SAHYOG PRATISHTHAN

Sahyog for Advanced Career

Sahyog College of Management Studies , Thane(W)

Affiliated to Mumbai University

Course : BSC (Information Technology)

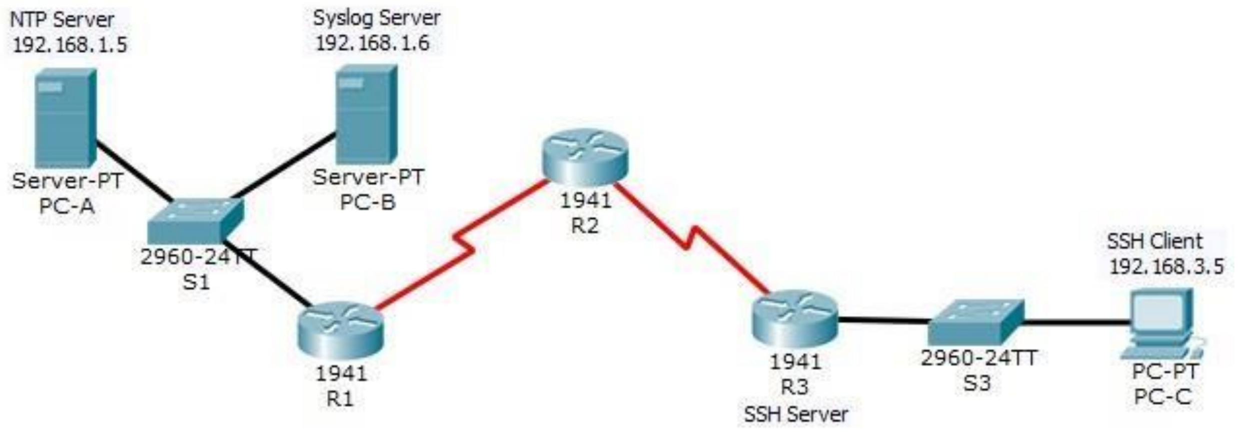
Semester : VI

Sub: Information Security

Lab Manual

Practical 1: Packet Tracer - Configure Cisco Routers for Syslog, NTP, and SSH Operations

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.5	255.255.255.0	192.168.1.1	S1 F0/6
PC-B	NIC	192.168.1.6	255.255.255.0	192.168.1.1	S2 F0/18
PC-C	NIC	192.168.3.5	255.255.255.0	192.168.3.1	S3 F0/18

Objectives

- Configure OSPF MD5 authentication.
- Configure NTP.
- Configure routers to log messages to the syslog server.
- Configure R3 to support SSH connections.

Background / Scenario

In this activity, you will configure OSPF MD5 authentication for secure routing updates.

The NTP Server is the master NTP server in this activity. You will configure authentication on the NTP server and the routers. You will configure the routers to allow the software clock to be synchronized by NTP to the time server. Also, you will configure the routers to periodically update the hardware clock with the time learned from NTP.

The Syslog Server will provide message logging in this activity. You will configure the routers to identify the remote host (Syslog server) that will receive logging messages.

You will need to configure timestamp service for logging on the routers. Displaying the correct time and date in Syslog messages is vital when using Syslog to monitor a network.

You will configure R3 to be managed securely using SSH instead of Telnet. The servers have been preconfigured for NTP and Syslog services respectively. NTP will not require authentication. The routers have been pre-configured with the following passwords:

- Enable password: **ciscoenpa55**
- Password for vty lines: **ciscovtypa55**

Note: Note: MD5 is the strongest encryption supported in the version of Packet Tracer used to develop this activity (v6.2). Although MD5 has known vulnerabilities, you should use the encryption that meets the security requirements of your organization. In this activity, the security requirement specifies MD5.

Part 1: Configure OSPF MD5 Authentication

Step 1: Test connectivity. All devices should be able to ping all other IP addresses.

Step 2: Configure OSPF MD5 authentication for all the routers in area 0. Configure

OSPF MD5 authentication for all the routers in area 0.

```
R1(config)# router ospf 1
R1(config-router)# area 0 authentication message-digest
R2(config)# router ospf 1
R2(config-router)# area 0 authentication message-digest

R3(config)# router ospf 1
R3(config-router)# area 0 authentication message-digest
```

Step 3: Configure the MD5 key for all the routers in area 0. Configure an MD5 key on the serial

interfaces on **R1**, **R2** and **R3**. Use the password **MD5pa55** for key **1**.

```
R1(config)# interface s0/0/0
R1(config-if)# ip ospf message-digest-key 1 md5 MD5pa55

R2(config)# interface s0/0/0
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55
R2(config-if)# interface s0/0/1
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55

R3(config)# interface s0/0/1
R3(config-if)# ip ospf message-digest-key 1 md5 MD5pa55
```

Step 4: Verify configurations.

a. Verify the MD5 authentication configurations using the commands **show ip ospf interface**. b.

Verify end-to-end connectivity.

Part 2: Configure NTP

Step 1: Enable NTP authentication on PC-A.

a. On **PC-A**, click **NTP** under the Services tab to verify NTP service is enabled.

b. To configure NTP authentication, click **Enable** under Authentication. Use key **1** and password **NTPpa55** for authentication.

Step 2: Configure R1, R2, and R3 as NTP clients.

```
R1(config)# ntp server 192.168.1.5
R2(config)# ntp server 192.168.1.5
R3(config)# ntp server 192.168.1.5
```

Verify client configuration using the command **show ntp status**.

Step 3: Configure routers to update hardware clock. Configure **R1**, **R2**, **and R3** to periodically

update the hardware clock with the time learned from NTP.

```
R1(config)# ntp update-calendar
R2(config)# ntp update-calendar
R3(config)# ntp update-calendar
```

Exit global configuration and verify that the hardware clock was updated using the command **show clock**.

Step 4: Configure NTP authentication on the routers. Configure NTP

authentication on **R1**, **R2**, and **R3** using key **1** and password **NTPpa55**.

```
R1(config)# ntp authenticate
R1(config)# ntp trusted-key 1
R1(config)# ntp authentication-key 1 md5 NTPpa55

R2(config)# ntp authenticate
R2(config)# ntp trusted-key 1
R2(config)# ntp authentication-key 1 md5 NTPpa55

R3(config)# ntp authenticate
R3(config)# ntp trusted-key 1
R3(config)# ntp authentication-key 1 md5 NTPpa55
```

Step 5: Configure routers to timestamp log messages.

Configure timestamp service for logging on the routers.

```
R1(config)# service timestamps log datetime msec
R2(config)# service timestamps log datetime msec
R3(config)# service timestamps log datetime msec
```

Part 3: Configure Routers to Log Messages to the Syslog Server

Step 1: Configure the routers to identify the remote host (Syslog Server) that will receive logging messages.

```
R1(config)# logging host 192.168.1.6
R2(config)# logging host 192.168.1.6
R3(config)# logging host 192.168.1.6
```

The router console will display a message that logging has started.

Step 2: Verify logging configuration.

Use the command **show logging** to verify logging has been enabled.

Step 3: Examine logs of the Syslog Server.

From the **Services** tab of the **Syslog Server**'s dialogue box, select the **Syslog** services button. Observe the logging messages received from the routers.

Note: Log messages can be generated on the server by executing commands on the router. For example, entering and exiting global configuration mode will generate an informational configuration message. You may need to click a different service and then click **Syslog** again to refresh the message display.

Part 4: Configure R3 to Support SSH Connections

Step 1: Configure a domain name.

Configure a domain name of **ccnasecurity.com** on **R3**.

```
R3(config)# ip domain-name ccnasecurity.com
```

Step 2: Configure users for login to the SSH server on R3.

Create a user ID of **SSHadmin** with the highest possible privilege level and a secret password of **ciscosshpa55**.

```
R3(config)# username SSHadmin privilege 15 secret ciscosshpa55
```

Step 3: Configure the incoming vty lines on R3.

Use the local user accounts for mandatory login and validation. Accept only SSH connections.

```
R3(config)# line vty 0 4
R3(config-line)# login local
R3(config-line)# transport input ssh
```

Step 4: Erase existing key pairs on R3.

Any existing RSA key pairs should be erased on the router.

```
R3(config)# crypto key zeroize rsa
```

Note: If no keys exist, you might receive this message: `% No Signature RSA Keys found in configuration`.

Step 5: Generate the RSA encryption key pair for R3.

The router uses the RSA key pair for authentication and encryption of transmitted SSH data. Configure the RSA keys with a modulus of **1024**. The default is 512, and the range is from 360 to 2048.

```
R3(config)# crypto key generate rsa
The name for the keys will be: R3.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.


How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

Note: The command to generate RSA encryption key pairs for **R3** in Packet Tracer differs from those used in the lab.

Step 6: Verify the SSH configuration.

Use the **show ip ssh** command to see the current settings. Verify that the authentication timeout and retries are at their default values of 120 and 3.

Step 7: Configure SSH timeouts and authentication parameters.

The default SSH timeouts and authentication parameters can be altered to be more restrictive. Set the timeout to **90** seconds, the number of authentication retries to **2**, and the version to **2**.

```
R3(config)# ip ssh time-out 90
R3(config)# ip ssh authentication-retries 2
R3(config)# ip ssh version 2
```

Issue the **show ip ssh** command again to confirm that the values have been changed.

Step 8: Attempt to connect to R3 via Telnet from PC-C.

Open the Desktop of **PC-C**. Select the Command Prompt icon. From **PC-C**, enter the command to connect to **R3** via Telnet.

```
PC> telnet 192.168.3.1
```

This connection should fail because **R3** has been configured to accept only SSH connections on the virtual terminal lines.

Step 9: Connect to R3 using SSH on PC-C.

Open the Desktop of **PC-C**. Select the Command Prompt icon. From **PC-C**, enter the command to connect to R3 via SSH. When prompted for the password, enter the password configured for the administrator **ciscosshpa55**.

```
PC> ssh –l SSHadmin 192.168.3.1
```

Step 10: Connect to R3 using SSH on R2.

To troubleshoot and maintain **R3**, the administrator at the ISP must use SSH to access the router CLI. From the CLI of **R2**, enter the command to connect to **R3** via SSH version **2** using the **SSHadmin** user account. When prompted for the password, enter the password configured for the administrator: **ciscosshpa55**.

```
R2# ssh –v 2 –l SSHadmin 10.2.2.1
```

Step 11: Check results.

Your completion percentage should be 100%. Click **Check Results** to view the feedback and verification of which required components have been completed.

!!!Scripts for R1!!!!

```
conf  t  interface
s0/0/0
 ip ospf message-digest-key 1 md5 MD5pa55
router ospf 1
area 0 authentication message-digest
service timestamps log datetime msec
logging 192.168.1.6
ntp server 192.168.1.5 ntp update-
calendar
ntp authentication-key 1 md5 NTPpa55
ntp authenticate ntp trusted-key 1
end
```

!!!Scripts for R2!!!!

```
conf  t  interface
s0/0/0
 ip ospf message-digest-key 1 md5 MD5pa55
interface s0/0/1
 ip ospf message-digest-key 1 md5 MD5pa55
router ospf 1
area 0 authentication message-digest
service timestamps log datetime msec
logging 192.168.1.6
ntp server 192.168.1.5 ntp update-
calendar
ntp authentication-key 1 md5 NTPpa55
ntp authenticate ntp trusted-key 1
end
```

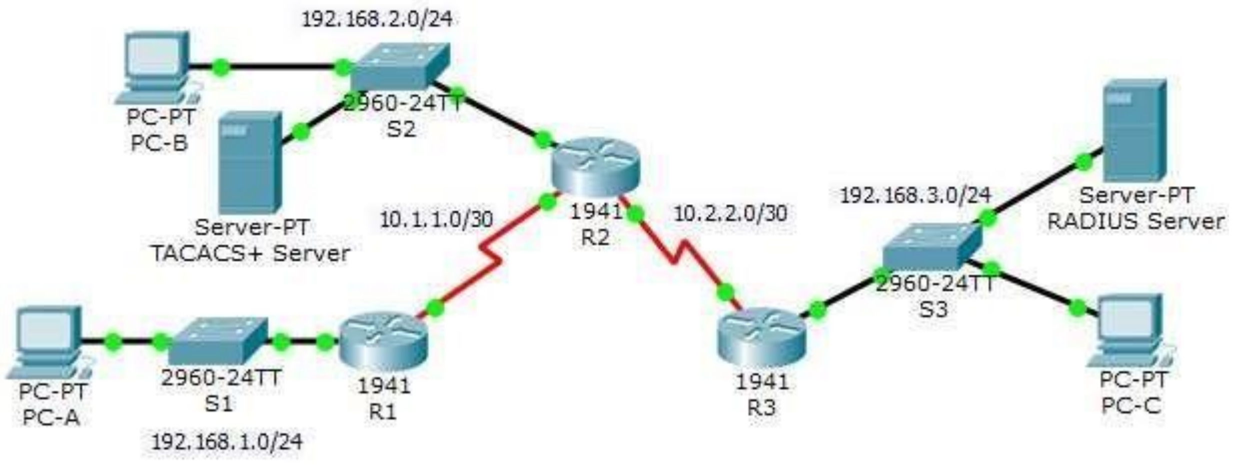
!!!Scripts for R3!!!!

```
conf  t  interface
s0/0/1
 ip ospf message-digest-key 1 md5 MD5pa55
router ospf 1
 area 0 authentication message-digest
service timestamps log datetime msec
logging 192.168.1.6
ntp server 192.168.1.5 ntp update-
calendar
ntp authentication-key 1 md5 NTPpa55 ntp
authenticate
ntp trusted-key 1
```

```
ip domain-name ccnasecurity.com username SSHadmin
privilege 15 secret ciscosshpa55  line vty 0 4
login local  transport input ssh crypto key
zeroize rsa crypto key generate rsa
1024 ip ssh time-out 90 ip ssh
authentication-retries 2
ip ssh version 2
end
```

Practical 2: Packet Tracer - Configure AAA Authentication on Cisco Routers

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/1
	S0/0/0 (DCE)	10.1.1.2	255.255.255.252	N/A	N/A
R2	G0/0	192.168.2.1	255.255.255.0	N/A	S2 F0/2
	S0/0/0	10.1.1.1	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.1	255.255.255.252	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.2	255.255.255.252	N/A	N/A
TACACS+ Server	NIC	192.168.2.2	255.255.255.0	192.168.2.1	S2 F0/6
RADIUS Server	NIC	192.168.3.2	255.255.255.0	192.168.3.1	S3 F0/1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/2
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1	S2 F0/1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Objectives

- Configure a local user account on R1 and configure authenticate on the console and vty lines using local AAA.
- Verify local AAA authentication from the R1 console and the PC-A client.

- Configure server-based AAA authentication using TACACS+.
- Verify server-based AAA authentication from the PC-B client.
- Configure server-based AAA authentication using RADIUS.
- Verify server-based AAA authentication from the PC-C client.

Background / Scenario

The network topology shows routers R1, R2 and R3. Currently, all administrative security is based on knowledge of the enable secret password. Your task is to configure and test local and server-based AAA solutions.

You will create a local user account and configure local AAA on router R1 to test the console and vty logins. o

User account: **Admin1** and password **admin1pa55**

You will then configure router R2 to support server-based authentication using the TACACS+ protocol. The TACACS+ server has been pre-configured with the following:

o Client: **R2** using the keyword **tacacspa55** o User account: **Admin2**

and password **admin2pa55**

Finally, you will configure router R3 to support server-based authentication using the RADIUS protocol. The RADIUS server has been pre-configured with the following:

o Client: **R3** using the keyword **radiuspa55** o User account: **Admin3**

and password **admin3pa55**

The routers have also been pre-configured with the following:

o Enable secret password: **ciscoenpa55**

o OSPF routing protocol with MD5 authentication using password: **MD5pa55**

Note: The console and vty lines have not been pre-configured.

Note: IOS version 15.3 uses SCRYPT as a secure encryption hashing algorithm; however, the IOS version that is currently supported in Packet Tracer uses MD5. Always use the most secure option available on your equipment.

Part 1: Configure Local AAA Authentication for Console Access on R1

Step 1: Test connectivity.

- Ping from **PC-A** to **PC-B**.
- Ping from **PC-A** to **PC-C**.
- Ping from **PC-B** to **PC-C**.

Step 2: Configure a local username on R1.

Configure a username of **Admin1** with a secret password of **admin1pa55**.

