

Lab 5.2.6 Configure a Cisco Router for IPSec using Digital Certificates

Objective

In this lab, the students will complete the following tasks:

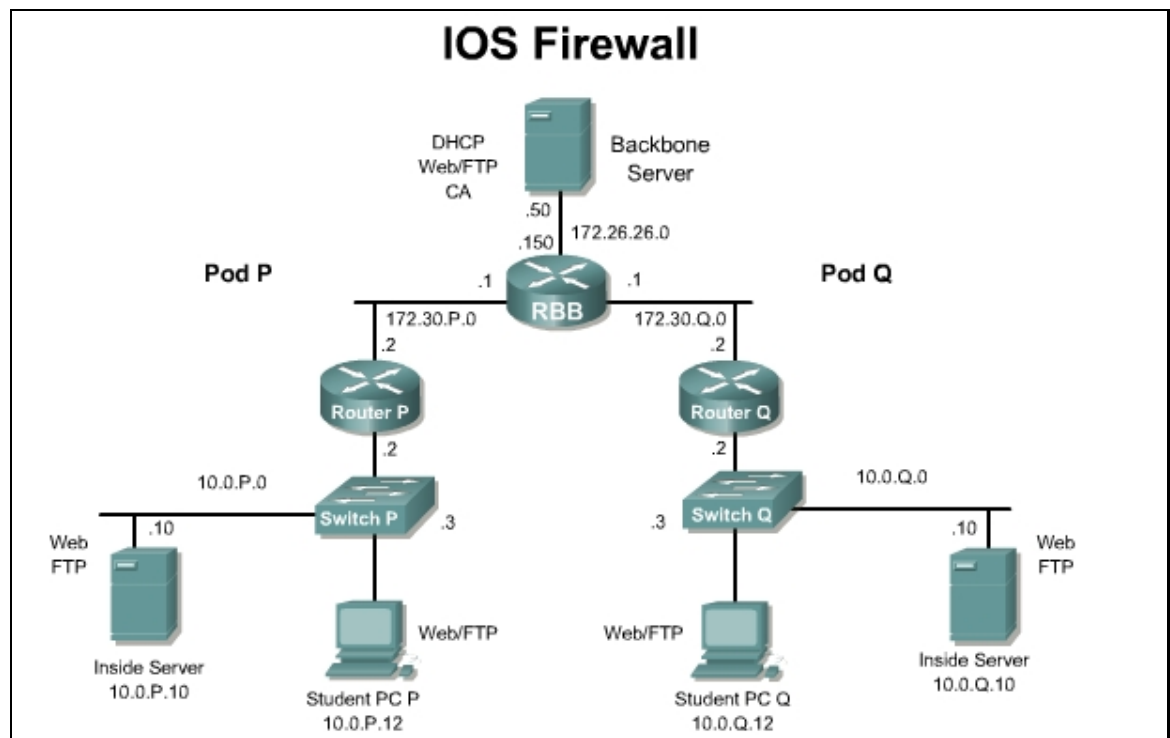
- Prepare for Internet Key Exchange (IKE) and IPSec configure certificate support
- Configure certificate support
- Configure IKE and IPSec
- Test and verify IPSec

Scenario

The XYZ Company has purchased Cisco routers and wants to create a secure Virtual Private Network (VPN) over the Internet between two sites. The company wants to configure a secure VPN gateway using IPSec between two Cisco routers using a certificate authority (CA) server.

Topology

This figure illustrates the lab network environment.



Preparation

Begin with the standard lab topology and verify the starting configuration on the pod router. Test the connectivity between the pod routers. Access the perimeter router console port using the terminal emulator on the Windows 2000 server. If desired, save the router configuration to a text file for later analysis. Refer back to the *Student Lab Orientation* if more help is needed.

Tools and resources

In order to complete the lab, the following is required:

- Standard IOS Firewall lab topology
- Console cable
- HyperTerminal
- Certificate Authority Server on the Backbone Server

Additional materials

Further information about the objectives covered in this lab can be found at:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7b2.html

Command list

In this lab exercise, the following commands will be used. Refer to this list if assistance or help is needed during the lab exercise.

