value specifies the number of seconds between retries if DPD messages fail (the range is between 2 and 60 seconds).

```
router(config)# crypto isakmp keepalive 20 10
```

Step 33 Exit global configuration mode.

```
router(config)# exit
```

Step 34 Save the router configuration.

```
router# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Activity Verification

You have completed this task when you attain these results:

- Use the various **show** commands from the steps to check your configuration.

Task 3: Configure a Router as a Cisco Easy VPN Client

In this task, you will configure a router as a Cisco Easy VPN remote client.

Activity Procedure

Complete these steps:

Step 1 Create a remote configuration and enter Cisco Easy VPN Remote configuration mode.

```
R6(config)# crypto ipsec client ezvpn R6-Client
```

Step 2 Specify the IPsec group and IPsec key values to be associated with this profile.

```
R6(config-crypto-ezvpn)# group R6 key VPNKEY
```

Step 3 Specify the IP address or hostname for the destination peer.

```
R6(config-crypto-ezvpn)# peer 172.30.Q.2
```

Step 4 Specify the type of VPN connection that should be made.

```
R6(config-crypto-ezvpn)# mode client
```

Step 5 Specify automatic connections.

```
R6(config-crypto-ezvpn)# connect auto
```

Step 6 Return to privileged EXEC mode.

```
R6(config-crypto-ezvpn)# end
```

Step 7 Access interface configuration mode.

```
R6(config)# interface FastEthernet 0/1
```

Step 8 Assign the client profile to the outside interface.

```
R6(config-if)# crypto ipsec client ezvpn R6-Client
```

Step 9 Change to inside interface.

```
R6(config-if)# exit
R6(config)# interface FastEthernet 0/0
```

Step 10 Assign an inside interface.

```
R6(config-if)# crypto ipsec client ezvpn R6-Client inside
```

Step 11 Return to privileged EXEC mode.

```
R6(config-if)# end
```

Step 12 Save your configuration.

```
R6# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Activity Verification

You have completed this task when you attain these results:

- Issue various **show** commands as with other VPN scenarios. The output should be similar to this:

```
R6# show crypto ipsec client ezvpn
Easy VPN Remote Phase: 6
Tunnel name : R6-Client
Inside interface list: FastEthernet0/0
Outside interface: FastEthernet0/1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.0.1.100
Mask: 255.255.255.255
Default Domain: cisco.com
Save Password: Allowed
Current EzVPN Peer: 172.30.1.2

R6# show crypto session
Crypto session current status
Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.30.1.2 port 500
  IKE SA: local 172.30.6.2/500 remote 172.30.1.2/500 Active
  IPSEC FLOW: permit ip host 10.0.1.100 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map

R6# show crypto session  detail
Crypto session current status
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, X - IKE Extended Authentication

Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.30.1.2 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 172.30.1.2
      Desc: (none)
  IKE SA: local 172.30.6.2/500 remote 172.30.1.2/500 Active
          Capabilities:C connid:0 lifetime:23:43:26
  IPSEC FLOW: permit ip host 10.0.1.100 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 4377612/2647
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 4377612/2647
```


Task 4: Configure Cisco Easy VPN Client on a Laptop

In this task, you will install the Cisco VPN client on a client laptop.

Activity Procedure

Complete these steps:

Step 1 Open the **CiscoApps** desktop folder.

Step 2 Open the **Cisco VPN Client** folder.

Step 3 Locate and run the Cisco VPN Client **setup.exe** executable. If this is the first time that the Cisco VPN Client is being installed, a window opens and displays the following message: "Do you want the installer to disable the IPsec policy agent?"

Step 4 Click **Yes** to disable the IPsec policy agent. The Welcome window opens.

Step 5 Read the Welcome window and click **Next**. The License Agreement window opens.

Step 6 Read the license agreement and click **Yes**. The Choose Destination Location window opens.

Step 7 Click **Next**. The Select Program Folder window opens.

Step 8 Accept the defaults by clicking **Next**. The Start Copying Files window opens.

Step 9 The files are copied to the hard disk drive of the student PC and the InstallShield Wizard Complete window opens.

Step 10 Choose **Yes, I Want to Restart My Computer Now** and click **Finish**. The student PC restarts.

Create a New Connection Entry

Step 11 Choose **Start > Programs > Cisco Systems VPN Client > VPN Client**. The Cisco Systems VPN Client window opens.

Step 12 Click the **New** icon. The Create New VPN Connection Entry window opens.

Step 13 Enter **VPN Server** in the connection entry field.

Step 14 Enter a perimeter router outside interface IP address of **172.30.P.2** in the host field (where P = pod number).

Step 15 Choose **Group Authentication** and complete the following fields (the entries are always case-sensitive):

Step 16 Enter a group name: **R6**. This is the group that you created earlier on the perimeter router.

Step 17 Enter the group password: **VPNKEY**. This is the key that you created earlier for the "vpn-group" group.

Step 18 Confirm the password: **VPNKEY**.

Step 19 Click **Save.**

Launch the Cisco VPN Client and Test Connectivity

You can now launch the VPN client and test connectivity.

Step 20 Choose **Start > Programs > Cisco Systems VPN Client > VPN Client**. The Cisco VPN Client should be launched.

Step 21 Click **Connect**. The Connection History window opens and several messages flash by quickly; the window closes and a Cisco VPN Dialer icon appears in the system tray.

Step 22 Right-click the **Cisco VPN Client** icon in the student PC system tray and choose the **Statistics** option.

Step 23 Open a command prompt shell and ping the inside interface of the perimeter router.

```
C:\> ping 10.0.P.2
(where P = pod number)
```

Step 24 Close the command prompt shell.

Activity Verification

You have completed this task when you attain these results:

- You can successfully connect using the VPN client.

Task 5: (Optional) Configure XAUTH

In this task, you will add XAUTH to the existing Cisco Easy VPN Server configuration.

Activity Procedure

Complete these steps:

Step 1 Enable AAA login authentication for the local **vpn-users** user group.

```
router(config)# aaa authentication login vpn-users local
```

Step 2 Set the timeout value (0 to 60 seconds) for the amount of time that the remote user has to enter a username and password on the client. Use **20** seconds for the timeout value for this lab exercise.

```
router(config)# crypto isakmp xauth timeout 20
```

Step 3 Enable IKE XAUTH for the ClientMap dynamic crypto map using the **vpn-users** user group.

```
router(config)# crypto map ClientMap client authentication
list vpn-users
```

Step 4 Exit global configuration mode.

```
router(config)# exit
```

Step 5 Save the router configuration to the startup configuration file.

```
router# copy running-config starting-config
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show running-config** command. The output should be similar to this:

```
router# show run
Your configuration should look similar to the following. Bold items
are associated with extended authentication:
!
aaa new-model
!
aaa authentication login vpn-users local
aaa authorization network vpn-group local
!
username cisco password 0 cisco
!
crypto isakmp policy 10
 encr 3des
 authentication pre-share
 group 2
crypto isakmp keepalive 20 10
```