```
R1(config)# username Admin1 secret admin1pa55
```

Step 3: Configure local AAA authentication for console access on R1.

Enable AAA on R1 and configure AAA authentication for the console login to use the local database.

```
R1(config)# aaa new-model
```

```
R1(config)# aaa authentication login default local
```

Step 4: Configure the line console to use the defined AAA authentication method.

Enable AAA on **R1** and configure AAA authentication for the console login to use the default method list.

```
R1(config)# line console 0
R1(config-line)# login authentication default
```

Step 5: Verify the AAA authentication method.

Verify the user EXEC login using the local database.

```
R1(config-line)# end
%SYS-5-CONFIG_I: Configured from console by console
R1# exit

R1 con0 is now available Press
RETURN to get started.

************ AUTHORIZED ACCESS ONLY *************
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.



User Access Verification

Username: Admin1
Password: admin1pa55 R1>
```

Part 2: Configure Local AAA Authentication for vty Lines on R1

Step 1: Configure domain name and crypto key for use with SSH.

a. Use ccnasecurity.com as the domain name on R1.

```
R1(config)# ip domain-name ccnasecurity.com
```

b. Create an RSA crypto key using 1024 bits.

```
R1(config)# crypto key generate rsa

Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

Step

2: Configure a named list AAA authentication method for the vty lines on R1.

Configure a named list called **SSH-LOGIN** to authenticate logins using local AAA.

```
R1(config)# aaa authentication login SSH-LOGIN local
```

Step 3: Configure the vty lines to use the defined AAA authentication method.

Configure the vty lines to use the named AAA method and only allow SSH for remote access.

```
R1(config)# line vty 0 4
R1(config-line)# login authentication SSH-LOGIN
R1(config-line)# transport input ssh R1(config-line)#
end
```

Step 4: Verify the AAA authentication method.

Verify the SSH configuration SSH to **R1** from the command prompt of **PC-A**..

```
PC> ssh –l Admin1 192.168.1.1
Open
Password: admin1pa55
```

Part 3: Configure Server-Based AAA Authentication Using TACACS+ on R2

Step 1: Configure a backup local database entry called Admin.

For backup purposes, configure a local username of **Admin2** and a secret password of **admin2pa55**.

```
R2(config)# username Admin2 secret admin2pa55
```

Step 2: Verify the TACACS+ Server configuration.

Click the TACACS+ Server. On the Services tab, click **AAA**. Notice that there is a Network configuration entry for **R2** and a User Setup entry for **Admin2**.

Step 3: Configure the TACACS+ server specifics on R2.

Configure the AAA TACACS server IP address and secret key on **R2**.

Note: The commands **tacacs-server host** and **tacacs-server key** are deprecated. Currently, Packet Tracer does not support the new command **tacacs server**.

```
R2(config)# tacacs-server host 192.168.2.2 R2(config)#
tacacs-server key tacacspa55
```

Step 4: Configure AAA login authentication for console access on R2.

Enable AAA on **R2** and configure all logins to authenticate using the AAA TACACS+ server. If it is not available, then use the local database.

```
R2(config)# aaa new-model
R2(config)# aaa authentication login default group tacacs+ local
```

Step 5: Configure the line console to use the defined AAA authentication method.

Configure AAA authentication for console login to use the default AAA authentication method.

```
R2(config)# line console 0
R2(config-line)# login authentication default
```

Step 6: Verify the AAA authentication method.

Verify the user EXEC login using the AAA TACACS+ server.

```
R2(config-line)# end
%SYS-5-CONFIG_I: Configured from console by console
R2# exit

R2 con0 is now available Press
RETURN to get started.

************ AUTHORIZED ACCESS ONLY *************
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.



User Access Verification

Username: Admin2
Password: admin2pa55
R2>
```

Part 4: Configure Server-Based AAA Authentication Using RADIUS on R3

Step 1: Configure a backup local database entry called Admin.

For backup purposes, configure a local username of **Admin3** and a secret password of **admin3pa55**.

```
R3(config)# username Admin3 secret admin3pa55
```

Step 2: Verify the RADIUS Server configuration.

Click the RADIUS Server. On the Services tab, click **AAA**. Notice that there is a Network configuration entry for **R3** and a User Setup entry for **Admin3**.

Step 3: Configure the RADIUS server specifics on R3.

Configure the AAA RADIUS server IP address and secret key on **R3**.

Note: The commands **radius-server host** and **radius-server key** are deprecated. Currently Packet Tracer does not support the new command **radius server**.

```
R3(config)# radius-server host 192.168.3.2 R3(config)#
radius-server key radiuspa55
```

Step 4: Configure AAA login authentication for console access on R3.

Enable AAA on **R3** and configure all logins to authenticate using the AAA RADIUS server. If it is not available, then use the local database.

```
R3(config)# aaa new-model
R3(config)# aaa authentication login default group radius local
```

Step 5: Configure the line console to use the defined AAA authentication method.

Configure AAA authentication for console login to use the default AAA authentication method.

```
R3(config)# line console 0
```

```
R3(config-line)# login authentication default
```

Step 6: Verify the AAA authentication method.

Verify the user EXEC login using the AAA RADIUS server.

```
R3(config-line)# end
%SYS-5-CONFIG_I: Configured from console by console
R3# exit

R3 con0 is now available Press
RETURN to get started.

************ AUTHORIZED ACCESS ONLY *************
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.



User Access Verification

Username: Admin3
Password: admin3pa55 R3>
```

Step 7: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.

!!!Script for R1

```
!!!Part 1 config t username Admin1
secret admin1pa55 aaa new-model aaa
authentication login default local
line console 0
 login authentication default
!!!Part 2
ip domain-name ccnasecurity.com crypto
key generate rsa
1024
aaa authentication login SSH-LOGIN local
line vty 0 4  login authentication SSH-
LOGIN  transport input ssh
```

!!!!Script for R2

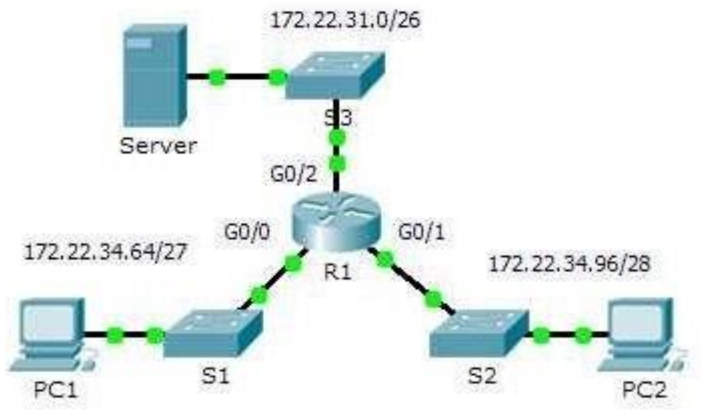
```
conf t
username Admin2 secret admin2pa55
tacacs-server host 192.168.2.2
tacacs-server key tacacspa55 aaa
new-model
aaa authentication login default group tacacs+ local
line console 0  login authentication default
```

!!!!Script for R3

```
conf t username Admin3 secret admin3pa55 radius-
server host 192.168.3.2
radius-server key radiuspa55 aaa new-model aaa
authentication login default group radius local
line console 0 login authentication default
```

Practical 3: Configuring Extended ACLs - Scenario 1

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	172.22.34.65	255.255.255.224	N/A
	G0/1	172.22.34.97	255.255.255.240	N/A
	G0/2	172.22.34.1	255.255.255.192	N/A
Server	NIC	172.22.34.62	255.255.255.192	172.22.34.1
PC1	NIC	172.22.34.66	255.255.255.224	172.22.34.65
PC2	NIC	172.22.34.98	255.255.255.240	172.22.34.97

Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Part 2: Configure, Apply and Verify an Extended Named ACL

Background / Scenario

Two employees need access to services provided by the server. **PC1** needs only FTP access while **PC2** needs only web access. Both computers are able to ping the server, but not each other.

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Step 1: Configure an ACL to permit FTP and ICMP.

a. From global configuration mode on **R1**, enter the following command to determine the first valid number for an extended access list.

```
R1(config)# access-list ?
  <1-99>      IP standard access list
  <100-199>  IP extended access list
```

b. Add **100** to the command, followed by a question mark.

```
R1(config)#  access-list  100  ?
deny   Specify  packets  to  reject
permit  Specify packets to  forward
remark  Access list entry comment
```

c. To permit FTP traffic, enter **permit,** followed by a question mark.

```
R1(config)# access-list 100 permit ?
ahp    Authentication Header Protocol
eigrp  Cisco's EIGRP routing protocol
esp    Encapsulation Security Payload
gre    Cisco's GRE tunneling   icmp
Internet Control Message Protocol   ip
Any Internet Protocol   ospf   OSPF
routing protocol   tcp    Transmission
Control Protocol   udp    User Datagram
Protocol
```

d. This ACL permits FTP and ICMP. ICMP is listed above, but FTP is not, because FTP uses TCP. Therefore,enter **tcp** to further refine the ACL help.

```
R1(config)# access-list 100 permit tcp ?
  A.B.C.D  Source  address  any
Any  source  host   host     A
single source host
```

e. Notice that we could filter just for **PC1** by using the **host** keyword or we could allow **any** host. In this case, any device is allowed that has an address belonging to the 172.22.34.64/27 network. Enter the network address, followed by a question mark.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 ?
  A.B.C.D  Source wildcard bits
```

f. Calculate the wildcard mask determining the binary opposite of a subnet mask.

```
11111111.11111111.11111111.11100000 = 255.255.255.224
00000000.00000000.00000000.00011111 = 0.0.0.31
```

g. Enter the wildcard mask, followed by a question mark.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 ?
  A.B.C.D  Destination address  any     Any destination
host   eq      Match only packets on a given port number
gt       Match  only  packets  with  a  greater  port  number
host     A  single  destination  host   lt      Match  only
packets  with  a  lower  port  number   neq      Match  only
packets  not  on  a  given  port  number   range   Match  only
packets in the range of port numbers
```

h. Configure the destination address. In this scenario, we are filtering traffic for a single destination, which is the server. Enter the **host** keyword followed by the server's IP address.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62
?
  dscp            Match  packets  with  given  dscp  value   eq
Match  only  packets  on  a  given  port  number   established
established   gt           Match only packets with a greater
```

```
port number  lt           Match only packets with a lower port
number  neq           Match only packets not on a given port
number  precedence  Match packets with given precedence value
range        Match only packets in the range of port numbers
  <cr>
```

i. Notice that one of the options is **<cr>** (carriage return). In other words, you can press **Enter** and the statement would permit all TCP traffic. However, we are only permitting FTP traffic; therefore, enter the **eq** keyword, followed by a question mark to display the available options. Then, enter **ftp** and press **Enter**.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62
eq ?
  <0-65535>  Port number    ftp        File
Transfer Protocol (21)   pop3       Post Office
Protocol v3 (110)   smtp       Simple Mail
Transport Protocol (25)   telnet     Telnet (23)
  www        World Wide Web (HTTP, 80)
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host
172.22.34.62 eq ftp
```

j. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC1** to **Server**. Note that the access list number remains the same and no particular type of ICMP traffic needs to be specified.

```
R1(config)# access-list 100 permit icmp 172.22.34.64 0.0.0.31 host
172.22.34.62
```

k. All other traffic is denied, by default.

Step 2: Apply the ACL on the correct interface to filter traffic.

From **R1**'s perspective, the traffic that ACL 100 applies to is inbound from the network connected to Gigabit Ethernet 0/0 interface. Enter interface configuration mode and apply the ACL.

```
R1(config)# interface gigabitEthernet 0/0
R1(config-if)# ip access-group 100 in
```

Step 3: Verify the ACL implementation.

a. Ping from **PC1** to **Server**. If the pings are unsuccessful, verify the IP addresses before continuing.

b. FTP from **PC1** to **Server**. The username and password are both **cisco**.

```
PC> ftp 172.22.34.62
```

c. Exit the FTP service of the **Server**.

```
ftp> quit
```

d. Ping from **PC1** to **PC2**. The destination host should be unreachable, because the traffic was not explicitly permitted.

Part 2: Configure, Apply and Verify an Extended Named ACL

Step 1: Configure an ACL to permit HTTP access and ICMP.

a. Named ACLs start with the **ip** keyword. From global configuration mode of **R1**, enter the following command, followed by a question mark.

```
R1(config)# ip access-list ?
```

```
  extended      Extended   Access   List
standard  Standard Access List
```

b. You can configure named standard and extended ACLs. This access list filters both source and destination IP addresses; therefore, it must be extended. Enter **HTTP_ONLY** as the name. (For Packet Tracer scoring, the name is case-sensitive.)

```
R1(config)# ip access-list extended HTTP_ONLY
```

c. The prompt changes. You are now in extended named ACL configuration mode. All devices on the **PC2** LAN need TCP access. Enter the network address, followed by a question mark.

```
R1(config-ext-nacl)# permit tcp 172.22.34.96 ?
  A.B.C.D  Source wildcard bits
```

d. An alternative way to calculate a wildcard is to subtract the subnet mask from 255.255.255.255.

```
  255.255.255.255
- 255.255.255.240
-----------------
=   0.  0.  0. 15
R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 ?
```

e. Finish the statement by specifying the server address as you did in Part 1 and filtering **www** traffic.

```
R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www
```

f. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC2** to **Server**. Note: The prompt remains the same and a specific type of ICMP traffic does not need to be specified.

```
R1(config-ext-nacl)# permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62
```

g. All other traffic is denied, by default. Exit out of extended named ACL configuration mode.