Command	Description
<code>authentication</code>	Sets IKE authentication method.
<code>crl optional</code>	Specifies that the router can still accept other peers certificates if the CRL is not accessible.
<code>crypto ca authenticate labca</code>	Authenticates the CA server. Verifies the fingerprint of the CA server with the CA administrator.
<code>crypto ca enroll labca</code>	Enrolls to the CA server.
<code>crypto ca trustpoint</code>	Creates a name for the CA.
<code>crypto isakmp ?</code>	Displays crypto ISAKMP options.
<code>crypto isakmp enable</code>	Enables IKE on the router.
<code>crypto isakmp policy</code>	Used to create IKE policy.
<code>crypto isakmp trustpoint address</code>	Sets isakmp trustpoint to address.
<code>crypto key generate rsa usage-keys</code>	Generates RSA usage-keys.
<code>encryption</code>	Sets IKE encryption method.
<code>enrollment url http://vpnca</code>	Specifies the URL of the CA.

Command	Description
group	Sets Diffie-Hellman group number.
hash	Sets hash algorithm.
ip domain-name cisco.com	Defines the router domain name.
ip host	Defines the CA server's static hostname-to-IP address mapping.
lifetime	Sets lifetime in seconds and KB.
show crypto ca certificate	Shows the CA certificates.

Step 1 Prepare for IKE and IPSec

Complete the following steps to prepare for IPSec configuration:

- Determine the IKE and IPSec policy. In this exercise, use the default values except when directed to enter a specific value.
 - The IKE policy is to use Rivest, Shamir, and Adleman (RSA) signature keys.
 - The IPSec policy is to use Encapsulating Security Payload (ESP) mode with Data Encryption Standard (DES) encryption.
 - The IPSec policy is to encrypt all traffic between perimeter routers.
- Set the router time zone, calendar, and time. Make sure to coordinate the time with the instructor who will set the time on the CA server. A time mismatch between the router and CA server will create invalid certificates and IPSec negotiation will fail during the main mode exchange of isakmp.

Note The easiest way to ensure proper time is to set all devices to (GMT 0) time zone. Make sure the certificates on the Backbone server are valid and display as OK. This is accessible in the Certificate Authority administration application.

```
RouterP(config)#clock timezone zone hours [minutes]
```

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

- Verify connectivity with the peer router:

```
RouterP>ping 172.30.Q.2
```

(where Q = peer pod number)

- Ensure connectivity to the CA server from the pod router:

```
RouterP#ping 172.26.26.50
```

- Establish an HTTP session to the CA server. Test this capability from the student PC by opening a web browser and entering the following URL:

```
http://172.26.26.50/certsrv
```

- Turn on console logging to see the debug output:

```
RouterP(config)#logging console
```

Logging messages should appear on the console by default, but if they do not appear this feature can be turned on with the **terminal monitor** command.

```
RouterP#terminal monitor
```

Step 2 Configure CA Support

Complete the following steps to configure CA support on the Cisco router. Make sure to work with the CA server administrator to complete this portion of the lab exercise.

- a. Define the router domain name:

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

- b. Define the CA server static hostname-to-IP address mapping:

```
RouterP(config)#ip host vpnca 172.26.26.50
```

- c. Generate RSA usage-keys:

```
RouterP(config)#crypto key generate rsa usage-keys
```

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

- d. Perform the following substeps to configure the CA server trustpoint:

- i. Create a name for the CA and enter ca-trustpoint mode:

```
RouterP(config)#crypto ca trustpoint vpnca
```

- ii. Choose the registration authority mode:

```
RouterP(ca-trustpoint)#enrollment mode ra
```

- iii. Specify the URL of the CA:

- For a Microsoft CA:

```
RouterP(ca-trustpoint)#enrollment url  
http://vpnca/certsrv/mscep/mscep.dll
```

Note Make sure this is spelled exactly.

- iv. Specify that the router can still accept other peers certificates if the certificate revocation list (CRL) is not accessible:

```
RouterP(ca-trustpoint)#crl optional
```

- v. Exit CA configuration mode by pressing Control+Z and save the configuration:

```
RouterP(ca-trustpoint)#^Z
```

```
RouterP#copy running-config startup-config
```

- vi. Turn on PKI debugging to observe debug messages for the CA process:

```
RouterP#debug crypto pki messages
```

```
RouterP#debug crypto pki transactions
```

- vii. Authenticate the CA server. Verify the fingerprint of the CA server with the CA administrator:

```
RouterP#configure terminal
```

```
RouterP(config)#crypto ca authenticate vpnca
```

```
Certificate has the following attributes:
```

```
Fingerprint: 527D8DCA 4D52A047 C8DA1DAD D5368629
```

% Do you accept this certificate? [yes/no]: **y**

Note Because debug is on, several full screen messages flash by, which may require the student to press **Enter** to see this question.