```
crypto isakmp xauth timeout 60
!
crypto isakmp client configuration group R6
 key VPNKEY
 domain cisco.com
 pool Remote-Pool
!
crypto ipsec transform-set VPNTRANSFORM esp-3des esp-sha-hmac
!
crypto dynamic-map Dynamic-Map 10
 set transform-set VPNTRANSFORM
 reverse-route
!
crypto map CLIENTMAP client authentication list VPNUSERS
crypto map CLIENTMAP isakmp authorization list vpn-group
crypto map CLIENTMAP client configuration address respond
crypto map CLIENTMAP 65535 ipsec-isakmp dynamic Dynamic-Map
!
interface Ethernet0/1
 ip address 172.30.P.2 255.255.255.0
 half-duplex
 crypto map DYNMAP
!
ip local pool Remote-Pool 10.0.P.32 10.0.P.64
ip http server
```


Task 6: (Optional) Test XAUTH

In this task, you will test the XAUTH configuration of the Cisco Easy VPN Server.

Activity Procedure

Complete these steps:

Step 1 Open the Cisco VPN Dialer application by choosing **Start > Programs > Cisco Systems VPN Client > VPN Client**.

Step 2 Ensure that the Cisco Easy VPN Server connection entry is selected and that the IP address of your Cisco Easy VPN Server appears in the Remote Server field.

Step 3 Click **Connect**. If XAUTH is working correctly, the User Authentication for the Easy VPN Server window should appear.

Step 4 Enter a username of **cisco**.

Step 5 Enter a password of **cisco123**.

Step 6 Click **OK**. The Cisco VPN Client icon should appear in the system tray of the student PC.

Step 7 Check the status of the VPN connection by right-clicking the **Cisco VPN Client** icon in the student PC system tray and choosing **Status** and the **Statistics** tab.

Step 8 With the Status window still open, open a command shell and establish a Telnet session to the Cisco Easy VPN Server. You should see the encrypted and decrypted counters of the packets increment.

Activity Verification

You have completed this task when you attain these results:

- You can connect successfully using the Cisco VPN Client.

Lab 5-1: Configure Cisco IOS Classic Firewall

Complete this lab activity to practice what you learned in the related module.

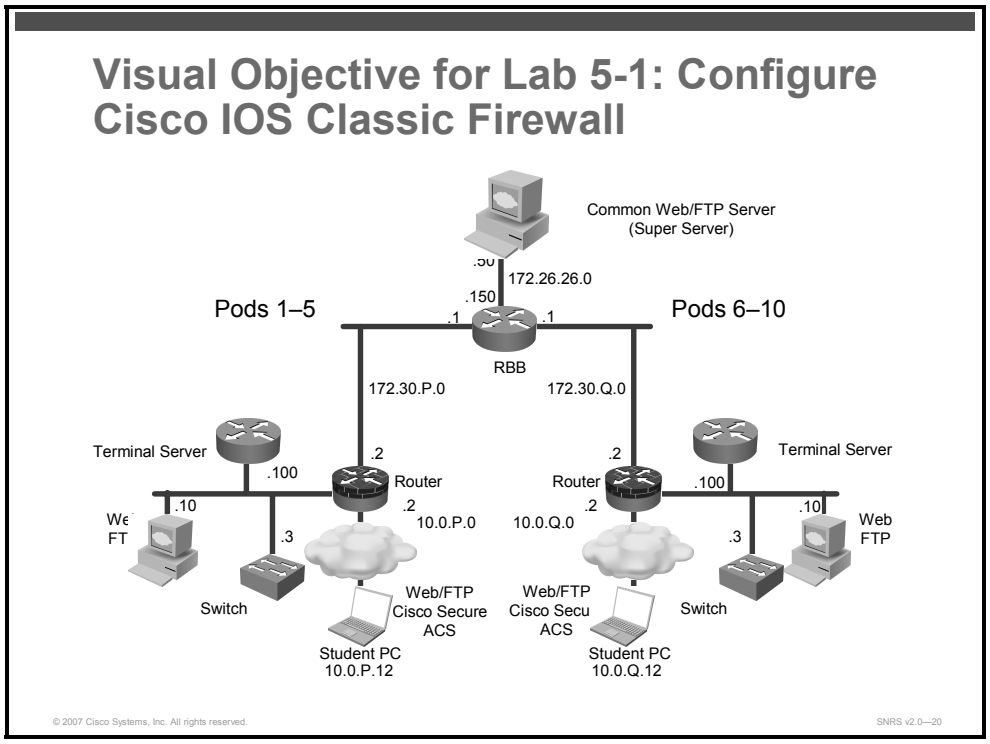
Activity Objective

In this activity, you will configure Cisco IOS classic firewall on a Cisco router. After completing this activity, you will be able to meet these objectives:

- Set up lab devices
- Define inspection rules for use with Cisco IOS classic firewall
- Apply inspection rules to an interface
- Configure logging and enable audit trails
- Test and verify Cisco IOS classic firewall operation

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student computers
- Pod routers

Command List

The table describes the commands that are used in this activity.

IOS Firewall Commands

Command	Description
access-list *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]	Defines an extended IP ACL, use the extended version of the **access-list** command in global configuration mode
ip access-group {*access-list-number* | *access-list-name*}{**in** | **out**}	Controls access to an interface.
ip inspect *inspection-name* {**in** | **out**}	Applies a set of inspection rules to an interface.
ip inspect audit trail	Enables Cisco IOS Classic Firewall audit trail messages, which will be displayed on the console after each Cisco IOS Classic Firewall session closes.
ip inspect name *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]	Defines a set of inspection rules.
line [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]	Identifies a specific line for configuration and enter line configuration collection mode.
logging console	Send syslog messages to all available tty lines and limit messages based on severity.
logging console [*severity-level*]	Enable logging of system messages.
ping [*protocol*] [**tag**] {*host-name* | *system-address*}	Diagnose basic network connectivity on AppleTalk, ATM, CLNS, DECnet, IP, Novell IPX, or SRB networks.
show access-lists [*access-list-number* | *access-list-name*]	Display the contents of current ACLs.
show ip inspect {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}	Display Cisco IOS Classic Firewall configuration and session information.

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will complete the lab exercise setup.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student PC is powered on and that the Microsoft Windows 2000 Server is operational. Your instructor will provide you with the correct username and password to log in to the student PC.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2** (where P = pod number).

Step 3 Make sure that your student PC has an appropriate syslog server application installed (for example, the Kiwi Syslog Daemon).

Step 4 Reload your perimeter router using the default lab configuration.

Step 5 Ensure that you can ping the peer router and network hosts before beginning.

Step 6 Make sure that your router is running the correct date and time.

Step 7 Make sure that your student PC is running the correct date and time.

Activity Verification

You have completed this task when you attain these results:

- You can ping the pod router and have checked that the date and time are correct.

Task 2: Define Inspection Rules and ACLs

In this task, you will define inspection rules and ACLs.

Activity Procedure

Complete these steps:

Step 1 Enter global configuration mode on your perimeter router.

Step 2 Define a CBAC rule to inspect all TCP and FTP traffic.

```
router(config)# ip inspect name FWRULE http timeout 300
router(config)# ip inspect name FWRULE ftp timeout 300
router(config)# ip inspect name FWRULE icmp timeout 300
```

Step 3 Define the inside interface ACL to allow outbound ICMP traffic and application traffic (FTP and World Wide Web). Block all other inside-initiated traffic.

```
router(config)# access-list 103 permit icmp any any
router(config)# access-list 103 permit tcp 10.0.P.0 0.0.0.255
any eq telnet
router(config)# access-list 103 permit tcp 10.0.P.0 0.0.0.255
any eq ftp
router(config)# access-list 103 permit tcp 10.0.P.0 0.0.0.255
any eq www
router(config)# access-list 103 deny ip any any
(where P = pod number)
```

Step 4 Define the outside interface ACL to allow inbound ICMP traffic and routing traffic. Block all other outside-initiated traffic.

```
router(config)# access-list 104 permit eigrp any any
router(config)# access-list 104 deny ip any any
```

Step 5 Exit configuration mode.

```
router(config)# exit
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show access-list** command. The output should be similar to this:

```
router#show ip access-lists
Extended IP access list 103
    10 permit icmp any any
    20 permit tcp 10.0.1.0 0.0.0.255 any eq telnet
    30 permit tcp 10.0.1.0 0.0.0.255 any eq ftp
    40 permit tcp 10.0.1.0 0.0.0.255 any eq www
    50 deny ip any any
Extended IP access list 104
    10 permit eigrp any any
    20 permit icmp any any
    30 deny ip any any
```

Task 3: Apply Inspection Rule and ACL to Interfaces

In this task, you will apply the inspection rule and ACLs to the appropriate interfaces.

Activity Procedure

Complete these steps:

Step 1 Apply the ACL to the inside interface.

```
router(config)# interface fastEthernet 0/0
router(config-if)# ip access-group 103 in
```

Step 2 Apply the inspection rule and ACL to the outside interface.

```
router(config-if)# interface fastEthernet 0/1
router(config-if)# ip inspect FWRULE out
router(config-if)# ip access-group 104 in
```

Step 3 Return to global configuration mode and save your configuration.

```
router(config-if)# end
router# copy run start
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show ip inspect interfaces** command. The output should be similar to this:

```
R1# show ip inspect interfaces
Interface Configuration
 Interface FastEthernet0/1
  Inbound inspection rule is not set
  Outgoing inspection rule is FWRULE
    tcp alert is on audit-trail is off timeout 300
    ftp alert is on audit-trail is off timeout 300
  Inbound access list is 104
  Outgoing access list is not set
```


Task 4: Configure Logging and Audit Trails

In this task, you will configure logging and audit trails.

Activity Procedure

Complete these steps:

Step 1 Log in to your perimeter router and access global configuration mode.

Step 2 Enable logging to the console and the syslog server.

```
router(config)# logging on
router(config)# logging 10.0.P.12
(where P = pod number)
```

Step 3 Enable audit trails.

```
router(config)# ip inspect audit-trail
```

Step 4 Return to global configuration mode.

```
router(config)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue the **show ip inspect config** and **show ip inspect interfaces** commands. The output should be similar to this:

```
R1# show ip inspect config
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 FWRULE
    http alert is on audit-trail is off timeout 300
    ftp alert is on audit-trail is off timeout 300
    icmp alert is on audit-trail is off timeout 300

R1# show ip inspect interfaces
Interface Configuration
 Interface FastEthernet0/1
  Inbound inspection rule is not set
  Outgoing inspection rule is FWRULE
    http alert is on audit-trail is off timeout 300
    ftp alert is on audit-trail is off timeout 300
    icmp alert is on audit-trail is off timeout 300
  Inbound access list is 104
  Outgoing access list is not set
```

Task 5: Test and Verify

In this task, you will test and verify Cisco IOS classic firewall.

Activity Procedure

Complete these steps:

Step 1 Check your ACLs.

```
router# show ip access-lists
```

```
R1# show ip access-lists
Extended IP access list 103
    10 permit icmp any any
    20 permit tcp 10.0.1.0 0.0.0.255 any eq ftp
    30 permit tcp 10.0.1.0 0.0.0.255 any eq www (21 matches)
    40 deny ip any any
Extended IP access list 104
    10 permit eigrp any any (264 matches)
    20 deny ip any any (117 matches)
```

Step 2 Ping the backbone server from the command prompt of your student PC.

```
C:\> ping 172.26.26.50
Pinging 172.26.26.50 with 32 bytes of data:
Reply from 172.26.26.50: bytes=32 time=34ms TTL=125
Reply from 172.26.26.50: bytes=32 time=34ms TTL=125
Reply from 172.26.26.50: bytes=32 time=34ms TTL=125
Reply from 172.26.26.50: bytes=32 time=36ms TTL=125
```

Step 3 Use your web browser to connect to the backbone web server.

```
http://172.26.26.50
```

Step 4 From the command prompt on your student PC, connect to the backbone FTP server using anonymous FTP.

```
C:\> ftp 172.26.26.50
...
User (10.0.P.12:(none)): anonymous
...
Password: user@
```

Step 5 Display a directory listing to verify data channel connectivity.

```
ftp> ls
```

Step 6 Use the following **show** commands to verify the CBAC operation:

```
router# show ip inspect sessions
router# show ip inspect sessions detail
router# show ip inspect name FWRULE
```



```
router# show ip inspect config
router# show ip inspect interfaces
router# show ip inspect statistics
router# show ip inspect all
```