Step 2: Apply the ACL on the correct interface to filter traffic.

From **R1**'s perspective, the traffic that access list **HTTP_ONLY** applies to is inbound from the network connected to Gigabit Ethernet 0/1 interface. Enter the interface configuration mode and apply the ACL.

```
R1(config)# interface gigabitEthernet 0/1
R1(config-if)# ip access-group HTTP_ONLY in
```

Step 3: Verify the ACL implementation.

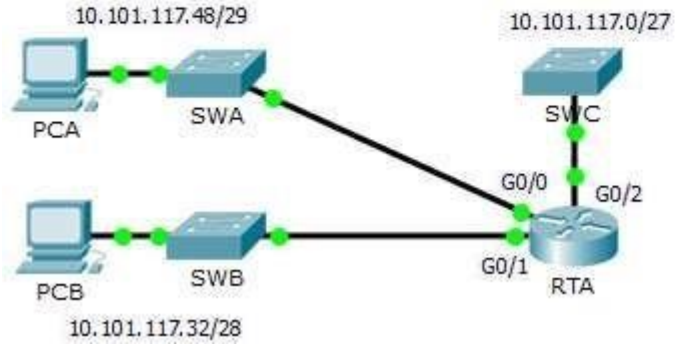
a. Ping from **PC2** to **Server**. The ping should be successful, if the ping is unsuccessful, verify the IP addresses before continuing.

b. FTP from **PC2** to **Server**. The connection should fail.

c. Open the web browser on **PC2** and enter the IP address of **Server** as the URL. The connection should be successful.

Practical 3: Configuring Extended ACLs - Scenario 2

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
RTA	G0/0	10.101.117.49	255.255.255.248	N/A
	G0/1	10.101.117.33	255.255.255.240	N/A
	G0/2	10.101.117.1	255.255.255.224	N/A
PCA	NIC	10.101.117.51	255.255.255.248	10.101.117.49
PCB	NIC	10.101.117.35	255.255.255.240	10.101.117.33
SWA	VLAN 1	10.101.117.50	255.255.255.248	10.101.117.49
SWB	VLAN 1	10.101.117.34	255.255.255.240	10.101.117.33
SWC	VLAN 1	10.101.117.2	255.255.255.224	10.101.117.1

Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Part 2: Reflection Questions

Background / Scenario

In this scenario, devices on one LAN are allowed to remotely access devices in another LAN using the SSH protocol. Besides ICMP, all traffic from other networks is denied.

The switches and router have also been pre-configured with the following:

- Enable secret password: **ciscoenpa55**
- Console password: **ciscoconpa55**
- Local username and password: **Admin** / **Adminpa55**

Packet Tracer - Configuring Extended ACLs - Scenario 2

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Configure, apply and verify an ACL to satisfy the following policy:

.

- SSH traffic from devices on the 10.101.117.32/28 network is allowed to devices on the 10.101.117.0/27 networks.
- ICMP traffic is allowed from any source to any destination.
- All other traffic to 10.101.117.0/27 is blocked.

Step 1: Configure the extended ACL.

a. From the appropriate configuration mode on **RTA**, use the last valid extended access list number to configure the ACL. Use the following steps to construct the first ACL statement:

1) The last extended list number is 199.
2) The protocol is TCP.
3) The source network is 10.101.117.32.
4) The wildcard can be determined by subtracting 255.255.255.240 from 255.255.255.255.
5) The destination network is 10.101.117.0.
6) The wildcard can be determined by subtracting 255.255.255.224 from 255.255.255.255.
7) The protocol is SSH (port 22).

What is the first ACL statement?

```
access-list 199 permit tcp 10.101.117.32 0.0.0.15 10.101.117.0 0.0.0.31 eq 22
```

b. ICMP is allowed, and a second ACL statement is needed. Use the same access list number to permit all ICMP traffic, regardless of the source or destination address. What is the second ACL statement? (Hint: Use the **any** keywords)

```
access-list 199 permit icmp any any
```

c. All other IP traffic is denied, by default.

Step 2: Apply the extended ACL.

The general rule is to place extended ACLs close to the source. However, because access list 199 affects traffic originating from both networks 10.101.117.48/29 and 10.101.117.32/28, the best placement for this ACL might be on interface Gigabit Ethernet 0/2 in the outbound direction. What is the command to apply ACL 199 to the Gigabit Ethernet 0/2 interface?

```
ip access-group 199 out
```

Step 3: Verify the extended ACL implementation.

a. Ping from **PCB** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addresses before continuing.

b. SSH from **PCB** to **SWC**. The username is **Admin**, and the password is **Adminpa55**.

```
PC> ssh -l Admin 10.101.117.2
```

c. Exit the SSH session to **SWC**.

d. Ping from **PCA** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addresses before continuing.

e. SSH from **PCA** to **SWC**. The access list causes the router to reject the connection.

Packet Tracer - Configuring Extended ACLs - Scenario 2

f. SSH from **PCA** to **SWB**. The access list is placed on **G0/2** and does not affect this connection. The username is **Admin**, and the password is **Adminpa55**.

g. After logging into **SWB**, do not log out. SSH to **SWC** in privileged EXEC mode.

```
SWB# ssh -l Admin 10.101.117.2
```

Part 2: Reflection Questions

1. How was PCA able to bypass access list 199 and SSH to SWC?

   Two steps were used: First, PCA used SSH to access SWB. From SWB, SSH was allowed to SWC.

2. What could have been done to prevent PCA from accessing SWC indirectly, while allowing PCB SSH access to SWC?

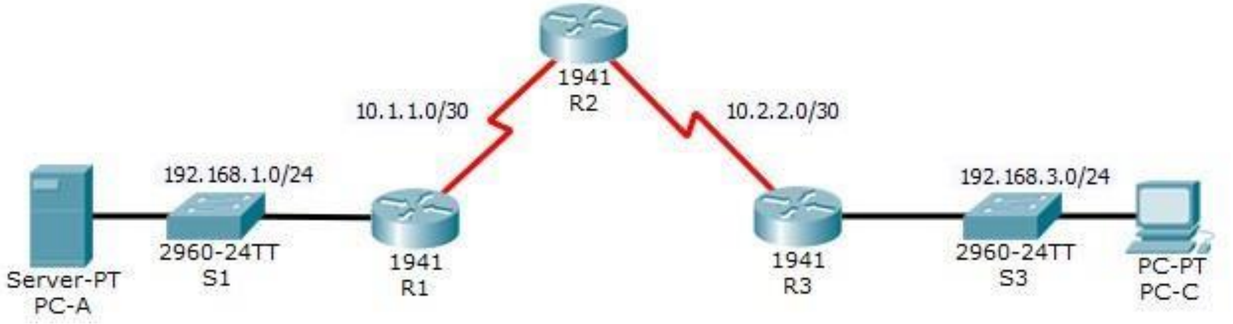
   Because it was requested to block all traffic to 10.101.117.0/27 except SSH traffic originating from 10.101.117.32/28 the access list could be written as is. Instead of applying the ACL to G0/2 outbound apply the same ACL to both G0/0 and G0/1 inbound.

Suggested Scoring Rubric

Activity Section	**Question Location**	**Possible Points**	**Earned Points**
Part 1: Configure, Apply and Verify an Extended Numbered ACL	Step 1a	4	
	Step 1b	4	
	Step 2	4	
	Part 1 Total	**12**	
Part 2: Reflection Questions	Question 1	4	
	Question 2	4	
	Part 2 Total	**8**	
	Packet Tracer Score	**80**	
	Total Score	**100**	

Practical 4: Configure IP ACLs to Mitigate Attacks.

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
	Lo0	192.168.2.1	255.255.255.0	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/6
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Objectives

- Verify connectivity among devices before firewall configuration.
- Use ACLs to ensure remote access to the routers is available only from management station PC-C.
- Configure ACLs on R1 and R3 to mitigate attacks.
- Verify ACL functionality.

Background/Scenario

Access to routers R1, R2, and R3 should only be permitted from PC-C, the management station. PC-C is also used for connectivity testing to PC-A, which is a server providing DNS, SMTP, FTP, and HTTPS services.

Standard operating procedure is to apply ACLs on edge routers to mitigate common threats based on source and destination IP address. In this activity, you will create ACLs on edge routers R1 and R3 to achieve this goal. You will then verify ACL functionality from internal and external hosts.

The routers have been pre-configured with the following:

○ Enable password: **ciscoenpa55** ○

Password for console: **ciscoconpa55**

○ SSH logon username and password:

SSHadmin/**ciscosshpa55** ○ IP addressing ○

Static routing

Part 1: Verify Basic Network Connectivity

Verify network connectivity prior to configuring the IP ACLs.

Step 1: From PC-A, verify connectivity to PC-C and R2.

a. From the command prompt, ping **PC-C** (192.168.3.3).

b. From the command prompt, establish an SSH session to **R2** Lo0 interface (192.168.2.1) using username **SSHadmin** and password **ciscosshpa55**. When finished, exit the SSH session. `SERVER> ssh -l SSHadmin 192.168.2.1`

Step 2: From PC-C, verify connectivity to PC-A and R2.

a. From the command prompt, ping **PC-A** (192.168.1.3).

b. From the command prompt, establish an SSH session to **R2** Lo0 interface (192.168.2.1) using username **SSHadmin** and password **ciscosshpa55**. Close the SSH session when finished. `PC> ssh -l SSHadmin 192.168.2.1`

c. Open a web browser to the **PC-A** server (192.168.1.3) to display the web page. Close the browser when done.

Part 2: Secure Access to Routers

Step 1: Configure ACL 10 to block all remote access to the routers except from PC-C.

Use the **access-list** command to create a numbered IP ACL on **R1**, **R2**, and **R3**.

```
R1(config)# access-list 10 permit host 192.168.3.3
R2(config)# access-list 10 permit host 192.168.3.3
R3(config)# access-list 10 permit host 192.168.3.3
```

Step 2: Apply ACL 10 to ingress traffic on the VTY lines.

Use the **access-class** command to apply the access list to incoming traffic on the VTY lines.

```
R1(config-line)# access-class 10 in
R2(config-line)# access-class 10 in
R3(config-line)# access-class 10 in
```

Step 3: Verify exclusive access from management station PC-C.

a. Establish an SSH session to 192.168.2.1 from **PC-C** (should be successful).

```
PC> ssh –l SSHadmin 192.168.2.1
```

b. Establish an SSH session to 192.168.2.1 from **PC-A** (should fail).

Part 3: Create a Numbered IP ACL 120 on R1

Create an IP ACL numbered 120 with the following rules:

○ Permit any outside host to access DNS, SMTP, and FTP services on server

PC-A. ○ Deny any outside host access to HTTPS services on **PC-A.** ○

Permit **PC-C** to access **R1** via SSH.

Note: Check Results will not show a correct configuration for ACL 120 until you modify it in Part 4.

Step 1: Verify that PC-C can access the PC-A via HTTPS using the web browser.

Be sure to disable HTTP and enable HTTPS on server **PC-A**.

Step 2: Configure ACL 120 to specifically permit and deny the specified traffic. Use

the **access-list** command to create a numbered IP ACL.

```
R1(config)# access-list 120 permit udp any host 192.168.1.3 eq domain
R1(config)# access-list 120 permit tcp any host 192.168.1.3 eq smtp
R1(config)# access-list 120 permit tcp any host 192.168.1.3 eq ftp
R1(config)# access-list 120 deny tcp any host 192.168.1.3 eq 443
R1(config)# access-list 120 permit tcp host 192.168.3.3 host 10.1.1.1 eq 22
```

Step 3: Apply the ACL to interface S0/0/0. Use the **ip access-group** command to apply the

access list to incoming traffic on interface S0/0/0.

```
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 120 in
```

Step 4: Verify that PC-C cannot access PC-A via HTTPS using the web browser. Part

4: Modify an Existing ACL on R1

Permit ICMP echo replies and destination unreachable messages from the outside network (relative to **R1**). Deny all other incoming ICMP packets.

Step 1: Verify that PC-A cannot successfully ping the loopback interface on R2.

Step 2: Make any necessary changes to ACL 120 to permit and deny the specified traffic. Use

the **access-list** command to create a numbered IP ACL.

```
R1(config)# access-list 120 permit icmp any any echo-reply
R1(config)# access-list 120 permit icmp any any unreachable
R1(config)# access-list 120 deny icmp any any
R1(config)# access-list 120 permit ip any any
```

Step 3: Verify that PC-A can successfully ping the loopback interface on R2. Part

5: Create a Numbered IP ACL 110 on R3

Deny all outbound packets with source address outside the range of internal IP addresses on **R3**.

Step 1: Configure ACL 110 to permit only traffic from the inside network. Use

the **access-list** command to create a numbered IP ACL.

```
R3(config)# access-list 110 permit ip 192.168.3.0 0.0.0.255 any
```

Step 2: Apply the ACL to interface G0/1. Use the **ip access-group** command to apply the

access list to incoming traffic on interface G0/1.

```
R3(config)# interface g0/1
R3(config-if)# ip access-group 110 in
```

Part 6: Create a Numbered IP ACL 100 on R3

On **R3**, block all packets containing the source IP address from the following pool of addresses: any RFC 1918 private addresses, 127.0.0.0/8, and any IP multicast address. Since **PC-C** is being used for remote administration, permit SSH traffic from the 10.0.0.0/8 network to return to the host **PC-C**.

Step 1: Configure ACL 100 to block all specified traffic from the outside network.

You should also block traffic sourced from your own internal address space if it is not an RFC 1918 address. In this activity, your internal address space is part of the private address space specified in RFC 1918. Use the **access-list** command to create a numbered IP ACL. **`access-list 100 permit tcp 10.0.0.0`**

```
R3(config)#

0.255.255.255 eq 22 host
192.168.3.3
R3(config)# access-list 100 deny ip 10.0.0.0 0.255.255.255 any
R3(config)# access-list 100 deny ip 172.16.0.0 0.15.255.255 any
R3(config)# access-list 100 deny ip 192.168.0.0 0.0.255.255 any
R3(config)# access-list 100 deny ip 127.0.0.0 0.255.255.255 any
R3(config)# access-list 100 deny ip 224.0.0.0 15.255.255.255 any R3(config)#
access-list 100 permit ip any any
```

Step 2: Apply the ACL to interface Serial 0/0/1. Use the **ip access-group** command to apply the

access list to incoming traffic on interface Serial 0/0/1.

```
R3(config)# interface s0/0/1
R3(config-if)# ip access-group 100 in
```

Step 3: Confirm that the specified traffic entering interface Serial 0/0/1 is handled correctly.

a. From the PC-C command prompt, ping the PC-A server. The ICMP echo replies are blocked by the ACL since they are sourced from the 192.168.0.0/16 address space.

b. Establish an SSH session to 192.168.2.1 from **PC-C** (should be successful).