- viii. Enroll to the CA server. Ensure that the CA administrator accepts the enrollment request. Answer the prompts as shown in the example.

WARNING Stop and ensure the instructor is ready to accept the enrollment request before continuing to the next step.

```
RouterP(config)#crypto ca enroll vpnca
```

```
% 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: cisco
```

```
Re-enter password: cisco
```

```
% The subject name in the certificate will be: rl.cisco.com
```

```
% Include the router serial number in the subject name? [yes/no]: n
```

```
% Include an IP address in the subject name? [yes/no]: n
```

```
Request certificate from CA? [yes/no]: y
```

```
% Certificate request sent to Certificate Authority
```

```
% The certificate request fingerprint will be displayed.
```

```
% The 'show crypto ca certificate' command will also show the fingerprint.
```

- ix. Verify the CA certificates:

```
RouterP(config)#exit
```

```
RouterP#copy running-config startup-config
```

```
RouterP#show crypto ca certificate
```

Step 3 Configure IKE

Complete the following steps to configure IKE on the Cisco router. Make sure to work with the members of the peer pod to complete this section of the lab.

Note While entering commands, notice when the command line prompt changes. This helps distinguish what configuration mode is active.

- a. Enable IKE/ISAKMP on the router:

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

- b. Create an IKE policy to use RSA signatures by completing the following substeps:

- i. Set the policy priority:

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

- ii. Set authentication to use RSA signatures:

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

- iii. Set the IKE encryption:

```
RouterP(config-isakmp)#encryption des
```

1. What other encryption choice can be used?
-

- iv. Set the Diffie-Hellman group:

```
RouterP(config-isakmp)#group 1
```

2. What would be the benefit of using Diffie-Hellman Group 2?
-

- v. Set the hash algorithm:

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

- vi. Set the IKE security association (SA) lifetime:

```
RouterP(config-isakmp)#lifetime 86400
```

- vii. Exit config-isakmp mode:

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

Step 4 Configure IPSec

Complete the following steps to configure IPSec on the Cisco router.

- a. Configure transform sets and security association parameters

Complete the following steps to configure transform sets and security association (SA) parameters:

- b. View the available crypto IPSec command options:

```
RouterP(config)#crypto ipsec ?
```

- c. Check the transform set options:

```
RouterP(config)#crypto ipsec transform-set ?
```

- d. Define a transform set. Use the following parameters:

- Transform name = **mine**
- ESP protocols = **des**
- Mode = **tunnel**

```
RouterP(config)#crypto ipsec transform-set mine esp-des
```

- e. Set the mode to tunnel:

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

- f. Exit configuration mode by pressing Control+Z:

```
RouterP(cfg-crypto-trans)#^Z
```

- g. Check the configuration:

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

- h. Configure crypto access lists

Complete the following steps to configure the crypto access lists. Create an access list to select the traffic to protect. The access list should encrypt traffic between perimeter routers. Use the following parameters:

- Traffic permitted = **all**
- Peer address = **Peer router outside interface**
- Access list number = **102**
- Protocol = **IP**

- i. Ensure that configuration mode is enabled:

```
RouterP(config)#config terminal
```

- j. Configure the access list:

```
RouterP(config)#access-list 102 permit ip host 172.30.P.2 host
172.30.Q.2
```

(where P = pod number and Q = peer's pod number)

- k. Configure crypto maps

Complete the following steps to configure a crypto map. Use the following parameters:

- Name of map = **mymap**
- Number of map = **10**
- Key exchange type = **isakmp**
- Peer = **172.30.Q.2**
- Transform set = **mine**
- Match address = **102**

- l. Set the name of the map, the map number, and the type of key exchange to be used:

```
RouterP(config)#crypto map mymap 10 ipsec-isakmp
```

- m. Specify the extended access list to use with this map:

```
RouterP(config-crypto-map)#match address 102
```

- n. Specify the transform set defined earlier:

```
RouterP(config-crypto-map)#set transform-set mine
```

- o. Assign the VPN peer using the hostname or IP address of the peer:

```
RouterP(config-crypto-map)#set peer 172.30.Q.2
```

- p. Exit crypto-map configuration mode:

```
RouterP(config-crypto-map)#exit
```

- q. Apply the crypto map to an interface

Complete the following steps to assign the crypto map to the appropriate router interface. Use the following parameters:

- Interface to configure = **FastEthernet0/1**

- Crypto map to use = **mymap**
- r. Access the interface configuration mode:

```
RouterP(config)#interface FastEthernet0/1
```
- s. Assign the crypto map to the interface:

```
RouterP(config-if)#crypto map mymap
```
- t. Exit configuration crypto mode by pressing Contol+Z:

```
RouterP(config-if)#^Z
```

Step 5 Test and Verify IPsec

Complete the following steps to verify and test the IPsec configuration. Coordinate test with the peer router pod group:

- a. Display the configured IKE policies:

```
RouterP#show crypto isakmp policy
Protection suite of priority 110
    encryption algorithm:  DES - Data Encryption Standard (56 bit
keys).
    hash algorithm:        Message Digest 5
    authentication method: Rivest-Shamir-Adelman Signature
    Diffie-Hellman group:  #1 (768 bit)
    lifetime:              86400 seconds, no volume limit
Default protection suite
    encryption algorithm:  DES - Data Encryption Standard (56 bit
keys).
    hash algorithm:        Secure Hash Standard
    authentication method: Rivest-Shamir-Adelman Signature
    Diffie-Hellman group:  #1 (768 bit)
    lifetime:              86400 seconds, no volume limit
```

- b. Display the configured transform sets:

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

- c. Display the configured crypto maps:

```
RouterP#show crypto map
Crypto Map "mymap" 10 ipsec-isakmp
    Peer = 172.30.Q.2
    Extended IP access list 102
    access-list 102 permit ip host 172.30.P.2 host 172.30.Q.2
    Current peer: 172.30.Q.2
    Security association lifetime: 4608000 kilobytes/3600
seconds
    PFS (Y/N): N
    Transform sets={ mine, }
```


Interfaces using crypto map mymap:

FastEthernet0/1

- d. Display the current state of the IPSec SAs. The IPSec SAs may have already been established by routing traffic.

```
RouterP#show crypto ipsec sa
```

```
interface: FastEthernet0/1
```

```
  Crypto map tag: mymap, local addr. 172.30.1.2
```

```
    local ident (addr/mask/prot/port):  
    (172.30.1.2/255.255.255.255/0/0)
```

```
    remote ident (addr/mask/prot/port):  
    (172.30.2.2/255.255.255.255/0/0)
```

```
    current_peer: 172.30.2.2
```

```
      PERMIT, flags={origin_is_acl,}
```

```
    #pkts encaps: 21, #pkts encrypt: 21, #pkts digest 0
```

```
    #pkts decaps: 21, #pkts decrypt: 21, #pkts verify 0
```

```
    #send errors 0, #recv errors 0
```

```
local crypto endpt.: 172.30.1.2, remote crypto endpt.: 172.30.2.2
```

```
  path mtu 1500, media mtu 1500
```

```
  current outbound spi: 8AE1C9C
```

```
inbound esp sas:
```

```
  spi: 0x1B781456(460854358)
```

```
    transform: esp-des ,
```

```
    in use settings ={Tunnel, }
```

```
    slot: 0, conn id: 17, crypto map: mymap
```

```
    sa timing: remaining key lifetime (k/sec): (4607997/3107)
```

```
    IV size: 8 bytes
```

```
    replay detection support: N
```

```
inbound ah sas:
```

```
outbound esp sas:
```

```
  spi: 0x8AE1C9C(145628316)
```

```
    transform: esp-des ,
```

```
    in use settings ={Tunnel, }
```

```
    slot: 0, conn id: 18, crypto map: mymap
```

```
    sa timing: remaining key lifetime (k/sec): (4607997/3107)
```

```
    IV size: 8 bytes
```

```
replay detection support: N
```

```
outbound ah sas:
```

- e. Clear any existing SAs:

```
RouterP#clear crypto sa
```

- f. Enable debug output for IPSec events:

```
RouterP#debug crypto ipsec
```

- g. Enable debug output for ISAKMP events:

```
RouterP#debug crypto isakmp
```

- h. Initiate a ping to the peer pod perimeter router. Observe the IKE and IPSec debug output.

```
RouterP#ping 172.30.Q.2
```

- i. Verify IKE and IPSec SAs. Note the number of packets encrypted and decrypted when viewing the IPSec SAs.

```
RouterP#show crypto isakmp sa
```

dst	src	state	conn-id	slot
172.30.1.2	172.30.2.2	QM_IDLE	1	0

The sample output below will indicate if there is a misconfiguration with isakmp. This could also indicate a problem with certificate validity on either router. The MM Exchange indicates the router cannot go beyond the main mode exchange of isakmp. This problem will also be indicated by a continuous looping of debug output of isakmp exchange messages.

