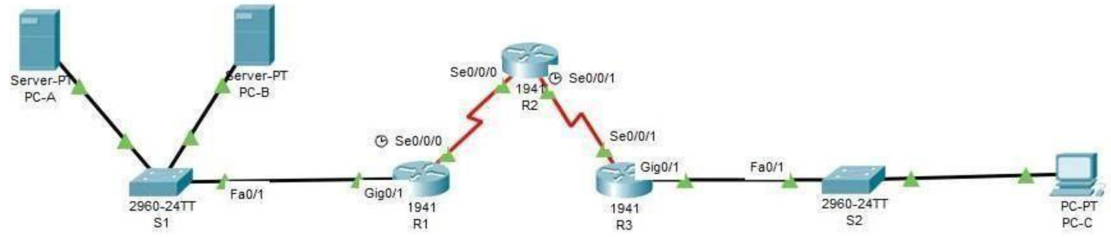
PRACTICAL 1

Aim: Packet Tracer - Configure Cisco Routers for Syslog, NTP, and SSH Operations.

Topology:



Addressing Table:

Device	Interfaces	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0	30.1.1.1	255.255.255.252	N/A
R2	S0/0/0	30.1.1.2	255.255.255.252	N/A
	S0/0/1	40.2.2.2	255.255.255.252	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	40.2.2.1	255.255.255.252	N/A
PC-A	NIC	192.168.1.5	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.1.6	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.5	255.255.255.0	192.168.3.1

❖ **Apply OSPF to alltherouters**

```
R1(config)# router ospf 1
R1(config-router)# network 192.168.1.0 0.0.0.255 area 0
                    network 30.0.0.0 0.255.255.255 area 0
R1(config-router)# exit
R1(config)# exit
R1# show ip ospf

R2(config)# router ospf 1
R2(config-router)# network 30.0.0.0 0.255.255.255 area0
                    network 40.0.0.0 0.255.255.255 area0
R2(config-router)# exit
R2(config)# exit
R2# show ip ospf

R3(config)# router ospf 1
R3(config-router)# network 40.0.0.0 0.255.255.255 area 0
                   network 192.168.3.0 0.0.0.255 area 0
R3(config-router)# exit R3(config)# exit R3# show ip ospf
```

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.

```
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.

```
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.

show ip ospf interface.

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 serviceisenabled.

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

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
show ntp status.
```

Step 3: Configure routers to update hardware clock.

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

clock

Step 4: Configure NTP authentication on the routers.

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
```

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
```

Step 2: Verify logging configuration.

```
show logging
```

Step 3: Examine logs of the Syslog Server.

Part 4: Configure R3 to Support SSH

Connections. Step 1: Configure a domain name.

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

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

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

Step 3: Configure the incoming vty lines on R3.

```
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.

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

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

R3(config)# **crypto key generate rsa**

How many bits in the modulus [512]: 1024

Step 6: Verify the SSH configuration.

show ip ssh

Step 7: Configure SSH timeouts and authentication parameters.

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

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

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.

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.

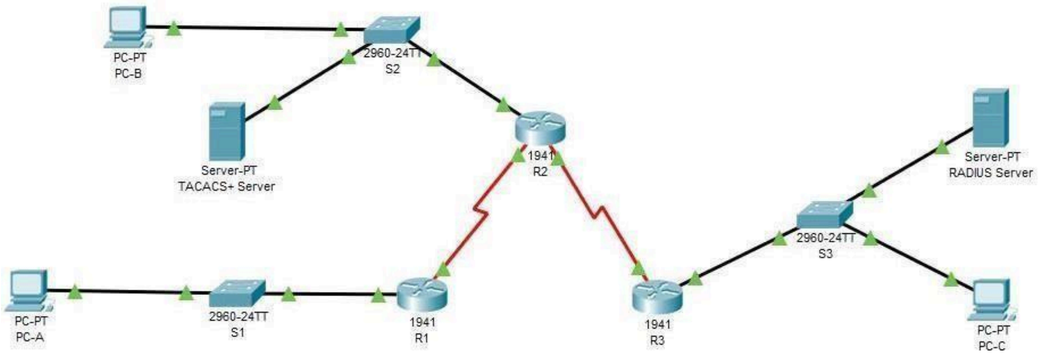
R2# **ssh –v 2 –l SSHadmin**

10.2.2.1 Step 11: Check result.

PRACTICAL 2

Aim: Packet Tracer - Configure AAA Authentication on Cisco Routers

Topology:



Device	Interfaces	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0	30.1.1.1	255.255.255.252	N/A
R2	S0/0/0	30.1.1.2	255.255.255.252	N/A
	S0/0/1	40.2.2.2	255.255.255.252	N/A
	G0/0	192.168.2.1	255.255.255.252	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	40.2.2.1	255.255.255.252	N/A
TACACS+Server	NIC	192.168.2.2	255.255.255.0	192.168.2.1
RADIUS Server	NIC	192.168.3.2	255.255.255.0	192.168.3.1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

Addressing Table:

Part 1: Configure Local AAA Authentication for Console Access on R1 Step 1: Test connectivity.