Step 7 Ping the inside server of your peer from your PC command prompt.

```
C:\> ping 10.0.Q.12
Pinging 10.0.Q.12 with 32 bytes of data:
Reply from 10.0.Q.12: bytes=32 time=34ms TTL=125
Reply from 10.0.Q.12: bytes=32 time=34ms TTL=125
Reply from 10.0.Q.12: bytes=32 time=34ms TTL=125
Reply from 10.0.Q.12: bytes=32 time=36ms TTL=125
(where Q = peer pod number)
```

Step 8 Use your web browser to connect to your peer inside server.

```
http://10.0.Q.12
```

Step 9 Connect to the peer FTP server using anonymous FTP.

```
C:\> ftp 10.0.Q.12
...
User (10.0.Q.12:(none)): anonymous
...
Password: user@
(where Q = peer pod number)
```

Activity Verification

You have completed this task when you attain these results:

- Use the following **show** commands to verify the CBAC operation:

```
R1# show ip inspect sessions
Established Sessions
 Session 641721A8 (10.0.1.12:3575)=>(10.0.6.12:80) http SIS_OPEN
 Session 64172460 (10.0.1.12:3573)=>(10.0.6.12:21) ftp SIS_OPEN
 Session 64171C38 (10.0.1.12:3576)=>(10.0.6.12:80) http SIS_OPEN
 Session 64171EF0 (10.0.1.12:8)=>(10.0.6.12:0) icmp SIS_OPEN

R1# show ip inspect sessions detail
Established Sessions
 Session 641721A8 (10.0.1.12:3575)=>(10.0.6.12:80) http SIS_OPEN
  Created 00:00:13, Last heard 00:00:12
  Bytes sent (initiator:responder) [1291:659]
  In  SID 10.0.6.12[80:80]=>10.0.1.12[3575:3575] on ACL 104  (5
matches)
```

```
 Session 64172460 (10.0.1.12:3573)=>(10.0.6.12:21) ftp SIS_OPEN
  Created 00:00:32, Last heard 00:00:18
  Bytes sent (initiator:responder) [28:154]
  In  SID 10.0.6.12[21:21]=>10.0.1.12[3573:3573] on ACL 104  (4
matches)
 Session 64171C38 (10.0.1.12:3576)=>(10.0.6.12:80) http SIS_OPEN
  Created 00:00:13, Last heard 00:00:12
  Bytes sent (initiator:responder) [683:281]
  In  SID 10.0.6.12[80:80]=>10.0.1.12[3576:3576] on ACL 104  (3
matches)
 Session 64171EF0 (10.0.1.12:8)=>(10.0.6.12:0) icmp SIS_OPEN
  Created 00:06:58, Last heard 00:00:00
   ECHO request
  Bytes sent (initiator:responder) [13408:13408]
  In  SID 10.0.6.12[0:0]=>10.0.1.12[0:0] on ACL 104  (369 matches)
  In  SID 0.0.0.0[0:0]=>10.0.1.12[3:3] on ACL 104
  In  SID 0.0.0.0[0:0]=>10.0.1.12[11:11] on ACL 104

R1# show ip inspect name FWRULE
Inspection name FWRULE
    http alert is on audit-trail is on timeout 300
    ftp alert is on audit-trail is on timeout 300
    icmp alert is on audit-trail is off timeout 300

R1# show ip inspect config
Session audit trail is enabled
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 FWRULE
    http alert is on audit-trail is on timeout 300
    ftp alert is on audit-trail is on timeout 300
    icmp alert is on audit-trail is off timeout 300
```



```
R1# show ip inspect interfaces
Interface Configuration
 Interface FastEthernet0/1
  Inbound inspection rule is not set
  Outgoing inspection rule is FWRULE
    http alert is on audit-trail is on timeout 300
    ftp alert is on audit-trail is on timeout 300
    icmp alert is on audit-trail is off timeout 300
  Inbound access list is 104
  Outgoing access list is not set

R1# show ip inspect all
Session audit trail is enabled
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 FWRULE
    http alert is on audit-trail is on timeout 300
    ftp alert is on audit-trail is on timeout 300
    icmp alert is on audit-trail is off timeout 300

Interface Configuration
 Interface FastEthernet0/1
  Inbound inspection rule is not set
  Outgoing inspection rule is FWRULE
    http alert is on audit-trail is on timeout 300
    ftp alert is on audit-trail is on timeout 300
    icmp alert is on audit-trail is off timeout 300
  Inbound access list is 104
  Outgoing access list is not set

Established Sessions
 Session 64171C38 (10.0.1.12:3598)=>(10.0.6.12:21) ftp SIS_OPEN
 Session 641721A8 (10.0.1.12:3597)=>(10.0.6.12:80) http SIS_OPEN
 Session 64172460 (10.0.1.12:3596)=>(10.0.6.12:80) http SIS_OPEN
```

```
 Session 64171EF0 (10.0.1.12:8)=>(10.0.6.12:0) icmp SIS_OPEN

R1# show ip inspect statistics
Packet inspection statistics [process switch:fast switch]
  tcp packets: [3:158]
   packets: [0:1870]
  http packets: [0:78]
  ftp packets: [0:10]
Interfaces configured for inspection 1
Session creations since subsystem startup or last reset 13
Current session counts (estab/half-open/terminating) [2:0:0]
Maxever session counts (estab/half-open/terminating) [4:1:0]
Last session created 00:00:56
Last statistic reset never
Last session creation rate 0
Last half-open session total 0
```

Syslog

Check your syslog server. You should see some traffic from the audit trails

```
11-13-2006    12:44:28    Local7.Info 10.0.1.2    200: *Nov 13
19:46:41.539: %FW-6-SESS_AUDIT_TRAIL: Stop tcp session: initiator
(10.0.1.12:2631) sent 256 bytes -- responder (10.0.6.12:80) sent 4203
bytes

11-13-2006    12:44:28    Local7.Info 10.0.1.2    199: *Nov 13
19:46:41.539: %FW-6-SESS_AUDIT_TRAIL: Stop tcp session: initiator
(10.0.1.12:2630) sent 257 bytes -- responder (10.0.6.12:80) sent 4203
bytes

11-13-2006    12:44:28    Local7.Info 10.0.1.2    198: *Nov 13
19:46:41.539: %FW-6-SESS_AUDIT_TRAIL: Stop tcp session: initiator
(10.0.1.12:2629) sent 559 bytes -- responder (10.0.6.12:80) sent 3967
bytes

11-13-2006    12:44:23    Local7.Info 10.0.1.2    197: *Nov 13
19:46:36.607: %FW-6-SESS_AUDIT_TRAIL_START: Start tcp session:
initiator (10.0.1.12:2631) -- responder (10.0.6.12:80)

11-13-2006    12:44:23    Local7.Info 10.0.1.2    196: *Nov 13
19:46:36.603: %FW-6-SESS_AUDIT_TRAIL_START: Start tcp session:
initiator (10.0.1.12:2630) -- responder (10.0.6.12:80)

11-13-2006    12:44:23    Local7.Info 10.0.1.2    195: *Nov 13
19:46:36.599: %FW-6-SESS_AUDIT_TRAIL_START: Start tcp session:
initiator (10.0.1.12:2629) -- responder (10.0.6.12:80)
```


Lab 5-2: Configure Cisco IOS Application Policy Firewall

Complete this lab activity to practice what you learned in the related module.

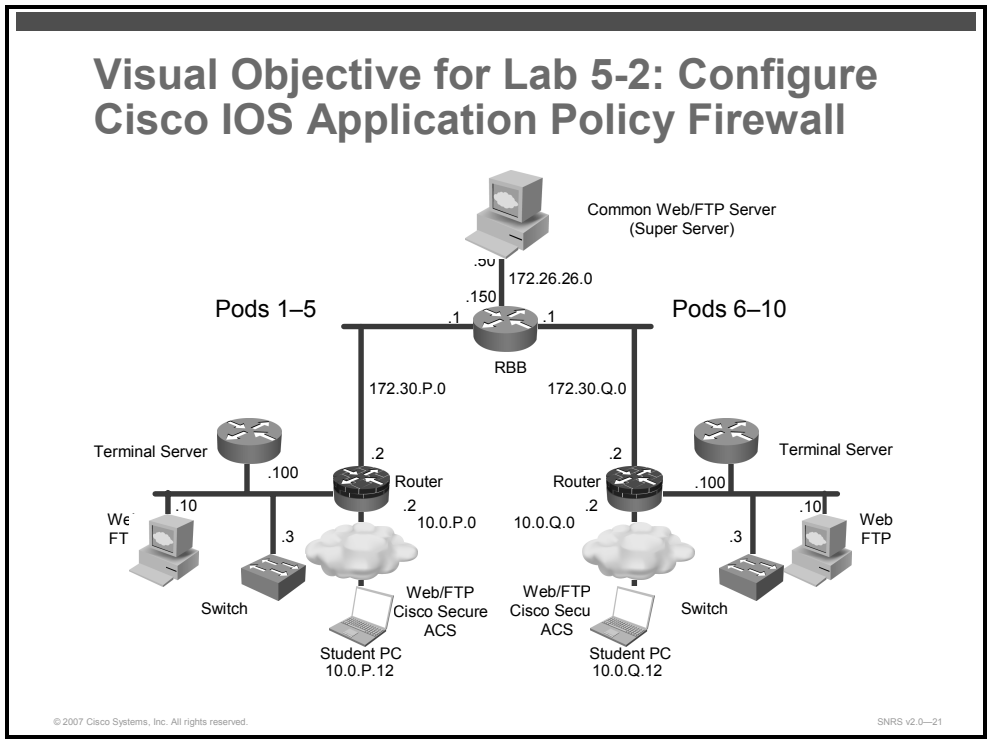
Activity Objective

In this activity, you will configure an application firewall for IM or HTTP. After completing this activity, you will be able to meet these objectives:

- Define an application policy and configure protocol-specific rules
- Apply an application policy to an inspection rule
- Display application firewall policy information

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers

Command List

The table describes the commands that are used in this activity.