dst	src	state	conn-id	slot
172.30.1.2	172.30.2.2	MM Exchange	1	0

```
RouterP#show crypto ipsec sa
```

```
interface: FastEthernet0/1
```

```
  Crypto map tag: mymap, local addr. 172.30.1.2
```

```
    local ident (addr/mask/prot/port):  
    (172.30.1.2/255.255.255.255/0/0)
```

```
    remote ident (addr/mask/prot/port):  
    (172.30.2.2/255.255.255.255/0/0)
```

```
    current_peer: 172.30.2.2
```

```
      PERMIT, flags={origin_is_acl,}
```

```
    #pkts encaps: 26, #pkts encrypt: 26, #pkts digest 0
```

```
    #pkts decaps: 26, #pkts decrypt: 26, #pkts verify 0
```

```
    #send errors 0, #recv errors 0
```

```
local crypto endpt.: 172.30.1.2, remote crypto endpt.: 172.30.2.2
```

```
  path mtu 1500, media mtu 1500
```

```
  current outbound spi: 8AE1C9C
```

```
inbound esp sas:
```

```

spi: 0x1B781456(460854358)
  transform: esp-des ,
  in use settings ={Tunnel, }
  slot: 0, conn id: 17, crypto map: mymap
  sa timing: remaining key lifetime (k/sec): (4607996/2963)
  IV size: 8 bytes
  replay detection support: N

```

inbound ah sas:

outbound esp sas:

```

spi: 0x8AE1C9C(145628316)
  transform: esp-des ,
  in use settings ={Tunnel, }
  slot: 0, conn id: 18, crypto map: mymap
  sa timing: remaining key lifetime (k/sec): (4607996/2963)
  IV size: 8 bytes
  replay detection support: N

```

outbound ah sas:

- j. Ensure that encryption is working between the routers by first generating additional traffic, and then by observing that the packets encrypted and decrypted counter has incremented:

```
RouterP#ping 172.30.Q.2
```

```
RouterP#show crypto ipsec sa
```

```
interface: FastEthernet0/1
```

```
  Crypto map tag: mymap, local addr. 172.30.1.2
```

```

    local ident (addr/mask/prot/port):
    (172.30.1.2/255.255.255.255/0/0)

```

```

    remote ident (addr/mask/prot/port):
    (172.30.2.2/255.255.255.255/0/0)

```

```
current_peer: 172.30.2.2:500
```

```
  PERMIT, flags={origin_is_acl,}
```

```
#pkts encaps: 31, #pkts encrypt: 31, #pkts digest 0
```

```
#pkts decaps: 31, #pkts decrypt: 31, #pkts verify 0
```

```
#send errors 0, #recv errors 0
```

```

    local crypto endpt.: 172.30.1.2, remote crypto endpt.:
    172.30.2.2

```

```
path mtu 1500, media mtu 1500
```

```
current outbound spi: 8AE1C9C
```

```

inbound esp sas:
  spi: 0x1B781456(460854358)
    transform: esp-des ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 17, crypto map: mymap
    sa timing: remaining key lifetime (k/sec): (4607995/2954)
    IV size: 8 bytes
    replay detection support: N

inbound ah sas:

outbound esp sas:
  spi: 0x8AE1C9C(145628316)
    transform: esp-des ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 18, crypto map: mymap
    sa timing: remaining key lifetime (k/sec): (4607996/2954)
    IV size: 8 bytes
    replay detection support: N

outbound ah sas:

```

Step 6 Fine Tuning the ACL

Fine-tune the crypto access lists used to determine interesting traffic to encrypt traffic only between the internal student PCs. Remember to work with the peer pod group to make the access lists symmetrical between the perimeter routers. Ensure that desired traffic is encrypted between the peers.

- a. Remove the previously configured access list:

```
RouterP(config)#no access-list 102
```

- b. Configure a new access list for the Windows 2000 servers:

```
RouterP(config)#access-list 102 permit ip host 10.0.P.12 host
10.0.Q.12
```

(where P = pod number, and Q = peer pod number)

- c. Verify the configuration by connecting to the web server at 10.0.Q.12 using the browser on the Student PC.

(where Q = peer pod number)