- Ping from PC-AtoPC-B.
- Ping from PC-AtoPC-C.
- Ping from PC-BtoPC-C.

Step 2: Configure a local username on R1.

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

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

```
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.

```
R1(config)# line console 0

R1(config-line)# login authentication
```

default Step 5: Verify the AAA authentication

method.

```
R1(config-line)#

end R1# exit

Username: Admin1
Password:
admin1pa55
```

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

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

SSH.

```
R1(config)# ip domain-name ccnasecurity.com
R1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
```

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

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

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

```
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.

R2(config)# **username Admin2 secret**

admin2pa55 Step 2: Verify the TACACS+ Server

configuration.

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

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.

```
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.

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

default Step 6: Verify the AAA authentication

method.

```
R2(config-line)# end
R2# exit
Username: Admin2
Password:
admin2pa55
```

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

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

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

Step 2: Verify the RADIUS Server configuration.

Step 3: Configure the RADIUS server specifics

onR3.

```
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.

```
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.

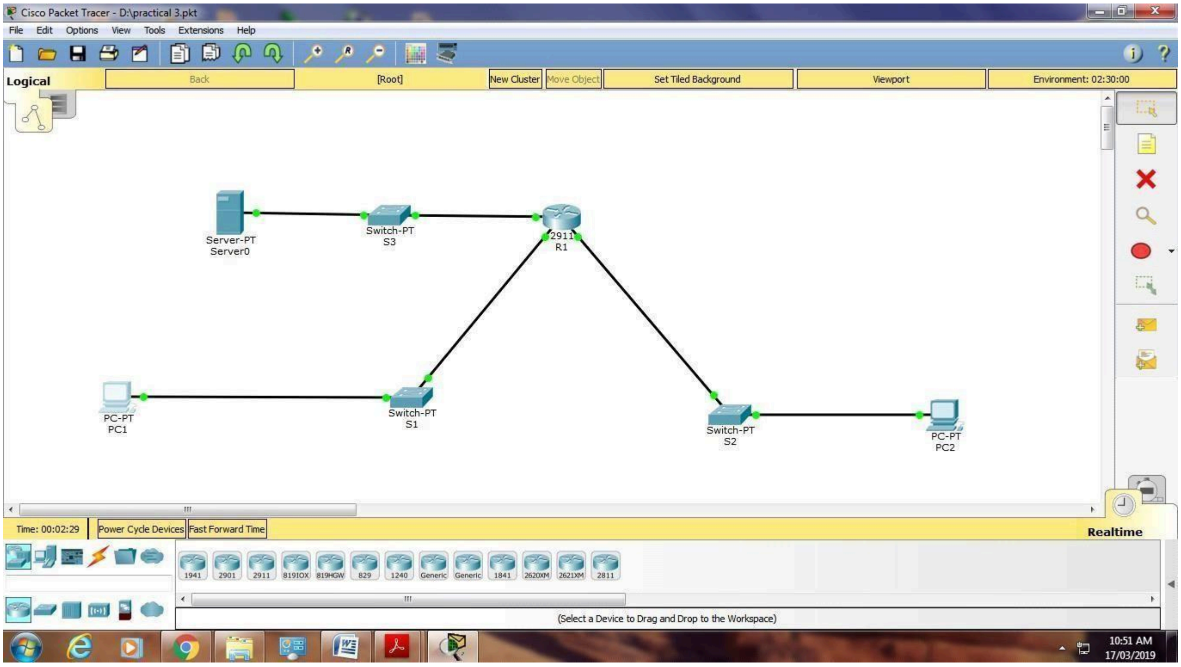
```
R3(config)# line console 0
```

R3(config-line)# **login authentication default Step 6: Verify the AAA authentication method.R3(config-line)#** end **R3#** exit **Username: Admin3 Password: admin3pa55 Step 7: Check results.**

PRACTICAL 3(A)

Aim: 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

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

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

- **From global configuration mode on R1, enter the following command to determinethe first valid number for an extendedaccesslist.**

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

- **Add 100 to the command, followed by aquestionmark.**

  ```
  R1(config)# access-list 100 ?
  ```

- **To permit FTP traffic, enter permit, followed by aquestionmark.**

  ```
  R1(config)# access-list 100 permit ?
  ```

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

  ```
  R1(config)# access-list 100 permit tcp ?
  ```

- **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 aquestionmark.**