Step 4: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.

!!!Script for R1

```
access-list 10 permit host 192.168.3.3
line vty 0 4
 access-class 10 in
access-list 120 permit udp any host 192.168.1.3 eq domain
access-list 120 permit tcp any host 192.168.1.3 eq smtp
access-list 120 permit tcp any host 192.168.1.3 eq ftp access-
list 120 deny tcp any host 192.168.1.3 eq 443 access-list 120
permit tcp host 192.168.3.3 host 10.1.1.1 eq 22 interface
s0/0/0  ip access-group 120 in
access-list 120 permit icmp any any echo-reply
access-list 120 permit icmp any any unreachable
access-list 120 deny icmp any any access-list
120 permit ip any any
```

!!!Script for R2

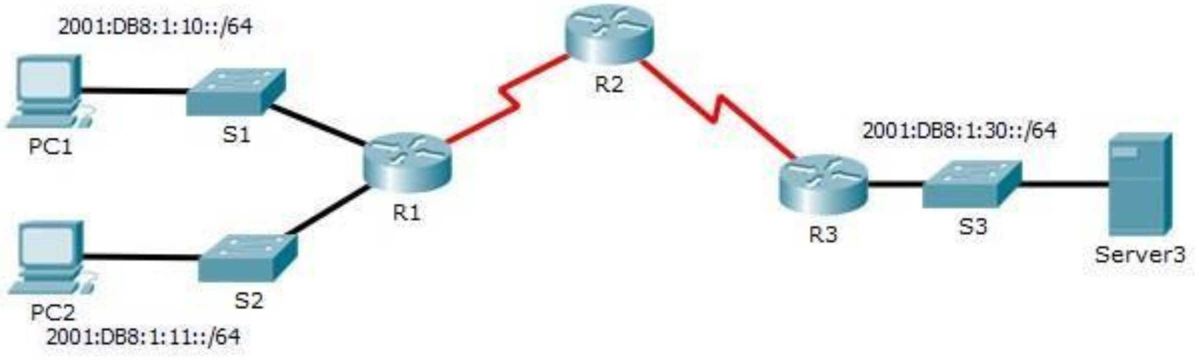
```
access-list 10 permit host 192.168.3.3

line vty 0 4

 access-class 10 in
```

!!!Script for R3

```
access-list 10 permit host 192.168.3.3
line vty 0 4
 access-class 10 in
access-list 100 permit tcp 10.0.0.0 0.255.255.255 eq 22 host 192.168.3.3
access-list 100 deny ip 10.0.0.0 0.255.255.255 any access-list
100 deny ip 172.16.0.0 0.15.255.255 any access-list 100 deny
ip 192.168.0.0 0.0.255.255 any access-list 100 deny ip
127.0.0.0 0.255.255.255 any access-list 100 deny ip 224.0.0.0
15.255.255.255 any
access-list 100 permit ip any any
interface s0/0/1  ip access-group
100 in
access-list 110 permit ip 192.168.3.0 0.0.0.255 any
interface g0/1  ip access-group 110 in
```

Practical 5: Configuring IPv6 ACLs

Topology



Addressing Table

Device	Interface	IPv6 Address/Prefix	Default Gateway
Server3	NIC	2001:DB8:1:30::30/64	FE80::30

Objectives

Part 1: Configure, Apply, and Verify an IPv6 ACL

Part 2: Configure, Apply, and Verify a Second IPv6 ACL

Part 1: Configure, Apply, and Verify an IPv6 ACL

Logs indicate that a computer on the 2001:DB8:1:11::0/64 network is repeatedly refreshing a web page. This is causing a Denial-of-Service (DoS) attack against **Server3**. Until the client can be identified and cleaned, you must block HTTP and HTTPS access to that network with an access list.

Step 1: Configure an ACL that will block HTTP and HTTPS access.

Configure an ACL named **BLOCK_HTTP** on **R1** with the following statements. a.

Block HTTP and HTTPS traffic from reaching **Server3**.

```
R1(config)# deny tcp any host 2001:DB8:1:30::30 eq www
R1(config)# deny tcp any host 2001:DB8:1:30::30 eq 443
```

b. Allow all other IPv6 traffic to pass.

```
R1(config)# permit ipv6 any any
```

Step 2: Apply the ACL to the correct interface. Apply the ACL on the

interface closest to the source of the traffic to be blocked.

```
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ipv6 traffic-filter BLOCK_HTTP in
```

Step 3: Verify the ACL implementation.

Verify that the ACL is operating as intended by conducting the following tests:

- Open the **web browser** of **PC1** to http://2001:DB8:1:30::30 or https://2001:DB8:1:30::30. The website should appear.
- Open the **web browser** of **PC2** to http://2001:DB8:1:30::30 or https://2001:DB8:1:30::30. The website should be blocked.
- Ping from **PC2** to 2001:DB8:1:30::30. The ping should be successful.

Part 2: Configure, Apply, and Verify a Second IPv6 ACL

The logs now indicate that your server is receiving pings from many different IPv6 addresses in a Distributed Denial of Service (DDoS) attack. You must filter ICMP ping requests to your server.

Step 1: Create an access list to block ICMP.

Configure an ACL named **BLOCK_ICMP** on **R3** with the following statements: a.

Block all ICMP traffic from any hosts to any destination.

```
R3(config)# deny icmp any any
```

b. Allow all other IPv6 traffic to pass.

```
R3(config)# permit ipv6 any any
```

Step 2: Apply the ACL to the correct interface.

In this case, ICMP traffic can come from any source. To ensure that ICMP traffic is blocked, regardless of its source or any changes that occur to the network topology, apply the ACL closest to the destination.

```
R3(config)# interface GigabitEthernet0/0

R3(config-if)# ipv6 traffic-filter BLOCK_ICMP out
```

Step 3: Verify that the proper access list functions.

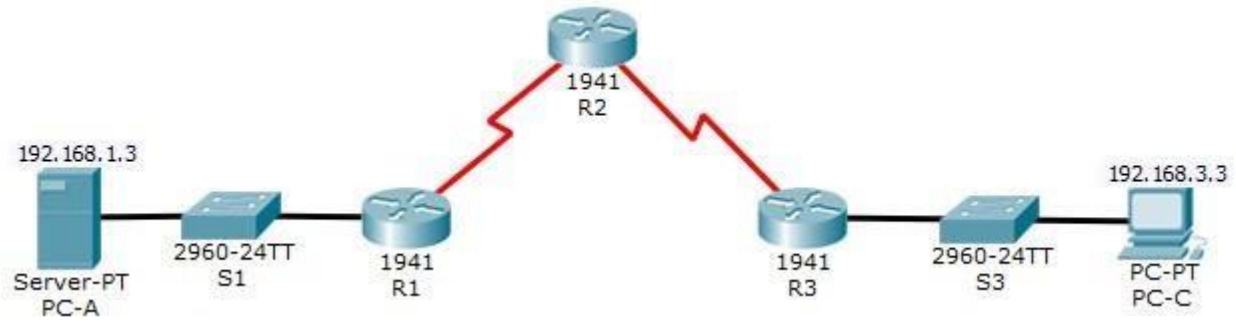
a. Ping from **PC2** to 2001:DB8:1:30::30. The ping should fail.

b. Ping from **PC1** to 2001:DB8:1:30::30. The ping should fail.

Open the **web browser** of **PC1** to http://2001:DB8:1:30::30 or https://2001:DB8:1:30::30. The website should display.

Practical 6: Configuring a Zone-Based Policy Firewall (ZPF)

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/6
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Objectives

- Verify connectivity among devices before firewall configuration.
- Configure a zone-based policy (ZPF) firewall on R3.
- Verify ZPF firewall functionality using ping, SSH, and a web browser.

Background/Scenario

ZPFs are the latest development in the evolution of Cisco firewall technologies. In this activity, you will configure a basic ZPF on an edge router R3 that allows internal hosts access to external resources and blocks external hosts from accessing internal resources. You will then verify firewall functionality from internal and external hosts.

The routers have been pre-configured with the following:

○ Console password: **ciscoconpa55**

○ Password for vty lines: **ciscovtypa55** ○ Enable password: **ciscoenpa55**

○ Host names and IP addressing ○ Local username and password:

Admin / **Adminpa55** ○ Static routing

Part 1: Verify Basic Network Connectivity

Verify network connectivity prior to configuring the zone-based policy firewall.

Step 1: From the PC-A command prompt, ping PC-C at 192.168.3.3.

Step 2: Access R2 using SSH.

a. From the **PC-C** command prompt, SSH to the S0/0/1 interface on **R2** at **10.2.2.2**. Use the username **Admin** and password **Adminpa55** to log in. `PC>` **`ssh -l Admin 10.2.2.2`**

b. Exit the SSH session.

Step 3: From PC-C, open a web browser to the PC-A server.

a. Click the **Desktop** tab and then click the **Web Browser** application. Enter the **PC-A** IP address **192.168.1.3** as the URL. The Packet Tracer welcome page from the web server should be displayed. b.

Close the browser on **PC-C**.

Part 2: Create the Firewall Zones on R3

Note: For all configuration tasks, be sure to use the exact names as specified.

Step 1: Enable the Security Technology package.

a. On **R3**, issue the **show version** command to view the Technology Package license information.

b. If the Security Technology package has not been enabled, use the following command to enable the package.

```
R3(config)# license boot module c1900 technology-package securityk9
```

c. Accept the end-user license agreement.

d. Save the running-config and reload the router to enable the security license.

e. Verify that the Security Technology package has been enabled by using the **show version** command.

Step 2: Create an internal zone. Use the **zone security** command

to create a zone named **IN-ZONE**. `R3(config)#` **`zone security`**

`IN-ZONE`

```
R3(config-sec-zone) exit
```

Step 3: Create an external zone. Use the **zone security** command to

create a zone named **OUT-ZONE**.

```
R3(config-sec-zone)# zone security OUT-ZONE R3(config-sec-
zone)# exit
```

Part 3: Identify Traffic Using a Class-Map

Step 1: Create an ACL that defines internal traffic.

Use the **access-list** command to create extended ACL **101** to permit all IP protocols from the **192.168.3.0/24** source network to any destination.

```
R3(config)# access-list 101 permit ip 192.168.3.0 0.0.0.255 any
```

.

Step 2: Create a class map referencing the internal traffic ACL.

Use the **class-map type inspect** command with the **match-all** option to create a class map named **IN-NETCLASS-MAP**. Use the **match access-group** command to match ACL **101**.

```
R3(config)# class-map type inspect match-all IN-NET-CLASS-MAP
R3(config-cmap)# match access-group 101
R3(config-cmap)# exit
```

Part 4: Specify Firewall Policies

Step 1: Create a policy map to determine what to do with matched traffic.

Use the **policy-map type inspect** command and create a policy map named **IN-2-OUT-PMAP**.

```
R3(config)# policy-map type inspect IN-2-OUT-PMAP
```

Step 2: Specify a class type of inspect and reference class map IN-NET-CLASS-MAP.

```
R3(config-pmap)# class type inspect IN-NET-CLASS-MAP
```

Step 3: Specify the action of inspect for this policy map.

The use of the **inspect** command invokes context-based access control (other options include pass and drop).

```
R3(config-pmap-c)# inspect

%No specific protocol configured in class IN-NET-CLASS-MAP for inspection. All
protocols will be inspected.
```

Issue the **exit** command twice to leave **config-pmap-c** mode and return to **config** mode.

```
R3(config-pmap-c)# exit
R3(config-pmap)# exit
```

Part 5: Apply Firewall Policies

Step 1: Create a pair of zones.

Using the **zone-pair security** command, create a zone pair named **IN-2-OUT-ZPAIR**. Specify the source and destination zones that were created in Task 1.

```
R3(config)# zone-pair security IN-2-OUT-ZPAIR source IN-ZONE destination
OUTZONE
```

Step 2: Specify the policy map for handling the traffic between the two zones.

Attach a policy-map and its associated actions to the zone pair using the **service-policy type inspect** command and reference the policy map previously created, **IN-2-OUT-PMAP**.

```
R3(config-sec-zone-pair)# service-policy type inspect IN-2-OUT-PMAP
R3(config-sec-zone-pair)# exit
R3(config)#
```

Step 3: Assign interfaces to the appropriate security zones.

Use the **zone-member security** command in interface configuration mode to assign G0/1 to **IN-ZONE** and S0/0/1 to **OUT-ZONE**.

```
R3(config)# interface g0/1
R3(config-if)# zone-member security IN-ZONE
R3(config-if)# exit
R3(config)# interface s0/0/1
R3(config-if)# zone-member security OUT-ZONE R3(config-if)#
exit
```

Step 4: Copy the running configuration to the startup configuration.

Part 6: Test Firewall Functionality from IN-ZONE to OUT-ZONE

Verify that internal hosts can still access external resources after configuring the ZPF.

Step 1: From internal PC-C, ping the external PC-A server.

From the **PC-C** command prompt, ping **PC-A** at 192.168.1.3. The ping should succeed.

Step 2: From internal PC-C, SSH to the R2 S0/0/1 interface.

a. From the **PC-C** command prompt, SSH to **R2** at 10.2.2.2. Use the username **Admin** and the password **Adminpa55** to access R2. The SSH session should succeed.

b. While the SSH session is active, issue the command **show policy-map type inspect zone-pair sessions** on **R3** to view established sessions.

```
R3# show policy-map type inspect zone-pair sessions

policy exists on zp IN-2-OUT-ZPAIR
Zone-pair: IN-2-OUT-ZPAIR

Service-policy inspect : IN-2-OUT-PMAP

Class-map: IN-NET-CLASS-MAP (match-all)
Match: access-group 101
Inspect

Number of Established Sessions = 1
Established Sessions
Session 175216232 (192.168.3.3:1028)=>(10.2.2.2:22) tcp SIS_OPEN/TCP_ESTAB
Created 00:00:25, Last heard 00:00:20
Bytes sent (initiator:responder) [1195:1256]
Class-map:  class-default  (match-any)
Match: any
Drop (default action)
0 packets, 0 bytes
```

What is the source IP address and port number?
192.168.3.3:1028 (port 1028 is random)

What is the destination IP address and port number?

10.2.2.2:22 (SSH = port 22)

Step 3: From PC-C, exit the SSH session on R2 and close the command prompt window.

Step 4: From internal PC-C, open a web browser to the PC-A server web page.

Enter the server IP address **192.168.1.3** in the browser URL field, and click **Go**. The HTTP session should succeed. While the HTTP session is active, issue the command **show policy-map type inspect zone-pair sessions** on **R3** to view established sessions.