Application Firewall Commands

Command	Description
alert {**on** | **off**}	Enables alerts.
appfw policy-name *policy-name*	Defines an application firewall policy.
application *protocol*	Put the router in *appfw-policy*-protocol configuration mode and begin configuring inspection parameters for a given protocol.
audit-trail {**on** | **off**}	Enables logging.
server {**permit** | **deny**} {**name** *string*	Allows or denies access to IM servers.
timeout *seconds*	Specifies the elapsed length of time before an inactive connection is torn down.
service text-chat} **action allow**	Allows the text chat service for IM.
service default action *action*	Specify a default action to take for all services that are not explicitly configured under the application.
strict-http action allow alarm	Enables strict HTTP compliance.
content-length maximum *length* **action allow alarm**	Specifies the range of content length.
content-type-verification match-req-rsp action allow alarm	Enables content-type inspection.
max-header-length request *length* **response 1 action allow alarm**	Specifies the maximum header length.
port-misuse default action allow alarm	Permits or denies HTTP traffic through the firewall on the basis of specified applications in the HTTP message.
request-method rfc default action allow alarm	Specifies that the supported methods of RFC 2616, *Hypertext Transfer Protocol— HTTP/1.1*, are to be used for traffic inspection.
request-method extension default action allow alarm	Specifies that the extension methods are to be used for traffic inspection.
transfer-encoding type default action allow alarm	Permit HTTP traffic according to the specified transfer-encoding of the message.
ip inspect name *inspection-name* **appfw** *policy-name*	Defines a set of inspection rules for the application policy.
ip inspect *inspection-name* **in**	Applies the inspection rules to all traffic entering the specified interface.
show appfw configuration	Displays application firewall configuration.
show appfw name *policy-name*	Displays application firewall configuration of a specific policy.

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will set up the lab devices.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student PC is powered on and that the Microsoft Windows 2000 Server is operational. Your instructor will provide you with the correct username and password to log in to the student PC.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2** (where P = pod number).

Step 3 Make sure that your student PC has an appropriate syslog server application installed (for example, the Kiwi Syslog Daemon).

Step 4 Reload your perimeter router using the default lab configuration.

Step 5 Ensure that you can ping the peer router and network hosts before beginning.

Activity Verification

You have completed this task when you attain these results:

- You can successfully ping your peer pod router.

Task 2: Define an Application Firewall Policy for IM and Configure Protocol-Specific Rules

In this task, you will define an IM application firewall policy and configure specific rules for that protocol.

Activity Procedure

Complete these steps:

Step 1 Define an application firewall policy and enter application firewall policy configuration mode.

```
router(config)# appfw policy-name IM-Policy
```

Step 2 Put the router in "appfw-policy-protocol"configuration mode and begin configuring IM inspection parameters.

```
router(cfg-appfw-policy)# application im aol
```

Step 3 Enable message logging for established or torn-down connections.

```
router(cfg-appfw-policy-aim)# audit-trail on
```

Step 4 Specify the access policy to IM servers.

```
router(cfg-appfw-policy-aim)# server permit name
login.oscar.aol.com
```

Step 5 (Optional) Specify the elapsed length of time before an inactive connection is torn down.

```
router(cfg-appfw-policy-aim)# timeout 30
```

Step 6 Specify an action when a specific service is detected in the IM traffic.

```
router(cfg-appfw-policy-aim)# service text-chat action allow
```

Step 7 Specify a default action to take for all services that are not explicitly configured under the application.

```
router(cfg-appfw-policy-aim)# service default action reset
```

Step 8 (Optional) Enable message logging when events, such as the start of a text chat, begin.

```
router(cfg-appfw-policy-aim)# alert on
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show appfw configuration** command. The output should be similar to this:

```
router# show appfw configuration
Application Firewall Rule configuration
  Application Policy name IM-Policy
    Application: im aol
     service default action: reset
     service text-chat action: allow
     server: permit name login.oscar.aol.com
     timeout: 30     audit-trail: on     alert: on
```


Task 3: Define an Application Firewall Policy for HTTP and Configure Protocol Specific Rules

In this task, you will define a HTTP application policy and configure specific rules for that protocol.

Activity Procedure

Complete these steps:

Step 1 Define an application firewall policy for HTTP and enter application firewall policy configuration mode.

```
router(config)# appfw policy-name HTTP-Policy
```

Step 2 Put the router in "appfw-policy-protocol" configuration mode and begin configuring HTTP inspection parameters.

```
router(cfg-appfw-policy)# application http
```

Step 3 Enable message logging for established or torn-down connections.

```
router(cfg-appfw-policy-http)# audit-trail on
```

Step 4 Enable strict HTTP compliance.

```
router(cfg-appfw-policy-http)# strict-http action allow alarm
```

Step 5 Specify the range of content length.

```
router(cfg-appfw-policy-http)# content-length maximum 1000
action allow alarm
```

Step 6 Enable content-type inspection.

```
router(cfg-appfw-policy-http)# content-type-verification
match-req-rsp action allow alarm
```

Step 7 Specify maximum header length.

```
router(cfg-appfw-policy-http)# max-header-length request 100
response 1 action allow alarm
```

Step 8 Permit or deny HTTP traffic through the firewall on the basis of specified applications in the HTTP message.

```
router(cfg-appfw-policy-http)# port-misuse default action
allow alarm
```

Step 9 Specify that the supported methods of RFC 2616, *Hypertext Transfer Protocol—HTTP/1.1*, are to be used for traffic inspection.

```
router(cfg-appfw-policy-http)# request-method rfc default
action allow alarm
```

Step 10 Specify that the extension methods are to be used for traffic inspection. Default is all types

```
router(cfg-appfw-policy-http)# request-method extension
default action allow alarm
```

Step 11 Permit HTTP traffic according to the specified transfer-encoding of the message. The default is all types.

```
router(cfg-appfw-policy-http)# transfer-encoding type default
action allow alarm
```

Step 12 Exit to global EXEC mode.

```
router(cfg-appfw-policy-http)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show appfw name** *policy-name* command. The output should be similar to this:

```
R1#show appfw name HTTP-Policy
  Application Policy name HTTP-Policy
    Application http
      content-length maximum 1000 action allow alarm
      content-type-verification match-req-rsp action allow alarm
      max-header-length request length 1 response length 1 action
allow
           alarm
      max-uri-length 100 action allow alarm
      port-misuse default action allow alarm
      request-method rfc default action allow alarm
      transfer-encoding default action allow alarm
      audit-trail is enabled
```


Task 4: Apply an Application Policy to a Firewall for Inspection

In this task, you will apply the application policy to a firewall.

Activity Procedure

Complete these steps:

Step 1 Define a set of inspection rules for the application policy.

```
router(config)# ip inspect name FIREWALL appfw IM-Policy
```

OR

```
router(config)# ip inspect name FIREWALL appfw HTTP-Policy
```

Step 2 Enter interface configuration mode.

```
router(config)# interface FastEthernet0/1
```

Step 3 Apply the inspection rules (defined in Step 1) to all traffic entering the specified interface.

```
router#(config-if)# ip inspect FIREWALL out
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show appfw configuration** command. The output should be similar to this:

```
router# show appfw configuration
Application Firewall Rule configuration
  Application Policy name IM-Policy
    Application: im aol
     service default action: reset
     service text-chat action: allow
     server: permit name login.oscar.aol.com
     timeout: 30     audit-trail: on     alert: on
  Application Policy name HTTP-Policy
    Application http
      content-length maximum 1000 action allow alarm
      content-type-verification match-req-rsp action allow alarm
      max-header-length request length 1 response length 1 action
allow
          alarm
      max-uri-length 100 action allow alarm
      port-misuse default action allow alarm
      request-method rfc default action allow alarm
      transfer-encoding default action allow alarm
      audit-trail is enabled
```

Lab 5-3: Configure a Cisco IOS Zone-Based Policy Firewall

Complete this lab activity to practice what you learned in the related module.

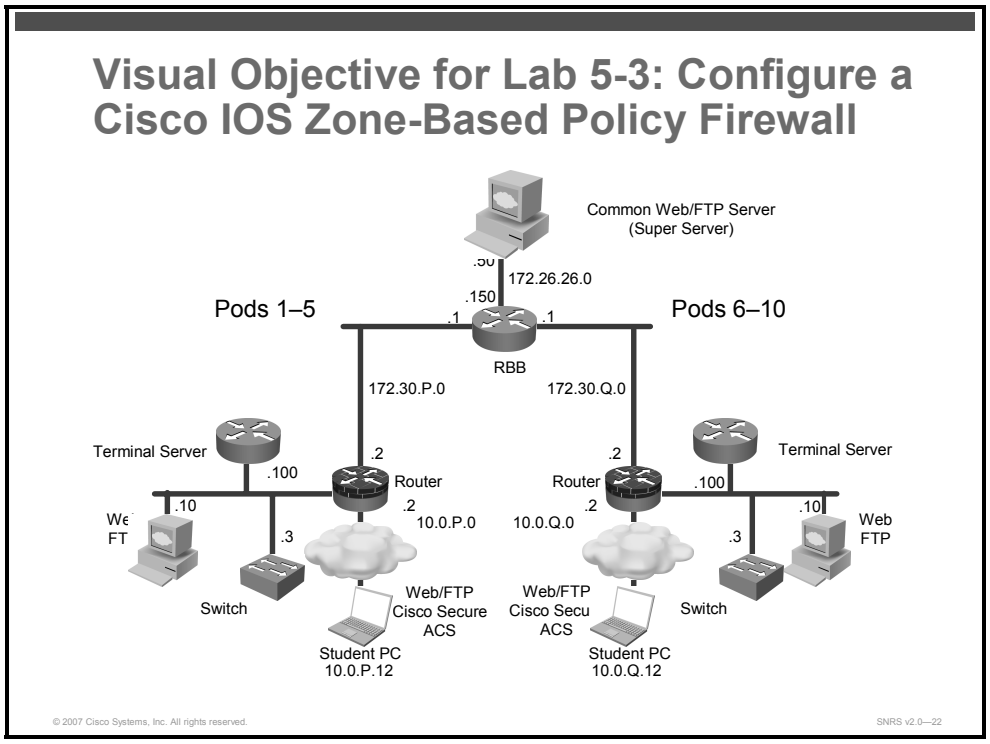
Activity Objective

In this activity, you will configure a Cisco IOS zone-based policy firewall on a perimeter router. After completing this activity, you will be able to meet these objectives:

- Create a class map and a policy map
- Configure a security zone
- Create a zone pair
- Assign interfaces to a zone pair
- Attach a policy map to a zone pair
- Configure the basic inspection of traffic

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student laptops
- Pod routers

Command List

The table describes the commands that are used in this activity.

Cisco IOS Zone-Based Policy Firewall Commands

Command	Description
class-map type inspect match-all *map-name*	Creates a Layer 3 or Layer 4 inspect type class map and enter class map configuration mode .
match access-group *acl-number*	Specify ACL to match.
match protocol *protocol*	Specify protocol to inspect.
policy-map type inspect *policy-name*	Creates an inspection policy map.
class type inspect *class-name*	Creates an inspection class map.
inspect	Enables inspection with the inspection policy map.
zone security *zone-name*	Creates a security zone.
zone-member security *zone-name*	Specifies an interface as a zone member.
zone-pair security *zone-pair-name* **source** *zone-name* **destination** *zone-name*	Creates a zone-pair.
show class-map type inspect	Displays inspection class map information.
show policy-map type inspect	Displays inspection policy map information.
show zone security	Displays information about configured security zones.
show zone-pair security	Displays information about configured security zone-pairs.

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will set up the lab devices.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student PC is powered on and that the Windows 2000 Server is operational. Your instructor will provide you with the correct username and password to log in to the student PC.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2** (where P = pod number).

Step 3 Make sure that your student PC has an appropriate syslog server application installed (for example, the Kiwi Syslog Daemon).

Step 4 Reload your perimeter router using the default lab configuration.

Step 5 Ensure that you can ping the peer router and network hosts before beginning.

Step 6 Make sure that your router is running the correct date and time.

Step 7 Make sure that your student PC is running the correct date and time.

Activity Verification

You have completed this task when you attain these results:

- You can successfully ping your pod router.

Task 2: Configure a Policy

In this task, you will create a class map and policy map for Layer 3 and Layer 4.

Activity Procedure

Complete these steps:

Create a Class Map

Step 1 Create an ACL to match in the class map.

```
router(config)# access-list 110 permit ip 10.0.P.0 0.0.0.255
10.0.Q.0 0.0.0.255
router(config)# access-list 110 permit ip 10.0.P.0 0.0.0.255
172.26.26.0 0.0.0.255
```

Step 2 Create a Layer 3 or Layer 4 inspect type class map and enter class map configuration mode.

```
router(config)# class-map type inspect match-all HTTP-Class
```

Step 3 Configure the match criteria for a class map based on an ACL name or number.

```
router(config-cmap)# match access-group 110
```

Step 4 Configure the match criteria for a class map on the basis of a specified protocol. In this case, HTTP.

```
router(config-cmap)# match protocol http
```

Step 5 Return to global configuration mode.

```
router(config-cmap)# exit
```

Create a Policy Map

Step 6 Create a Layer 3 and Layer 4 inspect type policy map and enter policy map configuration mode.

```
router(config)# policy-map type inspect HTTP-Policy
```

Step 7 Specify the traffic (class) on which an action is to be performed.

```
router(config-pmap)# class type inspect HTTP-Class
```

Step 8 Enable Cisco IOS stateful packet inspection.

```
router(config-pmap-c)# inspect
```

Step 9 Return to global configuration mode.

```
router(config-pmap-c)# exit
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show class-map type inspect** and **show policy-map type inspect** command.

```
R1# show class-map type inspect
 Class Map type inspect match-all HTTP-Class (id 1)
   Match access-group  110
   Match protocol http

R1# show policy-map type inspect
  Policy Map type inspect HTTP-Policy
    Class HTTP-Class
      Inspect ERROR ←(This is a bug in the IOS. The "error" after Inspect)
```


Task 3: Create a Security Zone and Assign Interfaces to a Security Zone

In this task, you will configure two security zones and assign interfaces to the zones.

Activity Procedure

Complete these steps:

Step 1 Create a security zone for the inside interface.

```
router(config)# zone security Inside
```

Step 2 Describe the zone.

```
router(config-sec-zone)# description Inside Security Zone
```

Step 3 Create a security zone for the outside interface.

```
router(config)# zone security Outside
```

Step 4 Describe the zone.

```
router(config-sec-zone)# description Outside Security Zone
```

Step 5 Return to global configuration mode.

```
router(config-sec-zone)# exit
```

Step 6 Specify the outside interface for configuration and enter interface configuration mode.

```
router(config)# interface fa0/1
```

Step 7 Assign the interface to a specified security zone.

```
router(config-if)# zone-member security Outside
```

Step 8 Return to global configuration mode.

```
router(config-sec-zone)# exit
```

Step 9 Specify the outside interface for configuration and enter interface configuration mode.

```
router(config)# interface fa0/0
```

Step 10 Assign the interface to a specified security zone.

```
router(config-if)# zone-member security Inside
```

Step 11 Return to privileged exec mode.

```
router(config-sec-zone)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show zone security** command. The output should look similar to this:

```
R1# show zone security
zone self
  Description: System defined zone

zone Inside
  Description: Inside Security
  Member Interfaces:
    FastEthernet0/0

zone Outside
  Description: Outside Security
  Member Interfaces:
    FastEthernet0/1
```


Task 4: Configure a Zone Pair

In this task, you will configure a zone pair.

Activity Procedure

Complete these steps:

Step 1 Create a zone pair.

```
router(config)# zone-pair security SNRS-PAIR source Inside
destination Outside
```

Step 2 Describe the zone pair.

```
router(config-sec-zone)# description SNRS Zone-pair
```

Step 3 Return to global configuration mode.

```
router(config-sec-zone)# exit
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show zone-pair security** command. The output should be similar to this:

```
R1# show zone-pair security
Zone-pair name SNRS-PAIR
Description: SNRS Zone-pair
    Source-Zone Inside  Destination-Zone Outside
    service-policy not configured
```

Task 5: Attach a Policy Map to the Zone Pair

In this task, you will attach a policy map to the zone pair that you created.

Activity Procedure

Complete these steps:

Step 1 Enter zone pair configuration mode.

```
router(config)# zone-pair security SNRS-PAIR
```

Step 2 Attach a firewall policy map to the zone pair.

```
router(config-sec-zone-pair)# service-policy type inspect
HTTP-Policy
```

Step 3 Return to global privileged EXEC mode.

```
router(config-sec-zone-pair)# end
```

Activity Verification

You have completed this task when you attain these results:

- Use the following **show** commands to verify Cisco IOS zone-based policy firewall configuration:

```
R1# show zone-pair security
Zone-pair name SNRS-PAIR
Description: SNRS Zone-pair
    Source-Zone Inside  Destination-Zone Outside
    service-policy HTTP-Policy


R1# show policy-map type inspect zone-pair SNRS-PAIR
 Zone-pair: SNRS-PAIR
  Service-policy inspect : HTTP-Policy
    Class-map: HTTP-Class (match-all)
      Match: access-group 110
      Match: protocol http
      Inspect
        Session creations since subsystem startup or last reset 0
        Current session counts (estab/half-open/terminating) [0:0:0]
        Maxever session counts (estab/half-open/terminating) [0:0:0]
        Last session created never
        Last session created never
        Last statistic reset never
        Last session creation rate 0
        Last half-open session total 0
    Class-map: class-default (match-any)
```



```
Match: any
Drop (default action)
  0 packets, 0 bytes
```

Lab 5-4: Configure Cisco IOS Authentication Proxy on a Cisco Router

Complete this lab activity to practice what you learned in the related module.

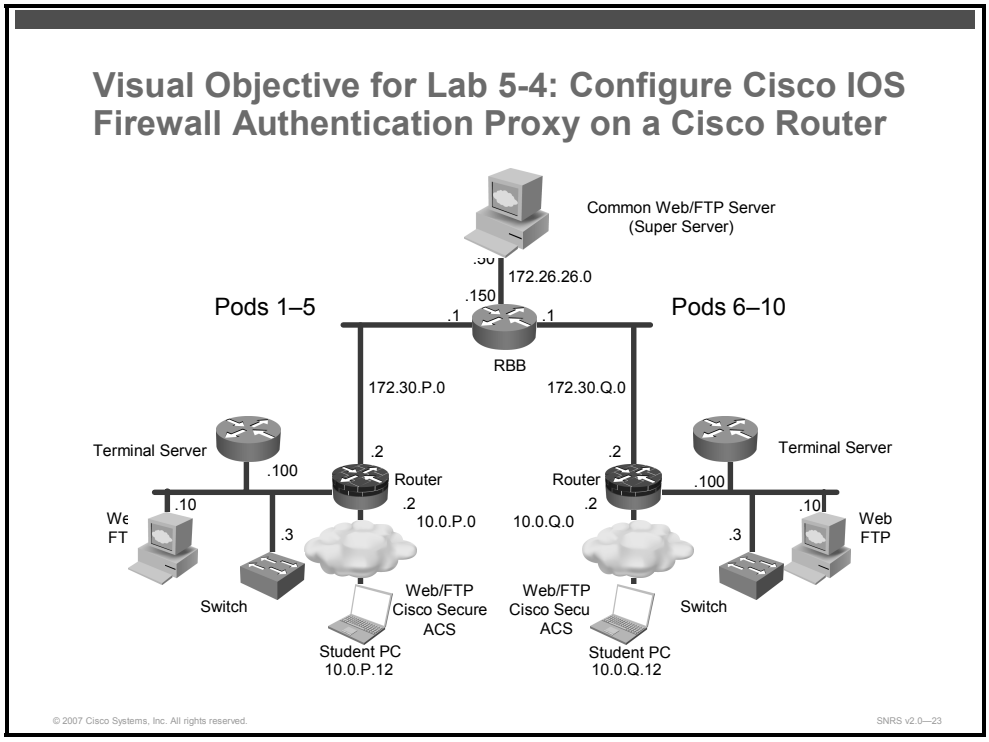
Activity Objective

In this activity, you will configure Cisco IOS Firewall authentication proxy on a Cisco router. After completing this activity, you will be able to meet these objectives:

- Set up lab devices
- Configure Cisco Secure ACS to support Cisco IOS Firewall authentication proxy
- Configure AAA
- Configure a Cisco IOS Firewall authentication proxy
- Test and verify auth-proxy configuration

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student computers
- Pod routers

Command List

The table describes the commands that are used in this activity.