  ```
  R1(config)# access-list 100 permit tcp 172.22.34.66 ?
  ```

- **Calculate the wildcard mask determining the binary opposite of a subnet mask. 11111111.11111111.11111111.111**00000=255.255.255.224

  00000000.00000000.00000000.000**11111** = 0.0.0.31

- **Enter the wildcard mask, followed by aquestionmark.**

  ```
  R1(config)# access-list 100 permit tcp 172.22.34.66 0.0.0.31 ?
  ```

- **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 172.22.34.66 0.0.0.31 host 172.22.34.62 ?
  ```

- 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 172.22.34.66 0.0.0.31 host 172.22.34.62 eq ?
  ```

  ```
  R1(config)# access-list 100 permit tcp 172.22.34.66 0.0.0.31 host
  172.22.34.62 eq ftp
  ```

- 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 172.22.34.66 0.0.0.31 host
172.22.34.62
```

All other traffic is denied, bydefault.

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

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

in Step 3: Verify the ACL implementation

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

- FTP from **PC1** to **Server**. The username and password are both **cisco**. **PC>ftp172.22.34.62**

- Exit the FTP service of the **Server**. **ftp>quit**

- Ping from **PC1** to **PC2**. The destination host should be unreachable, because the traffic wasnot explicitlypermitted.

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

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

- **Named ACLs start with the ip keyword. From global configuration mode of R1, enterthe following command, followed by aquestionmark**.

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

- **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 nameiscase-sensitive.)**

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

- **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 172.22.34.98 ?
```

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