Note: If the HTTP session times out before you execute the command on **R3**, you will have to click the **Go** button on **PC-C** to generate a session between **PC-C** and **PC-A**.

```
R3# show policy-map type inspect zone-pair sessions

policy exists on zp IN-2-OUT-ZPAIR
Zone-pair: IN-2-OUT-ZPAIR

Service-policy inspect : IN-2-OUT-PMAP

Class-map: IN-NET-CLASS-MAP (match-all)
Match: access-group 101
Inspect

Number of Established Sessions = 1
Established Sessions
Session 565266624 (192.168.3.3:1031)=>(192.168.1.3:80) tcp SIS_OPEN/TCP_ESTAB
Created 00:00:01, Last heard 00:00:01
Bytes sent (initiator:responder) [284:552]
Class-map: class-default (match-any)
Match: any
Drop (default action)
0 packets, 0 bytes
```

What is the source IP address and port number?

192.168.3.3:1031 (port 1031 is random)

What is the destination IP address and port number?

192.168.1.3:80 (HTTP web = port 80)

Step 5: Close the browser on PC-C.

Part 7: Test Firewall Functionality from OUT-ZONE to IN-ZONE

Verify that external hosts CANNOT access internal resources after configuring the ZPF.

Step 1: From the PC-A server command prompt, ping PC-C.

From the **PC-A** command prompt, ping **PC-C** at 192.168.3.3. The ping should fail.

Step 2: From R2, ping PC-C.

From **R2**, ping **PC-C** at 192.168.3.3. The ping should fail.

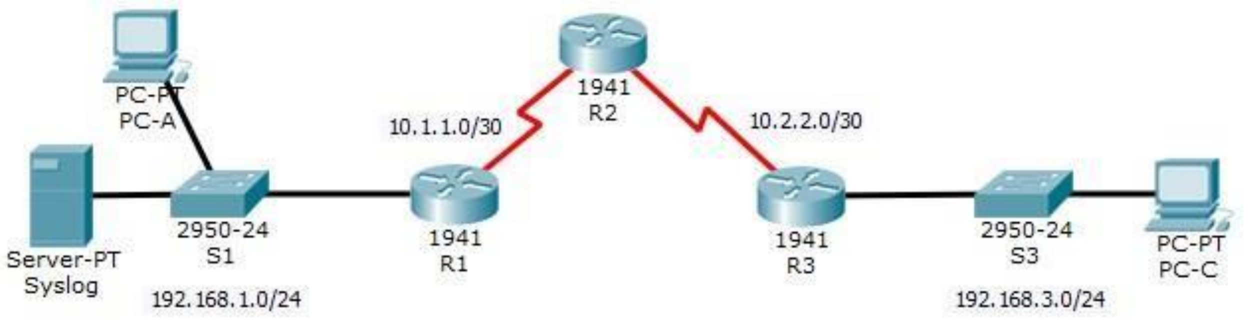
Step 3: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.

.

Practical 7: Configure IOS Intrusion Prevention System (IPS) Using the CLI

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/1
	S0/0/0	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0 (DCE)	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/1
	S0/0/0	10.2.2.1	255.255.255.252	N/A	N/A
Syslog	NIC	192.168.1.50	255.255.255.0	192.168.1.1	S1 F0/2
PC-A	NIC	192.168.1.2	255.255.255.0	192.168.1.1	S1 F0/3
PC-C	NIC	192.168.3.2	255.255.255.0	192.168.3.1	S3 F0/2

Objectives

- Enable IOS IPS.
- Configure logging.
- Modify an IPS signature.
- Verify IPS.

Background / Scenario

Your task is to enable IPS on R1 to scan traffic entering the 192.168.1.0 network.

The server labeled Syslog is used to log IPS messages. You must configure the router to identify the syslog server to receive logging messages. Displaying the correct time and date in syslog messages is vital when using syslog to monitor the network. Set the clock and configure the timestamp service for logging on the routers. Finally, enable IPS to produce an alert and drop ICMP echo reply packets inline.

The server and PCs have been preconfigured. The routers have also been preconfigured with the following:

○ Enable password: **ciscoenpa55** ○ Console
password: **ciscoconpa55** ○ SSH username and password:
SSHadmin / **ciscosshpa55** ○ OSPF 101

Part 1: Enable IOS IPS

Note: Within Packet Tracer, the routers already have the signature files imported and in place. They are the default xml files in flash. For this reason, it is not necessary to configure the public crypto key and complete a manual import of the signature files.

Step 1: Enable the Security Technology package.

a. On **R1**, issue the **show version** command to view the Technology Package license information.

b. If the Security Technology package has not been enabled, use the following command to enable the package.

```
R1(config)# license boot module c1900 technology-package securityk9
```

c. Accept the end user license agreement.

d. Save the running-config and reload the router to enable the security license.

e. Verify that the Security Technology package has been enabled by using the **show version** command.

Step 2: Verify network connectivity.

a. Ping from **PC-C** to **PC-A**. The ping should be successful.

b. Ping from **PC-A** to **PC-C**. The ping should be successful.

Step 3: Create an IOS IPS configuration directory in flash.

On **R1**, create a directory in flash using the **mkdir** command. Name the directory **ipsdir**.

```
R1# mkdir ipsdir
Create directory filename [ipsdir]? <Enter> Created
dir flash:ipsdir
```

Step 4: Configure the IPS signature storage location.

On **R1**, configure the IPS signature storage location to be the directory you just created.

```
R1(config)# ip ips config location flash:ipsdir
```

Step 5: Create an IPS rule.

On **R1**, create an IPS rule name using the **ip ips name** *name* command in global configuration mode. Name the IPS rule **iosips**.

```
R1(config)# ip ips name iosips
```

Step 6: Enable logging.

IOS IPS supports the use of syslog to send event notification. Syslog notification is enabled by default. If logging console is enabled, IPS syslog messages display. a. Enable syslog if it is not enabled.

```
R1(config)# ip ips notify log
```

b. If necessary, use the **clock set** command from privileged EXEC mode to reset the clock.

```
R1# clock set 10:20:00 10 january 2014
```

c. Verify that the timestamp service for logging is enabled on the router using the **show run** command.
Enable the timestamp service if it is not enabled.

```
R1(config)# service timestamps log datetime msec
```

d. Send log messages to the syslog server at IP address 192.168.1.50.

```
R1(config)# logging host 192.168.1.50
```

Step 7: Configure IOS IPS to use the signature categories.

Retire the **all** signature category with the **retired true** command (all signatures within the signature release). Unretire the **IOS_IPS Basic** category with the **retired false** command.

```
R1(config)# ip ips signature-category
R1(config-ips-category)# category all
R1(config-ips-category-action)# retired true
R1(config-ips-category-action)# exit
R1(config-ips-category)# category ios_ips basic
R1(config-ips-category-action)# retired false
R1(config-ips-category-action)# exit
R1(config-ips-cateogry)# exit
Do you want to accept these changes? [confirm] <Enter>
```

Step 8: Apply the IPS rule to an interface.

Apply the IPS rule to an interface with the **ip ips name** *direction* command in interface configuration mode. Apply the rule outbound on the G0/1 interface of **R1**. After you enable IPS, some log messages will be sent to the console line indicating that the IPS engines are being initialized.

Note: The direction **in** means that IPS inspects only traffic going into the interface. Similarly, **out** means that IPS inspects only traffic going out of the interface.

```
R1(config)# interface g0/1
R1(config-if)# ip ips iosips out
```

Part 2: Modify the Signature

Step 1: Change the event-action of a signature.

Un-retire the echo request signature (signature 2004, subsig ID 0), enable it, and change the signature action to alert and drop.

```
R1(config)# ip ips signature-definition
R1(config-sigdef)# signature 2004 0
R1(config-sigdef-sig)# status
R1(config-sigdef-sig-status)# retired false
R1(config-sigdef-sig-status)# enabled true
R1(config-sigdef-sig-status)# exit
```

```
R1(config-sigdef-sig)# engine
R1(config-sigdef-sig-engine)# event-action produce-alert R1(config-sigdef-sig-engine)# event-action deny-packet-inline
R1(config-sigdef-sig-engine)# exit
R1(config-sigdef-sig)# exit
R1(config-sigdef)# exit
Do you want to accept these changes? [confirm] <Enter>
```

Step 2: Use show commands to verify IPS.

Use the **show ip ips all** command to view the IPS configuration status summary.

To which interfaces and in which direction is the **iosips** rule applied?

____

G0/1 outbound.

Step 3: Verify that IPS is working properly.

a. From **PC-C**, attempt to ping **PC-A**. Were the pings successful? Explain.

   The pings should fail. This is because the IPS rule for event-action of an echo request was set to “deny-packet-inline”.

b. From **PC-A**, attempt to ping **PC-C**. Were the pings successful? Explain.

   The ping should be successful. This is because the IPS rule does not cover echo reply. When PC-A pings PC-C, PC-C responds with an echo reply.

Step 4: View the syslog messages.

a. Click the **Syslog** server.

b. Select the **Services** tab.

c. In the left navigation menu, select **SYSLOG** to view the log file.

Step 5: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.

!!!Script for R1

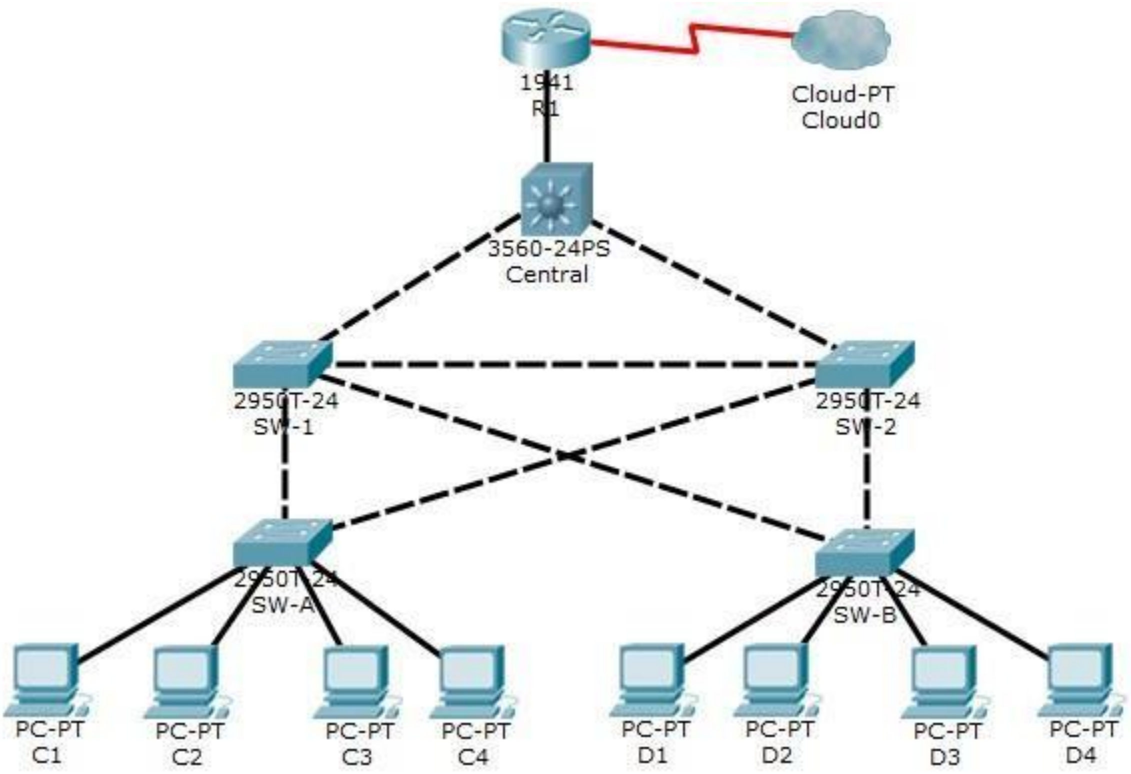
```
clock set 10:20:00 10 january 2014 mkdir
ipsdir
 config
t
```

Configure IOS Intrusion Prevention System (IPS) using CLI

```
license boot module c1900 technology-package securityk9
yes end reload  config t
ip ips config location flash:ipsdir
ip ips name iosips ip ips notify log
service timestamps log datetime msec
logging host 192.168.1.50
 ip ips signature-category
category all  retired true  exit
category ios_ips basic  retired
false  exit  exit  interface
g0/1  ip ips iosips out  exit ip
ips signature-definition
signature 2004 0  status
retired false  enabled true
exit  engine  event-action
produce-alert  event-action
deny-packet-inline  exit  exit
exit
```

Practical 8: Packet Tracer - Layer 2 Security

Topology



Objectives

- Assign the Central switch as the root bridge.
- Secure spanning-tree parameters to prevent STP manipulation attacks.
- Enable port security to prevent CAM table overflow attacks.

Background / Scenario

There have been a number of attacks on the network recently. For this reason, the network administrator has assigned you the task of configuring Layer 2 security.

For optimum performance and security, the administrator would like to ensure that the root bridge is the 3560 Central switch. To prevent spanning-tree manipulation attacks, the administrator wants to ensure that the STP parameters are secure. To prevent against CAM table overflow attacks, the network administrator has decided to configure port security to limit the number of MAC addresses each switch port can learn. If the number of MAC addresses exceeds the set limit, the administrator would like the port to be shutdown. All switch devices have been preconfigured with the following:

- Enable password: **ciscoenpa55** o Console password:

  ciscoconpa55

- SSH username and password: **SSHadmin** / **ciscosshpa55**

Part 1: Configure Root Bridge

Step 1: Determine the current root bridge.

.

From **Central**, issue the **show spanning-tree** command to determine the current root bridge, to see the ports in use, and to see their status.

Which switch is the current root bridge?

Current root is SW-1.

Based on the current root bridge, what is the resulting spanning tree? (Draw the spanning-tree topology.)

Step 2: Assign Central as the primary root bridge.

Using the **spanning-tree vlan 1 root primary** command, and assign **Central** as the root bridge.

```
Central(config)# spanning-tree vlan 1 root primary
```

Step 3: Assign SW-1 as a secondary root bridge.

Assign **SW-1** as the secondary root bridge using the **spanning-tree vlan 1 root secondary** command.

```
SW-1(config)# spanning-tree vlan 1 root secondary
```

Step 4: Verify the spanning-tree configuration.

Issue the **show spanning-tree** command to verify that **Central** is the root bridge.

```
Central# show spanning-tree
VLAN0001
   Spanning tree enabled protocol ieee
   Root ID  Priority       24577
            Address        00D0.D31C.634C
            This bridge is the root
            Hello Time  2 sec  Max Age  20 sec   Forward Delay  15 sec
```

Which switch is the current root bridge?

Current root is Central

Based on the new root-bridge, what is the resulting spanning tree? (Draw the spanning-tree topology.)

Part 2: Protect Against STP Attacks

Secure the STP parameters to prevent STP manipulation attacks.

Step 1: Enable PortFast on all access ports.

PortFast is configured on access ports that connect to a single workstation or server to enable them to become active more quickly. On the connected access ports of the **SW-A** and **SW-B**, use the **spanning-tree portfast** command.

```
SW-A(config)# interface range f0/1 - 4
SW-A(config-if-range)# spanning-tree portfast

SW-B(config)# interface range f0/1 - 4
SW-B(config-if-range)# spanning-tree portfast
```

Step 2: Enable BPDU guard on all access ports.