Cisco IOS Authentication Proxy Commands

Command	Description
aaa authentication enable default *method1* [***method2...***]	To enable AAA authentication to determine whether a user can access the privileged command level, use the **aaa authentication enable default** command in global configuration mode. To disable this authorization method, use the **no** form of this command.
aaa authentication login {**default** | *list-name*} *method1* [*method2...*]	To set AAA authentication at login, use the **aaa authentication login** command in global configuration mode. To disable AAA authentication, use the **no** form of this command.
aaa authorization {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*]	To set parameters that restrict user access to a network, use the **aaa authorization** command in global configuration mode. To disable authorization for a function, use the **no** form of this command.
aaa new-model	To enable the AAA access control model, issue the **aaa new-model** command in global configuration mode. To disable the AAA access control model, use the **no** form of this command.
access-list *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]	To define an extended IP ACL, use the extended version of the **access-list** command in global configuration mode. To remove the ACLs, use the **no** form of this command.
ip access-group {*access-list-number* | *access-list-name*}{**in** | **out**}	To control access to an interface, use the **ip access-group** command in interface configuration mode. To remove the specified access group, use the **no** form of this command.
ip auth-proxy {**inactivity-timer** *min* | **absolute-timer** *min*}	To set the Cisco IOS authentication proxy idle timeout value (the length of time that an authentication cache entry, along with its associated dynamic user ACL, is managed after a period of inactivity), use the **ip auth-proxy** command in global configuration mode. To set the default value, use the **no** form of this command.
ip auth-proxy *auth-proxy-name*	To apply a Cisco IOS authentication proxy rule at a firewall interface, use the **ip auth-proxy** command in interface configuration mode. To remove the Cisco IOS authentication proxy rules, use the **no** form of this command.
ip http authentication {**aaa** | **enable** | **local** | **tacacs**}	To specify a particular authentication method for HTTP server users, use the **ip http authentication** command in global configuration mode. To disable a configured authentication method, use the **no** form of this command.
ip http server	To enable the HTTP server on your system, including the Cisco web browser user interface, use the **ip http server** command in global configuration mode. To disable the HTTP server, use the **no** form of this command.

ping [*protocol*] [**tag**] {*host-name* | *system-address*}	To diagnose basic network connectivity on AppleTalk, ATM, CLNS, DECnet, IP, Novell IPX, or SRB networks, use the **ping** command in EXEC mode.
show access-lists [***access-list-number*** | *access-list-name*]	To display the contents of current ACLs, use the **show access-lists** command in privileged EXEC mode.
show ip auth-proxy {**cache** | **configuration**}	To display the Cisco IOS authentication proxy entries or the running Cisco IOS authentication proxy configuration, use the **show ip auth-proxy** command in privileged EXEC mode.
tacacs-server host *host-name* [**port** *integer*] [**timeout** *integer*] [**key** *string*] [**single-connection**] [**nat**]	To specify a TACACS+ host, use the **tacacs-server host** command in global configuration mode. To delete the specified name or address, use the **no** form of this command.
tacacs-server key *key*	To set the authentication encryption key used for all TACACS+ communications between the access server and the TACACS+ daemon, use the **tacacs-server key** command in global configuration mode. To disable the key, use the **no** form of this command.
username *name* {**nopassword** | **password** *password* | **password** *encryption-type encrypted-password*}	To establish a username-based authentication system, use the **username** command in global configuration mode.

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will set up the lab devices.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student PC is powered on and that the Microsoft Windows 2000 Server is operational. Your instructor will provide you with the correct username and password to log in to the student PC.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2** (where P = pod number).

Step 3 Reload your perimeter router using the default lab configuration.

Step 4 Ensure that you can ping the other routers and network hosts before beginning.

Activity Verification

You have completed this task when you attain these results:

- You can successfully ping the other hosts.

Task 2: Configure Cisco Secure ACS to Support Cisco IOS Authentication Proxy

In this task, you will configure the Cisco Secure ACS to work with Cisco IOS authentication proxy.

Activity Procedure

Complete these steps:

Step 1 On your student PC, open Cisco Secure ACS from the desktop.

Add the Cisco IOS NAD as a AAA Client

Complete these substeps.

Step 2 Click the **Network Configuration** button in the navigation bar.

Step 3 In the AAA Clients box, click **Add Entry**. The Add AAA Client window opens.

Step 4 Enter the hostname of your router as **RP** (where P = your pod number) in the AAA Client Hostname field.

Step 5 Enter an IP address of **10.0.P.2** (where P = your pod number) in the AAA Client IP Address field. This is the IP address of the switch (NAD) interface that will forward TACACS+ packets to the Cisco Secure ACS.

Step 6 Enter a shared TACACS key of **ciscosecure** in the Key field.

Step 7 Select **TACACS+ (Cisco IOS)** from the Authenticate Using list.

Step 8 Click **Submit + Apply**.

Step 9 Click **Interface Configuration** on the left column of Cisco Secure ACS. The Interface Configuration window opens.

Step 10 Click **TACACS+ (Cisco IOS)** to configure this option.

Step 11 Scroll down to locate the New Services area.

Step 12 Choose the first field under New Services and enter **auth-proxy** in the Service field.

Step 13 Check the **Service** field group check box. Make sure that you check the check box directly to the left of the Service field.

Step 14 Scroll to the Advanced Configuration Options area and verify that the **Advanced TACACS+ features** option is selected.

Step 15 Click the **Submit** button to submit your changes.

Step 16 Click the **Group Setup** button. The Group Setup window opens.

Step 17 Choose **Group 2** from the Group drop-down menu.

Step 18 Click **Edit Settings** to view the Group Settings for this group.

Step 19 Scroll down to the TACACS+ Settings area and locate the Auth-Proxy and Custom Attributes check boxes. Check both the **Auth-Proxy** check box and the **Custom Attributes** check box.

Step 20 Enter the following in the Custom Attributes box (note that long lines of text, such as the proxyacl#1 line shown here, can wrap within the Custom Attributes box and may look like two lines):

```
proxyacl#1=permit tcp any host 172.26.26.50 eq www
proxyacl#2=permit icmp any any
priv-lvl=15
```

Step 21 Click **Submit + Restart**.

Step 22 Return to the User Setup and add a new username of **aaauser** with a password of **cisco123** to Group 2.

Step 23 Click the **Submit + Restart** button to submit your changes and restart the Cisco Secure ACS. Wait for the interface to return to the Group Setup main window.

Activity Verification

You have completed this task when you attain these results:

- Review the settings that you just configured in Cisco Secure ACS.

Task 3: Configure AAA

In this task, you will configure AAA on the router.

Activity Procedure

Complete these steps:

Step 1 Create a user account in the local database.

```
router(config)# username cisco password cisco
```

Step 2 Enable AAA.

```
router(config)# aaa new-model
```

Step 3 Define the TACACS+ server and its key.

```
router(config)# tacacs-server host 10.0.P.12
router(config)# tacacs-server key ciscosecure
(where P = pod number)
```

Step 4 Specify the authentication protocol for logins.

```
router(config)# aaa authentication login default group tacacs+
local
```

Step 5 Specify the authorization protocol for Cisco IOS authentication proxy.

```
router(config)# aaa authorization auth-proxy default group
tacacs+ local
```


Step 6 Define a new ACL to allow TACACS+ traffic to the inside interface from your AAA server. Also allow outbound ICMP traffic and CBAC traffic (FTP and World Wide Web). Block all other inside-initiated traffic.

```
router(config)# access-list 101 permit tcp host 10.0.P.12 eq
tacacs host 10.0.P.2
router(config)# access-list 101 permit icmp any any
router(config)# access-list 101 deny ip any any
(where P = pod number)
```

Step 7 Apply the new ACL to the Fa0/0 interface of your perimeter router.

```
router(config)# interface Fa0/0
router(config-if)# ip access-group 101 in
router(config-if)# exit
```

Step 8 Enable the router HTTP server for AAA

```
router(config)# ip http server
router(config)# ip http secure-server
router(config)# ip http authentication aaa
router(config)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show access-lists** command and a **show ip http server status** command. The output should be similar to this:

```
router#show ip access-list
Extended IP access list 101
    10 permit tcp host 10.0.1.12 eq tacacs host 10.0.1.2
    20 permit icmp any any
    30 deny ip any any


R1# show ip http server status
HTTP server status: Enabled
HTTP server port: 80
HTTP server authentication method: aaa
HTTP server access class: 0
HTTP server base path:
HTTP server help root:
Maximum number of concurrent server connections allowed: 5
Server idle time-out: 5 seconds
Server life time-out: 86400 seconds
Maximum number of requests allowed on a connection: 10000
HTTP server active session modules: ALL
```

```
HTTP secure server capability: Present
HTTP secure server status: Enabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-
md5 rc4-128-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:
HTTP secure server active session modules: ALL
```


Task 4: Configure Cisco IOS Authentication Proxy

In this task, you will configure Cisco IOS authentication proxy on the router.

Activity Procedure

Complete these steps:

Step 1 Define a Cisco IOS authentication proxy rule.