BPDU guard is a feature that can help prevent rogue switches and spoofing on access ports. Enable BPDU guard on **SW-A** and **SW-B** access ports.

```
SW-A(config)# interface range f0/1 - 4
SW-A(config-if-range)# spanning-tree bpduguard enable

SW-B(config)# interface range f0/1 - 4
SW-B(config-if-range)# spanning-tree bpduguard enable
```

Note: Spanning-tree BPDU guard can be enabled on each individual port using the **spanning-tree bpduguard enable** command in interface configuration mode or the **spanning-tree portfast bpduguard default** command in global configuration mode. For grading purposes in this activity, please use the **spanning-tree bpduguard enable** command.

Step 3: Enable root guard.

Root guard can be enabled on all ports on a switch that are not root ports. It is best deployed on ports that connect to other non-root switches. Use the **show spanning-tree** command to determine the location of the root port on each switch.

On **SW-1**, enable root guard on ports F0/23 and F0/24. On **SW-2**, enable root guard on ports F0/23 and F0/24.

```
SW-1(config)# interface range f0/23 - 24
SW-1(config-if-range)# spanning-tree guard root

SW-2(config)# interface range f0/23 - 24
SW-2(config-if-range)# spanning-tree guard root
```

Part 3: Configure Port Security and Disable Unused Ports

Step 1: Configure basic port security on all ports connected to host devices.

This procedure should be performed on all access ports on **SW-A** and **SW-B**. Set the maximum number of learned MAC addresses to **2**, allow the MAC address to be learned dynamically, and set the violation to **shutdown**. **Note**: A switch port must be configured as an access port to enable port security.

```
SW-A(config)# interface range f0/1 - 22
SW-A(config-if-range)# switchport mode access
SW-A(config-if-range)# switchport port-security
SW-A(config-if-range)# switchport port-security maximum 2
SW-A(config-if-range)# switchport port-security violation shutdown SW-
A(config-if-range)# switchport port-security mac-address sticky

SW-B(config)# interface range f0/1 - 22
SW-B(config-if-range)# switchport mode access
SW-B(config-if-range)# switchport port-security
SW-B(config-if-range)# switchport port-security maximum 2
SW-B(config-if-range)# switchport port-security violation shutdown
SW-B(config-if-range)# switchport port-security mac-address sticky
```

Why is port security not enabled on ports that are connected to other switch devices?

Ports connected to other switch devices have a multitude of MAC addresses learned for that single port. Limiting the number of MAC addresses that can be learned on these ports can significantly impact network functionality.

Step 2: Verify port security.

a. On **SW-A**, issue the command **show port-security interface f0/1** to verify that port security has been configured.

```
SW-A# show port-security interface f0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses     : 2
Total  MAC  Addresses    :  0
Configured MAC Addresses  : 0
Sticky MAC Addresses       : 0
Last  Source  Address:Vlan  :  0000.0000.0000:0
Security Violation Count  : 0
```

b. Ping from **C1** to **C2** and issue the command **show port-security interface f0/1** again to verify that the switch has learned the MAC address for **C1**.

Step 3: Disable unused ports.

Disable all ports that are currently unused.

```
SW-A(config)# interface range f0/5 - 22
SW-A(config-if-range)# shutdown


SW-B(config)# interface range f0/5 - 22
SW-B(config-if-range)# shutdown
```

Step 4: Check results.

Your completion percentage should be 100%. Click **Check Results** to view feedback and verification of which of the required components have been completed.

!!!Script for Central

```
conf t

spanning-tree vlan 1 root primary end
```

```
!!!Script for SW-1 conf

t

spanning-tree vlan 1 root secondary
interface range f0/23 - 24
spanning-tree guard root end
```

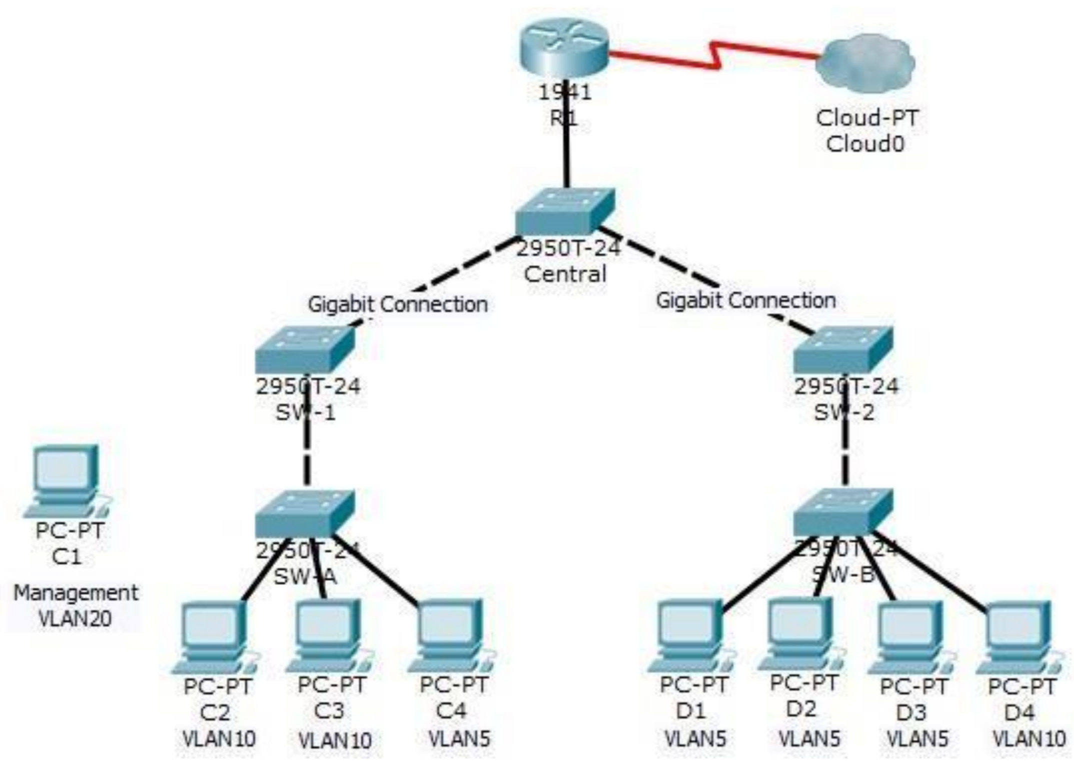
```
!!!Script for SW-2 conf t
interface range f0/23 - 24
spanning-tree guard root
end
```

```
!!!Script for SW-A conf t interface range f0/1
- 4 spanning-tree portfast spanning-tree
bpduguard enable interface range f0/1 - 22
switchport mode access switchport port-
security switchport port-security maximum 2
switchport port-security violation shutdown
switchport port-security mac-address sticky
interface range f0/5 - 22 shutdown end
```

```
!!!Script for SW-B conf t interface range f0/1
- 4

 spanning-tree portfast spanning-tree
bpduguard enable interface range f0/1 - 22
switchport mode access switchport port-
security switchport port-security maximum 2
switchport port-security violation shutdown
switchport port-security mac-address sticky
interface range f0/5 - 22 shutdown end
```

Practical 9: Layer 2 VLAN Security

Topology



Objectives

- Connect a new redundant link between SW-1 and SW-2.
- Enable trunking and configure security on the new trunk link between SW-1 and SW-2.
- Create a new management VLAN (VLAN 20) and attach a management PC to that VLAN.
- Implement an ACL to prevent outside users from accessing the management VLAN.

Background / Scenario

A company's network is currently set up using two separate VLANs: VLAN 5 and VLAN 10. In addition, all trunk ports are configured with native VLAN 15. A network administrator wants to add a redundant link between switch SW-1 and SW-2. The link must have trunking enabled and all security requirements should be in place.

In addition, the network administrator wants to connect a management PC to switch SW-A. The administrator would like to enable the management PC to connect to all switches and the router, but does not want any other devices to connect to the management PC or the switches. The administrator would like to create a new VLAN 20 for management purposes.

All devices have been preconfigured with: ○

Enable secret password: **ciscoenpa55** ○

Console password: **ciscoconpa55**

o SSH username and password: **SSHadmin** / **ciscosshpa55** **Part**

1: Verify Connectivity

Step 1: Verify connectivity between C2 (VLAN 10) and C3 (VLAN 10).

Step 2: Verify connectivity between C2 (VLAN 10) and D1 (VLAN 5).

Note: If using the simple PDU GUI packet, be sure to ping twice to allow for ARP.

Part 2: Create a Redundant Link Between SW-1 and SW-2

Step 1: Connect SW-1 and SW-2.

Using a crossover cable, connect port F0/23 on **SW-1** to port F0/23 on **SW-2**.

Step 2: Enable trunking, including all trunk security mechanisms on the link between SW-1 and SW-2.

Trunking has already been configured on all pre-existing trunk interfaces. The new link must be configured for trunking, including all trunk security mechanisms. On both **SW-1** and **SW-2**, set the port to trunk, assign native VLAN 15 to the trunk port, and disable auto-negotiation.

```
SW-1(config)# interface f0/23
SW-1(config-if)# switchport mode trunk
SW-1(config-if)# switchport trunk native vlan 15
SW-1(config-if)# switchport nonegotiate
SW-1(config-if)# no shutdown

SW-2(config)# interface f0/23
SW-2(config-if)# switchport mode trunk
SW-2(config-if)# switchport trunk native vlan 15
SW-2(config-if)# switchport nonegotiate
SW-2(config-if)# no shutdown
```

Part 3: Enable VLAN 20 as a Management VLAN

The network administrator wants to access all switch and routing devices using a management PC. For security purposes, the administrator wants to ensure that all managed devices are on a separate VLAN.

Step 1: Enable a management VLAN (VLAN 20) on SW-A.

a. Enable VLAN 20 on **SW-A**.

```
SW-A(config)# vlan 20
SW-A(config-vlan)# exit
```

b. Create an interface VLAN 20 and assign an IP address within the 192.168.20.0/24 network.

```
SW-A(config)# interface vlan 20
SW-A(config-if)# ip address 192.168.20.1 255.255.255.0
```

Step 2: Enable the same management VLAN on all other switches.

a. Create the management VLAN on all switches: **SW-B**, **SW-1**, **SW-2**, and **Central**.

```
SW-B(config)# vlan 20
SW-B(config-vlan)# exit

SW-1(config)# vlan 20
SW-1(config-vlan)# exit

SW-2(config)# vlan 20
SW-2(config-vlan)# exit

Central(config)# vlan 20
Central(config-vlan)# exit
```

b. Create an interface VLAN 20 on all switches and assign an IP address within the 192.168.20.0/24 network.

```
SW-B(config)# interface vlan 20
SW-B(config-if)# ip address 192.168.20.2 255.255.255.0

SW-1(config)# interface vlan 20
SW-1(config-if)# ip address 192.168.20.3 255.255.255.0

SW-2(config)# interface vlan 20
SW-2(config-if)# ip address 192.168.20.4 255.255.255.0

Central(config)# interface vlan 20
Central(config-if)# ip address 192.168.20.5 255.255.255.0
```

Step 3: Connect and configure the management PC.

Connect the management PC to **SW-A** port F0/1 and ensure that it is assigned an available IP address within the 192.168.20.0/24 network.

Step 4: On SW-A, ensure the management PC is part of VLAN 20.

Interface F0/1 must be part of VLAN 20.

```
SW-A(config)# interface f0/1
SW-A(config-if)# switchport access vlan 20 SW-A(config-if)#
no shutdown
```

Step 5: Verify connectivity of the management PC to all switches.

The management PC should be able to ping **SW-A**, **SW-B**, **SW-1**, **SW-2**, and **Central**.

Part 4: Enable the Management PC to Access Router R1

Step 1: Enable a new subinterface on router R1.

a. Create subinterface g0/0.3 and set encapsulation to dot1q 20 to account for VLAN 20.

```
R1(config)# interface g0/0.3
R1(config-subif)# encapsulation dot1q 20
```

b. Assign an IP address within the 192.168.20.0/24 network.

```
R1(config)# interface g0/0.3
R1(config-subif)# ip address 192.168.20.100 255.255.255.0
```

Step 2: Verify connectivity between the management PC and R1.

Be sure to configure the default gateway on the management PC to allow for connectivity.

Step 3: Enable security.

While the management PC must be able to access the router, no other PC should be able to access the management VLAN.

a. Create an ACL that allows only the Management PC to access the router. Example: (may vary from student configuration)

```
R1(config)# access-list 101 deny ip any 192.168.20.0 0.0.0.255
R1(config)# access-list 101 permit ip any any


R1(config)# access-list 102 permit ip host 192.168.20.50 any
```

b. Apply the ACL to the proper interface(s).

Example: (may vary from student configuration)

```
R1(config)# interface g0/0.1
R1(config-subif)# ip access-group 101 in
R1(config-subif)# interface g0/0.2
R1(config-subif)# ip access-group 101 in
R1(config-subif)# line vty 0 4
R1(config-line)# access-class 102 in
```

Note: Access list 102 is used to only allow the Management PC (192.168.20.50 in this example) to access the router. This prevents an IP address change to bypass the ACL.

Note: There are multiple ways in which an ACL can be created to accomplish the necessary security. For this reason, grading on this portion of the activity is based on the correct connectivity requirements. The management PC must be able to connect to all switches and the router. All other PCs should not be able to connect to any devices within the management VLAN.

Step 4: Verify security.

a. Verify only the Management PC can access the router. Use SSH to access **R1** with username **SSHadmin** and password **ciscosshpa55**.

```
PC> ssh -l SSHadmin 192.168.20.100
```

b. From the management PC, ping **SW-A**, **SW-B**, and **R1**. Were the pings successful? Explain.

____

The pings should have been successful because all devices within the 192.168.20.0 network should be able to ping one another. Devices within VLAN20 are not required to route through the router.

c. From **D1**, ping the management PC. Were the pings successful? Explain.

_

The ping should have failed because for a device within a different VLAN to successfully ping a device within VLAN20, it must be routed. The router has an ACL that prevents all packets from accessing the 192.168.20.0 network.

Step 5: Check results.

Your completion percentage should be 100%. Click **Check Results** to view feedback and verification of which required components have been completed.

If all components appear to be correct and the activity still shows incomplete, it could be due to the connectivity tests that verify the ACL operation.

```
!!!   Script   for   SW-1   conf   t
interface f0/23  switchport mode
trunk   switchport  trunk  native
vlan  15   switchport  nonegotiate
no   shutdown   vlan   20    exit
interface vlan 20
 ip address 192.168.20.3 255.255.255.0
```

```
!!!   Script   for   SW-2   conf   t
interface f0/23  switchport mode
trunk   switchport  trunk  native
vlan  15   switchport  nonegotiate
no   shutdown   vlan   20    exit
interface vlan 20
 ip address 192.168.20.4 255.255.255.0
```

```
!!! Script for SW-A
conf   t   vlan   20
exit      interface
vlan 20
 ip address 192.168.20.1 255.255.255.0
interface f0/1  switchport access vlan
20  no shutdown
```

```
!!! Script for SW-B conf t vlan 20  exit
interface   vlan   20     ip   address
192.168.20.2 255.255.255.0
```

!!! Script for Central `conf t vlan 20 exit`

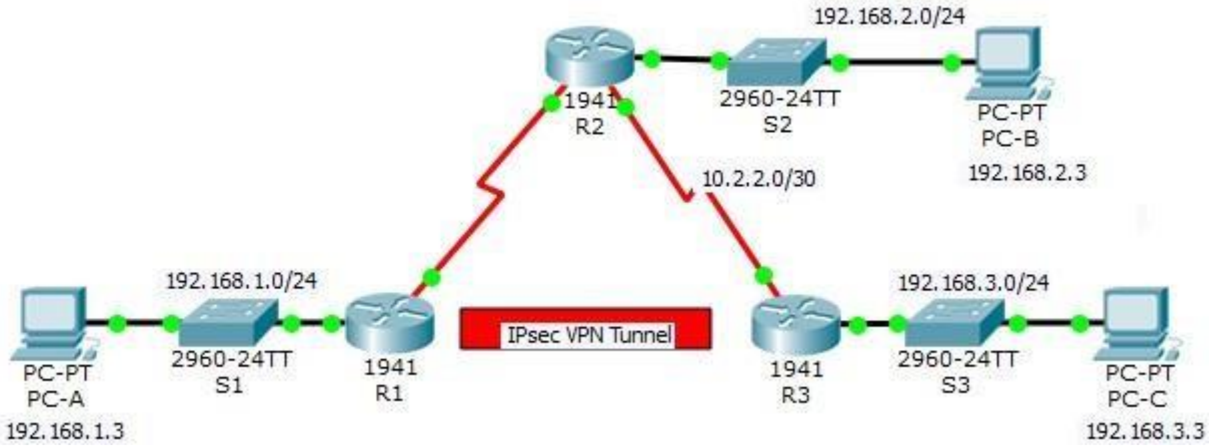
```
interface vlan 20 ip address
192.168.20.5 255.255.255.0
```

!!! Script for R1 `conf`

```
t
interface GigabitEthernet0/0.1 ip access-group
101 in interface GigabitEthernet0/0.2 ip access-
group 101 in interface g0/0.3 encapsulation dot1q
20 ip address 192.168.20.100 255.255.255.0
access-list 101 deny ip any 192.168.20.0 0.0.0.255
access-list 101 permit ip any any access-list
102 permit ip host 192.168.20.50 any line vty 0
4
 access-class 102 in
```

Practical 10: Configure and Verify a Site-to-Site IPsec VPN Using CLI

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/0	192.168.1.1	255.255.255.0	N/A	S1 F0/1
	S0/0/0 (DCE)	10.1.1.2	255.255.255.252	N/A	N/A
R2	G0/0	192.168.2.1	255.255.255.0	N/A	S2 F0/2
	S0/0/0	10.1.1.1	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.1	255.255.255.252	N/A	N/A
R3	G0/0	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.2	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/2
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1	S2 F0/1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Objectives

- Verify connectivity throughout the network.
- Configure R1 to support a site-to-site IPsec VPN with R3.

Background / Scenario

The network topology shows three routers. Your task is to configure R1 and R3 to support a site-to-site IPsec VPN when traffic flows between their respective LANs. The IPsec VPN tunnel is from R1 to R3 via R2. R2 acts as a pass-through and has no knowledge of the VPN. IPsec provides secure transmission of sensitive information over unprotected networks, such as the Internet. IPsec operates at the network layer and protects and authenticates IP packets between participating IPsec devices (peers), such as Cisco routers. **ISAKMP Phase 1 Policy Parameters**

Parameters		R1	R3
Key Distribution Method	Manual or **ISAKMP**	**ISAKMP**	**ISAKMP**
Encryption Algorithm	**DES**, 3DES, or AES	AES 256	AES 256
Hash Algorithm	MD5 or **SHA-1**	**SHA-1**	**SHA-1**
Authentication Method	Pre-shared keys or **RSA**	pre-share	pre-share
Key Exchange	DH Group 1, 2, or 5	DH 5	DH 5
IKE SA Lifetime	86400 seconds or less	**86400**	**86400**
ISAKMP Key		vpnpa55	vpnpa55

Note: Bolded parameters are defaults. Only unbolded parameters have to be explicitly configured.

IPsec Phase 2 Policy Parameters

Parameters	R1	R3
Transform Set Name	VPN-SET	VPN-SET
ESP Transform Encryption	esp-aes	esp-aes
ESP Transform Authentication	esp-sha-hmac	esp-sha-hmac
Peer IP Address	10.2.2.2	10.1.1.2
Traffic to be Encrypted	access-list 110 (source 192.168.1.0 dest 192.168.3.0)	access-list 110 (source 192.168.3.0 dest 192.168.1.0)
Crypto Map Name	VPN-MAP	VPN-MAP
SA Establishment	ipsec-isakmp	ipsec-isakmp

The routers have been pre-configured with the following:

- Password for console line: **ciscoconpa55**
- Password for vty lines: **ciscovtypa55**
- Enable password: **ciscoenpa55**
- SSH username and password: **SSHadmin** / **ciscosshpa55**
- OSPF 101

Part 1: Configure IPsec Parameters on R1

Step 1: Test connectivity.

Ping from PC-A to PC-C.

Step 2: Enable the Security Technology package.

a. On R1, issue the **show version** command to view the Security Technology package license information.

b. If the Security Technology package has not been enabled, use the following command to enable the package.

```
R1(config)# license boot module c1900 technology-package securityk9
```

c. Accept the end-user license agreement.

d. Save the running-config and reload the router to enable the security license.

e. Verify that the Security Technology package has been enabled by using the **show version** command.

Step 3: Identify interesting traffic on R1.

Configure ACL 110 to identify the traffic from the LAN on R1 to the LAN on R3 as interesting. This interesting traffic will trigger the IPsec VPN to be implemented when there is traffic between the R1 to R3 LANs. All other traffic sourced from the LANs will not be encrypted. Because of the implicit **deny all**, there is no need to configure a **deny ip any any** statement.

```
R1(config)# access-list 110 permit ip 192.168.1.0 0.0.0.255 192.168.3.0
0.0.0.255
```

Step 4: Configure the IKE Phase 1 ISAKMP policy on R1.

Configure the **crypto ISAKMP policy 10** properties on R1 along with the shared crypto key **vpnpa55**. Refer to the ISAKMP Phase 1 table for the specific parameters to configure. Default values do not have to be configured. Therefore, only the encryption method, key exchange method, and DH method must be configured.

Note: The highest DH group currently supported by Packet Tracer is group 5. In a production network, you would configure at least DH 14.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 5
R1(config-isakmp)# exit
R1(config)# crypto isakmp key vpnpa55 address 10.2.2.2
```

Step 5: Configure the IKE Phase 2 IPsec policy on R1.

a. Create the transform-set VPN-SET to use **esp-aes** and **esp-sha-hmac**.

```
R1(config)# crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
```

b. Create the crypto map VPN-MAP that binds all of the Phase 2 parameters together. Use sequence number 10 and identify it as an ipsec-isakmp map.

```
R1(config)# crypto map VPN-MAP 10 ipsec-isakmp
R1(config-crypto-map)# description VPN connection to R3
R1(config-crypto-map)# set peer 10.2.2.2
R1(config-crypto-map)# set transform-set VPN-SET
R1(config-crypto-map)# match address 110 R1(config-crypto-
map)# exit
```

Step 6: Configure the crypto map on the outgoing interface.

Bind the **VPN-MAP** crypto map to the outgoing Serial 0/0/0 interface.

```
R1(config)# interface s0/0/0
R1(config-if)# crypto map VPN-MAP
```

Part 2: Configure IPsec Parameters on R3

Step 1: Enable the Security Technology package.

a. On R3, issue the **show version** command to verify that the Security Technology package license information has been enabled.

b. If the Security Technology package has not been enabled, enable the package and reload R3.

Step 2: Configure router R3 to support a site-to-site VPN with R1.

Configure reciprocating parameters on R3. Configure ACL 110 identifying the traffic from the LAN on R3 to the LAN on R1 as interesting.

```
R3(config)# access-list 110 permit ip 192.168.3.0 0.0.0.255 192.168.1.0
0.0.0.255
```

Step 3: Configure the IKE Phase 1 ISAKMP properties on R3. Configure the crypto ISAKMP

policy 10 properties on R3 along with the shared crypto key vpnpa55.

```
R3(config)# crypto isakmp policy 10
R3(config-isakmp)# encryption aes 256
R3(config-isakmp)# authentication pre-share R3(config-isakmp)#
group 5
R3(config-isakmp)# exit
R3(config)# crypto isakmp key vpnpa55 address 10.1.1.2
```

Step 4: Configure the IKE Phase 2 IPsec policy on R3.

a. Create the transform-set VPN-SET to use **esp-aes** and **esp-sha-hmac**.

```
R3(config)# crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
```

b. Create the crypto map VPN-MAP that binds all of the Phase 2 parameters together. Use sequence number 10 and identify it as an ipsec-isakmp map.

```
R3(config)# crypto map VPN-MAP 10 ipsec-isakmp
R3(config-crypto-map)# description VPN connection to R1
R3(config-crypto-map)# set peer 10.1.1.2
R3(config-crypto-map)# set transform-set VPN-SET
R3(config-crypto-map)# match address 110
R3(config-crypto-map)# exit
```

Step 5: Configure the crypto map on the outgoing interface. Bind the VPN-MAP crypto

map to the outgoing Serial 0/0/1 interface. **Note**: This is not graded.

```
R3(config)# interface s0/0/1
R3(config-if)# crypto map VPN-MAP
```

Part 3: Verify the IPsec VPN

Step 1: Verify the tunnel prior to interesting traffic.

Issue the **show crypto ipsec sa** command on R1. Notice that the number of packets encapsulated, encrypted, decapsulated, and decrypted are all set to 0.

Step 2: Create interesting traffic.

Ping PC-C from PC-A.

Step 3: Verify the tunnel after interesting traffic.

On R1, re-issue the **show crypto ipsec sa** command. Notice that the number of packets is more than 0, which indicates that the IPsec VPN tunnel is working.

Step 4: Create uninteresting traffic.

Ping PC-B from PC-A. **Note**: Issuing a ping from router R1 to PC-C or R3 to PC-A is not interesting traffic.

Step 5: Verify the tunnel.

On R1, re-issue the **show crypto ipsec sa** command. Notice that the number of packets has not changed, which verifies that uninteresting traffic is not encrypted.

Step 6: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.

!!! Script for R1

```
config t

license boot module c1900 technology-package securityk9
yes end
copy running-config startup-config
 reload
config t
access-list 110 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
crypto isakmp policy 10  encryption aes 256  authentication pre-
share  group 5  exit
crypto isakmp key vpnpa55 address 10.2.2.2 crypto ipsec
transform-set VPN-SET esp-aes esp-sha-hmac crypto map
VPN-MAP 10 ipsec-isakmp  description VPN connection to
R3  set peer 10.2.2.2  set transform-set VPN-SET  match
address 110  exit
interface S0/0/0
crypto map VPN-MAP
```

!!! Script for R3

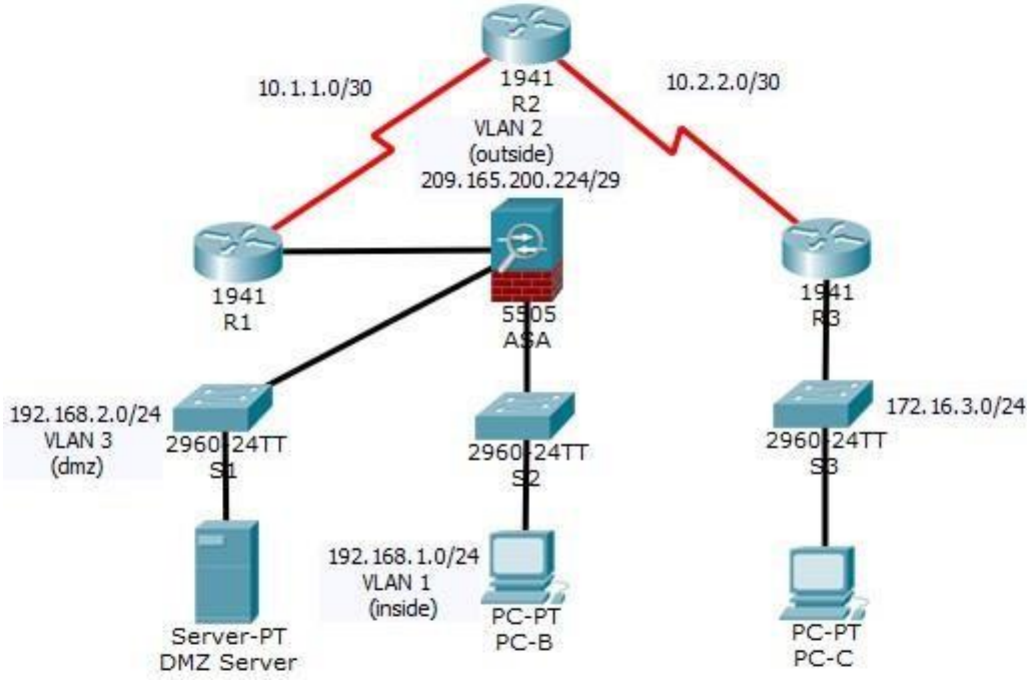
```
config t

access-list 110 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
crypto isakmp policy 10
 encryption aes 256  authentication pre-share  group 5  exit
```