```
router(config)# ip auth-proxy name APRULE http inactivity-time
5
```

Step 2 Apply the Cisco IOS authentication proxy rule to the inside interface.

```
router(config)# interface fast 0/0
router(config-if)# ip auth-proxy APRULE
router(config-if)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show ip auth-proxy configuration** command. The output should be similar to this:

```
R1# show ip auth-proxy configuration
Authentication Proxy Banner not configured
Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication global init state time is 2 minutes
Authentication Proxy Session ratelimit is 100
Authentication Proxy Watch-list is disabled
Authentication Proxy Auditing is disabled
Max Login attempts per user is 5

Authentication Proxy Rule Configuration
 Auth-proxy name APRULE
    http list not specified inactivity-timer 5 minutes
```

Task 5: Verify and Test the Configuration

In this task, you will test and verify Cisco IOS authentication proxy.

Activity Procedure

Complete these steps:

Step 1 Use your web browser to connect to the backbone web server. In the URL field, enter the following:

```
http://172.26.26.50
```

Step 2 Enter the following when the web browser prompts you for a username and password:

```
Username: aaauser
Password: cisco123
```

Step 3 From your workstation command prompt, ping the backbone server.

```
C:\> ping 172.26.26.50
Pinging 172.26.26.50 with 32 bytes of data:
Reply from 172.26.26.50: bytes=32 time=34ms TTL=125
Reply from 172.26.26.50: bytes=32 time=34ms TTL=125
Reply from 172.26.26.50: bytes=32 time=34ms TTL=125
Reply from 172.26.26.50: bytes=32 time=36ms TTL=125
```

Step 4 Use the **show ip access-list** command to check your ACLs.

```
router# show ip access-list
```

Step 5 Use the **show ip auth-proxy cache** command to verify the Cisco IOS authentication proxy configuration.

```
router# show ip auth-proxy cache
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show ip access-list** and a **show ip auth-proxy cache** command. The output should be similar to this:

```
R1# show ip access-lists
Extended IP access list 101
     permit ip host 10.0.1.12 any (31 matches)
    10 permit tcp host 10.0.1.12 eq tacacs host 10.0.1.2
    20 permit icmp any any
    30 deny ip any any (143 matches)


R1# show ip auth-proxy cache
Authentication Proxy Cache
Client Name cisco, Client IP 10.0.1.12, Port 2141, timeout 5, Time
Remaining 3, state ESTAB
```


Lab 5-5: Configure a Cisco Router with Cisco IOS IPS

Complete this lab activity to practice what you learned in the related module.

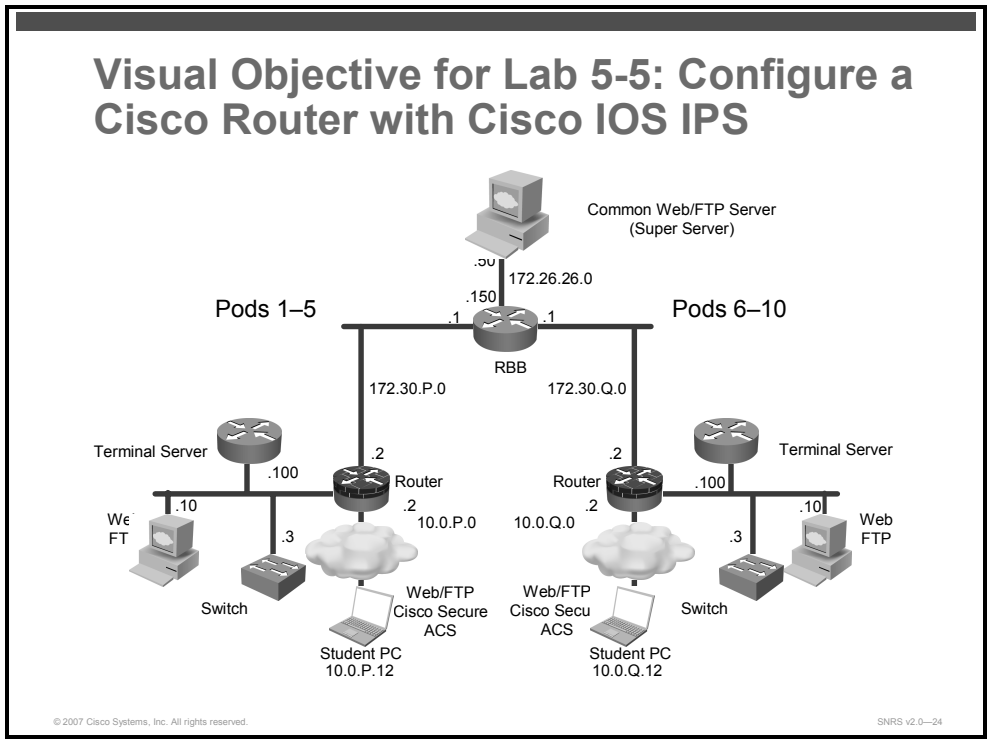
Activity Objective

In this activity, you will configure a Cisco router with Cisco IOS Firewall IPS. After completing this activity, you will be able to meet these objectives:

- Set up lab devices
- Initialize IPS
- Load signatures
- Merge the 128MB.sdf file with the default, built-in signatures
- Verify the configuration
- Generate a test message

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- Student computers
- Pod routers
- Cisco Secure ACS

Command List

The table describes the commands that are used in this activity.

IPS Commands

Command	Description
ip ips *ips-name* {**in** | **out**} [**list** *acl*]	Applies an IPS rule to an interface.
ip ips fail closed	Instructs the router to drop all packets until the signature engine is built and ready to scan traffic.
ip ips name *ips-name*	Specifies an IPS rule.
ip ips sdf location *url*	Specifies the location in which the router will load the SDF.
ip virtual-reassembly	Enables virtual reassembly of IP packets.
copy flash:*name***.sdf ips-sdf**	Merges SDF in flash with built-in signatures.
copy ips-sdf flash:*name***.sdf**	Saves signatures in a new file.
show ip ips {[**all**] [**configuration**] [**interfaces**] [**name** *name*] [**statistics** [**reset**]] [**sessions** [**details**]] [**signatures** [**details**]]}	Displays IPS information, such as configured sessions and signatures.

Job Aids

There are no job aids for this activity.

Task 1: Set Up Lab Devices

In this task, you will set up the lab devices.

Activity Procedure

Complete these steps:

Step 1 Ensure that your student PC is powered on and that the Microsoft Windows 2000 Server is operational.

Step 2 Configure your student PC for IP address **10.0.P.12** with a default gateway of **10.0.P.2** (where P = pod number).

Step 3 Make sure that your student PC has an appropriate syslog server application installed (for example, Kiwi Syslog Daemon).

Step 4 Reload your perimeter router using the default lab configuration.

Step 5 Ensure that you can ping the other routers and network hosts before beginning.

Activity Verification

You have completed this task when you attain these results:

- You can successfully ping the other hosts.

Task 2: Initialize IPS

In this task, you will initialize IPS on the router. This task allows you to load the default, built-in signatures. If you want to merge the two signature files, you must load the default, built-in signatures as described in this task. Then, you can merge the default signatures with the attack-drop.sdf file.

Activity Procedure

Complete these steps:

Step 1 Create an IPS rule.

```
router(config)# ip ips name SECURIPS
```

Step 2 Enter interface configuration mode on the outside interface of your router.

```
router(config)# interface Fa0/1
```

Step 3 Apply an IPS rule at an interface. This command automatically loads the signatures and builds the signature engines.

```
router(config-if)# ip ips SECURIPS in
```

Step 4 Enable virtual reassembly.

```
router(config-if)# ip virtual-reassembly
```

Step 5 Exit to global configuration mode.

```
router(config-if)# exit
```

Step 6 Turn on logging.

```
router(config)#logging on
```

Step 7 Configure the logging host.

```
router(config)#logging 10.0.P.12
(Where P = pod number)
```

Step 8 Configure the trap level.

```
router(config)#logging trap
```

Step 9 Turn on logging.

```
router(config)#logging on
```

Step 10 Exit to privileged mode.

```
router(config)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show ips configuration** command. The output should be similar to this:

```
R1# show ip ips configuration
Configured SDF Locations: none
Built-in signatures are enabled and loaded
Last successful SDF load time: 13:32:37 CST Oct 16 2006
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: 135
Total Inactive Signatures: 0
Signature 50000:0 disable
Signature 50000:1 disable
Signature 50000:2 disable
Signature 1107:0 disable
IPS Rule Configuration
 IPS name SECURIPS
Interface Configuration
 Interface FastEthernet0/1
  Inbound IPS rule is SECURIPS
  Outgoing IPS rule is not set
```


Task 3: Load Signatures

In this task, you will replace the existing signatures in your router with the latest IPS signature file, 128MB.sdf.

Activity Procedure

Complete these steps:

Step 1 Specify the location where the router will load the SDF. If this command is not issued, the router will load the default SDF.

```
router(config)# ip ips sdf location flash:128MB.sdf
```

Step 2 (Optional) Instruct the router to drop all packets until the signature engine is built and ready to scan traffic. If this command is issued, one of the following scenarios will occur:

- If IPS fails to load the SDF, all packets will be dropped—unless the user specifies an ACL for packets to send to IPS.
- If IPS successfully loads the SDF but fails to build a signature engine, all packets that are destined for that engine will be dropped.

```
router(config)# ip ips fail closed
```

Note If this command is not issued, all packets will be passed without scanning if the signature engine fails to build.