```
crypto isakmp key vpnpa55 address 10.1.1.2 crypto ipsec
transform-set VPN-SET esp-aes esp-sha-hmac  crypto map
VPN-MAP 10 ipsec-isakmp  description VPN connection to
R1  set peer 10.1.1.2
 set transform-set VPN-SET  match address 110  exit
interface S0/0/1  crypto map VPN-MAP
```

Packet Tracer - Configuring ASA Basic Settings and Firewall Using CLI (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Topology



IP Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	209.165.200.225	255.255.255.248	N/A
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A
R3	G0/1	172.16.3.1	255.255.255.0	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
ASA	VLAN 1 (E0/1)	192.168.1.1	255.255.255.0	NA
ASA	VLAN 2 (E0/0)	209.165.200.226	255.255.255.248	NA
ASA	VLAN 3 (E0/2)	192.168.2.1	255.255.255.0	NA
DMZ Server	NIC	192.168.2.3	255.255.255.0	192.168.2.1
PC-B	NIC	192.168.1.3	255.255.255.0	192.168.1.1

PC-C	NIC	172.16.3.3	255.255.255.0	172.16.3.1

Objectives

- Verify connectivity and explore the ASA
- Configure basic ASA settings and interface security levels using CLI
- Configure routing, address translation, and inspection policy using CLI
- Configure DHCP, AAA, and SSH
- Configure a DMZ, Static NAT, and ACLs

Scenario

Your company has one location connected to an ISP. R1 represents a CPE device managed by the ISP. R2 represents an intermediate Internet router. R3 represents an ISP that connects an administrator from a network management company, who has been hired to remotely manage your network. The ASA is an edge CPE security device that connects the internal corporate network and DMZ to the ISP while providing NAT and DHCP services to inside hosts. The ASA will be configured for management by an administrator on the internal network and by the remote administrator. Layer 3 VLAN interfaces provide access to the three areas created in the activity: Inside, Outside, and DMZ. The ISP assigned the public IP address space of 209.165.200.224/29, which will be used for address translation on the ASA.

All router and switch devices have been preconfigured with the following:

- Enable password: **ciscoenpa55** o Console password:

  ciscoconpa55
- Admin username and password: **admin/adminpa55**

Note: This Packet Tracer activity is not a substitute for the ASA labs. This activity provides additional practice and simulates most of the ASA 5505 configurations. When compared to a real ASA 5505, there may be slight differences in command output or commands that are not yet supported in Packet Tracer.

Part 1: Verify Connectivity and Explore the ASA

Note: This Packet Tracer activity starts with 20% of the assessment items marked as complete. This is to ensure that you do not inadvertently change some ASA default values. For example, the default name of the inside interface is “inside” and should not be changed. Click **Check Results** to see which assessment items are already scored as correct.

Step 1: Verify connectivity.

The ASA is not currently configured. However, all routers, PCs, and the DMZ server are configured. Verify that PC-C can ping any router interface. PC-C is unable to ping the ASA, PC-B, or the DMZ server.

Step 2: Determine the ASA version, interfaces, and license.

Use the **show version** command to determine various aspects of this ASA device.

Step 3: Determine the file system and contents of flash memory.

a. Enter privileged EXEC mode. A password has not been set. Press **Enter** when prompted for a password.

b. Use the **show file system** command to display the ASA file system and determine which prefixes are supported.

c. Use the **show flash:** or **show disk0:** command to display the contents of flash memory.

Part 2: Configure ASA Settings and Interface Security Using the CLI

Tip: Many ASA CLI commands are similar to, if not the same, as those used with the Cisco IOS CLI. In addition, the process of moving between configuration modes and submodes is essentially the same.

Step 1: Configure the hostname and domain name.

a. Configure the ASA hostname as **CCNAS-ASA**.

b. Configure the domain name as **ccnasecurity.com**.

Step 2: Configure the enable mode password.

Use the **enable password** command to change the privileged EXEC mode password to **ciscoenpa55**.

Step 3: Set the date and time.

Use the **clock set** command to manually set the date and time (this step is not scored).

Step 4: Configure the inside and outside interfaces.

You will only configure the VLAN 1 (inside) and VLAN 2 (outside) interfaces at this time. The VLAN 3 (dmz) interface will be configured in Part 5 of the activity.

a. Configure a logical VLAN 1 interface for the inside network (192.168.1.0/24) and set the security level to the highest setting of 100.

```
CCNAS-ASA(config)# interface vlan 1 CCNAS-ASA(config-if)#
nameif inside
CCNAS-ASA(config-if)# ip address 192.168.1.1 255.255.255.0 CCNAS-ASA(config-
if)# security-level 100
```

b. Create a logical VLAN 2 interface for the outside network (209.165.200.224/29), set the security level to the lowest setting of 0, and enable the VLAN 2 interface.

```
CCNAS-ASA(config-if)# interface vlan 2
CCNAS-ASA(config-if)# nameif outside
CCNAS-ASA(config-if)# ip address 209.165.200.226 255.255.255.248 CCNAS-
ASA(config-if)# security-level 0
```

c. Use the following verification commands to check your configurations:

1) Use the **show interface ip brief** command to display the status for all ASA interfaces. **Note**: This command is different from the IOS command **show ip interface brief**. If any of the physical or logical interfaces previously configured are not up/up, troubleshoot as necessary before continuing.

   Tip: Most ASA **show** commands, including **ping**, **copy**, and others, can be issued from within any configuration mode prompt without the **do** command.

2) Use the **show ip address** command to display the information for the Layer 3 VLAN interfaces.

3) Use the **show switch vlan** command to display the inside and outside VLANs configured on the ASA and to display the assigned ports.

Step 5: Test connectivity to the ASA.

a. You should be able to ping from PC-B to the ASA inside interface address (192.168.1.1). If the pings fail, troubleshoot the configuration as necessary.

b. From PC-B, ping the VLAN 2 (outside) interface at IP address 209.165.200.226. You should not be able to ping this address.

Part 3: Configure Routing, Address Translation, and Inspection Policy Using the CLI

Step 1: Configure a static default route for the ASA.

Configure a default static route on the ASA outside interface to enable the ASA to reach external networks.

a. Create a "quad zero" default route using the **route** command, associate it with the ASA outside interface, and point to the R1 G0/0 IP address (209.165.200.225) as the gateway of last resort.

```
CCNAS-ASA(config)# route outside 0.0.0.0 0.0.0.0 209.165.200.225
```

b. Issue the **show route** command to verify the static default route is in the ASA routing table.

c. Verify that the ASA can ping the R1 S0/0/0 IP address 10.1.1.1. If the ping is unsuccessful, troubleshoot as necessary.

Step 2: Configure address translation using PAT and network objects.

a. Create network object **inside-net** and assign attributes to it using the **subnet** and **nat** commands.

```
CCNAS-ASA(config)# object network inside-net
CCNAS-ASA(config-network-object)# subnet 192.168.1.0 255.255.255.0
CCNAS-ASA(config-network-object)# nat (inside,outside) dynamic interface
CCNAS-ASA(config-network-object)#  end
```

b. The ASA splits the configuration into the object portion that defines the network to be translated and the actual **nat** command parameters. These appear in two different places in the running configuration. Display the NAT object configuration using the **show run** command.

c. From PC-B attempt to ping the R1 G0/0 interface at IP address 209.165.200.225. The pings should fail.

d. Issue the **show nat** command on the ASA to see the translated and untranslated hits. Notice that, of the pings from PC-B, four were translated and four were not. The outgoing pings (echos) were translated and sent to the destination. The returning echo replies were blocked by the firewall policy. You will configure the default inspection policy to allow ICMP in Step 3 of this part of the activity.

Step 3: Modify the default MPF application inspection global service policy.

For application layer inspection and other advanced options, the Cisco MPF is available on ASAs.

The Packet Tracer ASA device does not have an MPF policy map in place by default. As a modification, we can create the default policy map that will perform the inspection on inside-to-outside traffic. When configured correctly only traffic initiated from the inside is allowed back in to the outside interface. You will need to add ICMP to the inspection list.

a. Create the class-map, policy-map, and service-policy. Add the inspection of ICMP traffic to the policy map list using the following commands:

```
CCNAS-ASA(config)# class-map inspection_default
CCNAS-ASA(config-cmap)# match default-inspection-traffic
CCNAS-ASA(config-cmap)# exit
CCNAS-ASA(config)# policy-map global_policy
CCNAS-ASA(config-pmap)# class inspection_default
CCNAS-ASA(config-pmap-c)# inspect icmp
CCNAS-ASA(config-pmap-c)# exit
CCNAS-ASA(config)# service-policy global_policy global
```

b. From PC-B, attempt to ping the R1 G0/0 interface at IP address 209.165.200.225. The pings should be successful this time because ICMP traffic is now being inspected and legitimate return traffic is being allowed. If the pings fail, troubleshoot your configurations.

Part 4: Configure DHCP, AAA, and SSH

Step 1: Configure the ASA as a DHCP server.

a. Configure a DHCP address pool and enable it on the ASA inside interface.

```
CCNAS-ASA(config)# dhcpd address 192.168.1.5-192.168.1.36 inside
```

b. (Optional) Specify the IP address of the DNS server to be given to clients.

```
CCNAS-ASA(config)# dhcpd dns 209.165.201.2 interface inside
```

c. Enable the DHCP daemon within the ASA to listen for DHCP client requests on the enabled interface (inside).

```
CCNAS-ASA(config)# dhcpd enable inside
```

d. Change PC-B from a static IP address to a DHCP client, and verify that it receives IP addressing information. Troubleshoot, as necessary to resolve any problems.

Step 2: Configure AAA to use the local database for authentication.

a. Define a local user named **admin** by entering the **username** command. Specify a password of **adminpa55**.

```
CCNAS-ASA(config)# username admin password adminpa55
```

b. Configure AAA to use the local ASA database for SSH user authentication.

```
CCNAS-ASA(config)# aaa authentication ssh console LOCAL
```

Step 3: Configure remote access to the ASA.

The ASA can be configured to accept connections from a single host or a range of hosts on the inside or outside network. In this step, hosts from the outside network can only use SSH to communicate with the ASA. SSH sessions can be used to access the ASA from the inside network.

a. Generate an RSA key pair, which is required to support SSH connections. Because the ASA device has RSA keys already in place, enter **no** when prompted to replace them.

```
CCNAS-ASA(config)# crypto key generate rsa modulus 1024
WARNING: You have a RSA keypair already defined named <Default-RSA-Key>.


Do you really want to replace them? [yes/no]: no
ERROR: Failed to create new RSA keys named <Default-RSA-Key>
```

b. Configure the ASA to allow SSH connections from any host on the inside network (192.168.1.0/24) and from the remote management host at the branch office (172.16.3.3) on the outside network. Set the SSH timeout to 10 minutes (the default is 5 minutes).

```
CCNAS-ASA(config)# ssh 192.168.1.0 255.255.255.0 inside
CCNAS-ASA(config)# ssh 172.16.3.3 255.255.255.255 outside CCNAS-ASA(config)#
ssh timeout 10
```

c. Establish an SSH session from PC-C to the ASA (209.165.200.226). Troubleshoot if it is not successful.

```
PC> ssh -l admin 209.165.200.226
```

d. Establish an SSH session from PC-B to the ASA (192.168.1.1). Troubleshoot if it is not successful.

```
PC> ssh -l admin 192.168.1.1
```

Part 5: Configure a DMZ, Static NAT, and ACLs

R1 G0/0 and the ASA outside interface already use 209.165.200.225 and .226, respectively. You will use public address 209.165.200.227 and static NAT to provide address translation access to the server.

Step 1: Configure the DMZ interface VLAN 3 on the ASA.

a. Configure DMZ VLAN 3, which is where the public access web server will reside. Assign it IP address 192.168.2.1/24, name it **dmz**, and assign it a security level of 70. Because the server does not need to initiate communication with the inside users, disable forwarding to interface VLAN 1.

```
CCNAS-ASA(config)# interface vlan 3
CCNAS-ASA(config-if)# ip address 192.168.2.1 255.255.255.0 CCNAS-
ASA(config-if)# no forward interface vlan 1 CCNAS-ASA(config-if)#
nameif dmz
INFO: Security level for "dmz" set to 0 by default. CCNAS-
ASA(config-if)# security-level 70
```

b. Assign ASA physical interface E0/2 to DMZ VLAN 3 and enable the interface.

```
CCNAS-ASA(config-if)# interface Ethernet0/2
CCNAS-ASA(config-if)# switchport access vlan 3
```

c. Use the following verification commands to check your configurations:

1) Use the **show interface ip brief** command to display the status for all ASA interfaces.

2) Use the **show ip address** command to display the information for the Layer 3 VLAN interfaces.

3) Use the **show switch vlan** command to display the inside and outside VLANs configured on the ASA and to display the assigned ports.

Step 2: Configure static NAT to the DMZ server using a network object.

Configure a network object named **dmz-server** and assign it the static IP address of the DMZ server (192.168.2.3). While in object definition mode, use the **nat** command to specify that this object is used to translate a DMZ address to an outside address using static NAT, and specify a public translated address of 209.165.200.227.

```
CCNAS-ASA(config)# object network dmz-server
CCNAS-ASA(config-network-object)# host 192.168.2.3
CCNAS-ASA(config-network-object)# nat (dmz,outside) static 209.165.200.227
CCNAS-ASA(config-network-object)#  exit
```

Step 3: Configure an ACL to allow access to the DMZ server from the Internet.

Configure a named access list **OUTSIDE-DMZ** that permits the TCP protocol on port 80 from any external host to the internal IP address of the DMZ server. Apply the access list to the ASA outside interface in the “IN” direction.

```
CCNAS-ASA(config)# access-list OUTSIDE-DMZ permit icmp any host 192.168.2.3
CCNAS-ASA(config)# access-list OUTSIDE-DMZ permit tcp any host 192.168.2.3 eq
80
```

```
CCNAS-ASA(config)# access-group OUTSIDE-DMZ in interface outside
```

Note: Unlike IOS ACLs, the ASA ACL permit statement must permit access to the internal private DMZ address. External hosts access the server using its public static NAT address, the ASA translates it to the internal host IP address, and then applies the ACL.

Step 4: Test access to the DMZ server.

At the time this Packet Tracer activity was created, the ability to successfully test outside access to the DMZ web server was not in place; therefore, successful testing is not required.

Step 5: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.

Scripts

ASA

```
enable

!<Enter> for password conf t
hostname CCNAS-ASA domain-name
ccnasecurity.com enable
password ciscoenpa55 clock set
13:52:51 June 10 2015 interface
vlan 1  nameif inside
 ip address 192.168.1.1 255.255.255.0
security-level 100 interface vlan 2
 nameif outside
 ip address 209.165.200.226 255.255.255.248
security-level 0
route outside 0.0.0.0 0.0.0.0 209.165.200.225
object network inside-net  subnet 192.168.1.0
255.255.255.0  nat (inside,outside) dynamic
interface class-map inspection_default  match
default-inspection-traffic  exit
policy-map global_policy
class inspection_default
inspect icmp  exit
service-policy global_policy global dhcpd
address 192.168.1.5-192.168.1.36 inside dhcpd
dns 209.165.201.2 interface inside dhcpd
enable inside username admin password
adminpa55 aaa authentication ssh console
LOCAL crypto key generate rsa modulus 1024 no
ssh 192.168.1.0 255.255.255.0 inside
ssh 172.16.3.3 255.255.255.255 outside
ssh timeout 10 interface vlan 3
 ip address 192.168.2.1 255.255.255.0
no forward interface vlan 1  nameif
dmz  security-level 70 interface
Ethernet0/2  switchport access vlan 3
```

```
object network dmz-server  host
192.168.2.3
 nat (dmz,outside) static 209.165.200.227 access-list
OUTSIDE-DMZ permit icmp any host 192.168.2.3
access-list OUTSIDE-DMZ permit tcp any host 192.168.2.3 eq 80
access-group OUTSIDE-DMZ in interface outside
```

PC-B

-Change from static to DHCP addressing