Step 3 Enter interface configuration mode for the outside interface.

```
router(config)# interface Fa0/1
```

Step 4 Remove the IPS rule at the interface.

```
router(config-if)# no ip ips SECURIPS in
```

Step 5 Apply the IPS rule at the interface. This command automatically loads the new signatures and builds the signature engines.

```
router(config-if)# ip ips SECURIPS in
```

Note Whenever signatures are replaced or merged, the router prompt is suspended while the signature engines for the newly added or merged signatures are being built. The router prompt will be available again after the engines are built.

Step 6 Exit back to privileged EXEC mode.

```
router(config-if)# end
```

Activity Verification

You have completed this task when you attain these results:

- Issue another **show ip ips configuration** command. The output should be similar to this:

```
R1# show ip ips configuration
Configured SDF Locations:
```

```
 flash:128MB.sdf
Builtin signatures are enabled but not loaded
Last successful SDF load time: 13:39:29 CST Oct 16 2006
IPS fail closed is enabled
Fastpath ips is enabled
Quick run mode is enabled
Event notification through syslog is enabled
Event notification through SDEE is disabled
Total Active Signatures: 303
Total Inactive Signatures: 0
Signature 50000:0 disable
Signature 50000:1 disable
Signature 50000:2 disable
IPS Rule Configuration
 IPS name SECURIPS
Interface Configuration
 Interface FastEthernet0/1
  Inbound IPS rule is SECURIPS
  Outgoing IPS rule is not set
```

- Issue a **show ip ips signatures** command. The output should be similar to this:

```
router# show ip ips signatures
Builtin signatures are configured
Signatures were last loaded from flash:128MB.sdf
Cisco SDF release version 128MB.sdf v2
Trend SDF release version V0.0
*=Marked for Deletion  Action=(A)larm,(D)rop,(R)eset   Trait=AlarmTraits
MH=MinHits             AI=AlarmInterval               CT=ChokeThreshold
TI=ThrottleInterval    AT=AlarmThrottle               FA=FlipAddr
WF=WantFrag

Signature Micro-Engine: OTHER (4 sigs)
 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF
Version
 ----------- -- ------ ---- -----  ----- ----- ----- ----- -- -- -- -----
--
 1203:0      Y   A     HIGH     0      0     0    30    15 FA  N  N
2.2.1.5
 1202:0      Y   A     HIGH     0      0     0   100    15 FA  N  N
2.2.1.5
 3050:0      Y   A     HIGH     0      0     0   100    15 FA  N    1.0
```

```
 1201:0      Y   A     HIGH     0      0     0    30    15 FA  N  N
2.2.1.5


Signature Micro-Engine: STRING.ICMP (1 sigs)

 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF
Version

 ----------- -- ------ ---- -----  ----- ----- ----- ----- -- -- -- -----
--

 2156:0      Y   A      MED     0      0     0   100    15 FA  N    S54


Signature Micro-Engine: STRING.UDP (16 sigs)

 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF
Version

 ----------- -- ------ ---- -----  ----- ----- ----- ----- -- -- -- -----
--

 11209:0     Y   A     INFO     0      0     0   100    15 FA  N    S139

 11208:0     Y   A     INFO     0      0     0   100    15 FA  N    S139

 4608:2      Y   A     HIGH     0      1     0   100    15 FA  N    S30

 4608:1      Y   A     HIGH     0      1     0   100    15 FA  N    S30

 4608:0      Y   A     HIGH     0      1     0   100    15 FA  N    S30

 11000:2     Y   A      LOW     0      0     0   100    15 FA  N    S136

 11000:1     Y   A      LOW     0      0     0   100    15 FA  N    S37

 11000:0     Y   A      LOW     0      0     0   100    15 FA  N    S37

 11207:0     Y   A     INFO     0      0     0   100    15 FA  N    S139

 4607:4      Y   A     HIGH     0      0     0   100    15 FA  N    S30
```

Task 4: Merge the 128MB.sdf File with the Default, Built-in Signatures

You may want to merge the built-in signatures with the attack-drop.sdf file if you find that the built-in signatures are not providing your network with adequate protection from security threats. Use this task to add the SDF and to change default parameters for a specific signature within the SDF or signature engine.

Activity Procedure

Complete these steps:

Step 1 Reload built-in signatures.

```
router(config)# no ip ips sdf location flash:128MB.sdf
router(config)# int Fa0/1
router(config-if)# no ip ips SECURIPS in
router(config-if)# ip ips SECURIPS in
router(config-if)# end
```

Step 2 Merge the flash memory-based SDF (128MB.sdf) with the built-in signatures.

```
router# copy flash:128MB.sdf ips-sdf
```

Note	This command loads the SDF in the router. The SDF will merge with the signatures that are already loaded in the router, unless the **/erase** keyword is issued.

Step 3 Save the newly merged signatures in a new file.

```
router# copy ips-sdf flash:snrs-signatures.sdf
```

Step 4 Configure the router to use the new SDF

```
router(config)# ip ips sdf location flash:snrs-signatures.sdf
```

Step 5 Reinitialize the IPS by removing the IPS rule set and reapplying the rule set.

```
router(config-if)# interface fa 0/1
router(config-if)# no ip ips SECURIPS in
```

Step 6 Reapply the rule set to the interface.

```
router(config-if)# ip ips SECURIPS in
```

Step 7 Exit back to privileged EXEC mode.

```
router(config-if)# end
```


Activity Verification

You have completed this task when you attain these results:

- Issue a **show ip ips configuration** command. The output should be similar to this:

```
R1# show ip ips configuration
Configured SDF Locations:
 flash:snrs-signatures.sdf
Builtin signatures are enabled but not loaded
Last successful SDF load time: 13:51:07 CST Oct 16 2006
IPS fail closed is enabled
Fastpath ips is enabled
Quick run mode is enabled
Event notification through syslog is enabled
Event notification through SDEE is disabled
Total Active Signatures: 370
Total Inactive Signatures: 0
Signature 50000:0 disable
Signature 50000:1 disable
Signature 50000:2 disable
Signature 1107:0 disable
IPS Rule Configuration
 IPS name SECURIPS
Interface Configuration
 Interface FastEthernet0/1
  Inbound IPS rule is SECURIPS
  Outgoing IPS rule is not set
```

Task 5: Verify the Configuration

In this task, you will verify the IPS router configuration.

Activity Procedure

Complete these steps:

Step 1 Display your IPS interface configuration. The parameters that you just configured along with several default settings are displayed.

```
router# show ip ips interfaces
```

Activity Verification

You have completed this task when you attain these results:

- Issue a **show ip ips interfaces** command. The output should be similar to the following:

```
R1#show ip ips interfaces
Interface Configuration
 Interface FastEthernet0/1
  Inbound IPS rule is SECURIPS
  Outgoing IPS rule is not set
```


Task 6: Generate a Test Message

In this task, you will generate a test message to test IPS.

Activity Procedure

Complete these steps:

Step 1 Start the syslog server on your Microsoft Windows 2000 Server.

Step 2 Send multiple fragmented packets to the perimeter router of another pod using the following special technique:

```
router# ping
Protocol [IP] <Enter>
Target IP address: 172.30.Q.2<Enter>
Repeat count [5]: 20
Datagram size [100]: 2000
Timeout in seconds [2]: <Enter>
Extended commands [n]: <Enter>
Sweep range of sizes [n]: <Enter>
```

Step 3 Analyze the syslog messages on the syslog server.

Activity Verification

You have completed this task when you attain these results:

- Check the syslog server log file. The output should resemble the following:

```
10-16-2006      14:04:48    Local7.Warning          10.0.1.2    253:
*Oct 16 20:06:35.962: %IPS-4-SIGNATURE: Sig:2000 Subsig:0 Sev:2 ICMP
Echo Rply [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    252:
*Oct 16 20:06:35.962: %IPS-4-SIGNATURE: Sig:2150 Subsig:0 Sev:2
Fragmented ICMP [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    251:
*Oct 16 20:06:35.962: %IPS-4-SIGNATURE: Sig:2151 Subsig:0 Sev:2 Large
ICMP [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    250:
*Oct 16 20:06:35.942: %IPS-4-SIGNATURE: Sig:2000 Subsig:0 Sev:2 ICMP
Echo Rply [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    249:
*Oct 16 20:06:35.942: %IPS-4-SIGNATURE: Sig:2150 Subsig:0 Sev:2
Fragmented ICMP [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    248:
*Oct 16 20:06:35.942: %IPS-4-SIGNATURE: Sig:2151 Subsig:0 Sev:2 Large
ICMP [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    247:
*Oct 16 20:06:35.938: %IPS-4-SIGNATURE: Sig:2000 Subsig:0 Sev:2 ICMP
Echo Rply [172.30.6.2:0 -> 172.30.1.2:0]
```

```
10-16-2006      14:04:48    Local7.Warning          10.0.1.2    246:
*Oct 16 20:06:35.938: %IPS-4-SIGNATURE: Sig:2150 Subsig:0 Sev:2
Fragmented ICMP [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    245:
*Oct 16 20:06:35.938: %IPS-4-SIGNATURE: Sig:2151 Subsig:0 Sev:2 Large
ICMP [172.30.6.2:0 -> 172.30.1.2:0]

10-16-2006      14:04:48    Local7.Warning          10.0.1.2    244:
*Oct 16 20:06:35.934: %IPS-4-SIGNATURE: Sig:2000 Subsig:0 Sev:2 ICMP
Echo Rply [172.30.6.2:0 -> 172.30.1.2:0]
```