```
R1(config-ext-nacl)# permit tcp 172.22.34.98 0.0.0.15 ?
```

- **Finish the statement by specifying the server address as you did in Part 1 and filtering wwwtraffic.**

  R1(config-ext-nacl)# **permit tcp 172.22.34.98 0.0.0.15 host 172.22.34.62 eq www**

- **Create a second access list statement to permit ICMP (ping, etc.) traffic from PC2 toServer. 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 172.22.34.98 0.0.0.15 host 172.22.34.62**

- **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.

R1(config)# **interface gigabitEthernet 0/1**

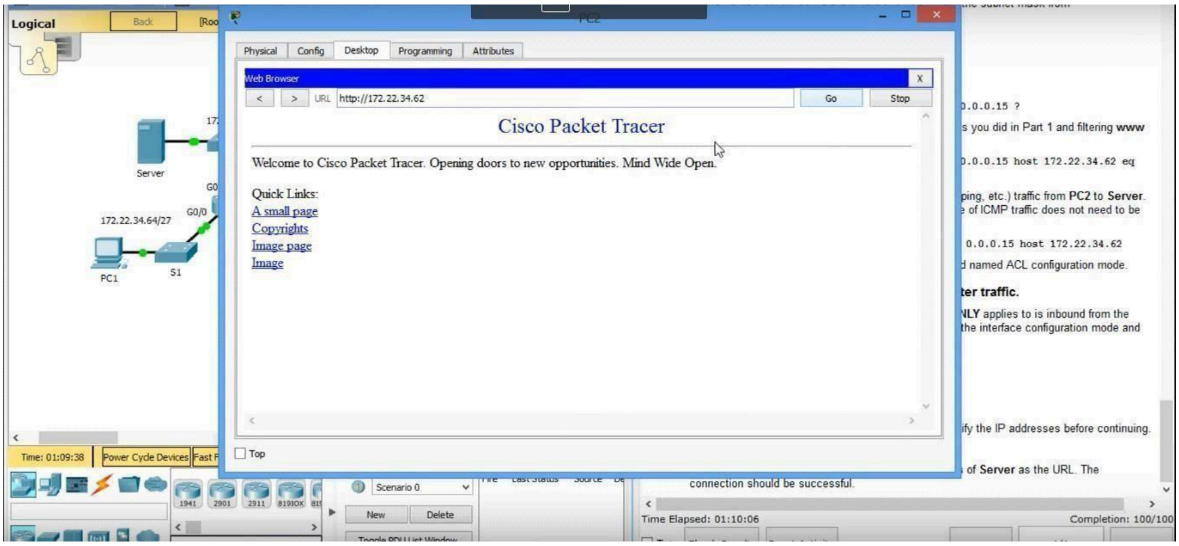
R1(config-if)# **ip access-group HTTP_ONLY in**

Step 3: Verify the ACL implementation.

- **Ping from PC2 to Server. The ping should be successful, if the ping is unsuccessful, verify the IP addressesbeforecontinuing.**

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

- Open the web browser on **PC2** and enter the IP address of **Server** as the URL. The connection shouldbesuccessful.



Web Browser

URL http://172.22.34.62

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

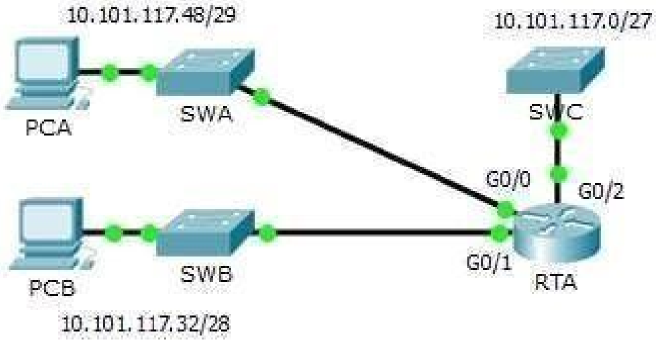
Quick Links:
A small page
Copyrights
Image page
Image

Logical
Back
PC2
Physical
Config
Desktop
Programming
Attributes
X
<
>
Go
Stop
Server
172.22.34.64/27
G0/0
PC1
S1
Time: 01:09:38
Power Cycle Devices
☐ Top
1941
2901
2911
819IOX
Scenario 0
New
Delete
0.0.0.15 ?
s you did in Part 1 and filtering www
0.0.0.15 host 172.22.34.62 eq
ping, etc.) traffic from PC2 to Server.
of ICMP traffic does not need to be
0.0.0.15 host 172.22.34.62
d named ACL configuration mode.
ter traffic.
NLY applies to is inbound from the
the interface configuration mode and
ify the IP addresses before continuing.
of Server as the URL. The
connection should be successful.
Time Elapsed: 01:10:06
Completion: 100/100

PRACTICAL 3(B)

Aim: Configuring Extended ACLs - Scenario 2

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask		Default Gateway
	G0/0	10.101.117.49	255.255.255.248		N/A
RTA	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

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

Configure, apply and verify an ACL to satisfy the followingpolicy:

- SSH traffic from devices on the 10.101.117.32/28 network is allowed to devices on the10.101.117.0/27networks.
- ICMP traffic is allowed from any source toanydestination.

- All other traffic to 10.101.117.0/27isblocked.

Step 1: Configure the extended ACL.

- 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 firstACLstatement:

  - The last extended list numberis199.
  - The protocolisTCP.
  - The source networkis10.101.117.32.
  - The wildcard can be determined by subtracting 255.255.255.240from255.255.255.255.
  - The destination networkis10.101.117.0.
  - The wildcard can be determined by subtracting 255.255.255.224from255.255.255.255.
  - The protocol is SSH(port22).

  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

- 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

- All other IP traffic is denied,bydefault.

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.

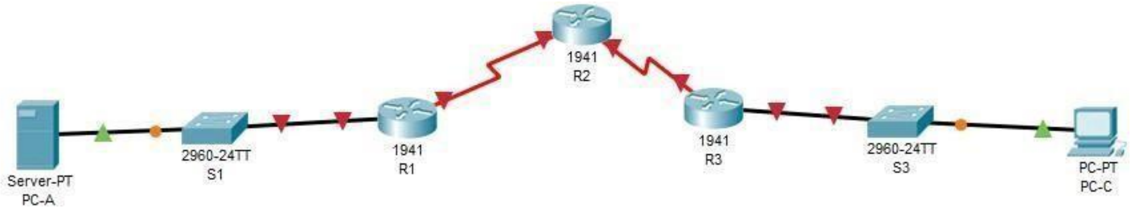
- Ping from **PCB** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addressesbeforecontinuing.
- SSH from **PCB** to **SWC**. The username is **Admin**, and the password is **Adminpa55**.PC>**ssh -l Admin 10.101.117.2**
- Exit the SSH sessionto**SWC**.
- Ping from **PCA** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addressesbeforecontinuing.

- SSH from **PCA** to **SWC**. The access list causes the router to rejecttheconnection

PRACTICAL 4

Aim: Configure IP ACLs to Mitigate Attacks and ACLs

Topology



AddressingTable:

Device	Interfaces	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0	30.1.1.1	255.255.255.252	N/A
R2	S0/0/0	30.1.1.2	255.255.255.252	N/A
	S0/0/1	40.2.2.2	255.255.255.252	N/A
	Lo0	192.168.2.1	255.255.255.0	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	40.2.2.1	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

ForloopbackCLI

R1(config)# **int Lo0**

R1(config)# **ip add 192.168.2.1 255.255.255.0**

R1(config)# **no shut**

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.

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

- 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**

- Open a web browser to the **PC-A** server (192.168.1.3) to display the web page. Closethe browser whendone.

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 host192.168.3.3
R2(config)# access-list 10 permit host192.168.3.3
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)# int g0/1
R1(config-if)# ip access-group 10 in

R2(config)# int s0/0/0
R2(config-if)# ip access-group 10 in
R2(config)# int s0/0/1
R2(config-if)# ip access-group 10 in

R3(config)# int g0/1
R3(config-if)# ip access-group 10 in
```

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

- Establish an SSH session to 192.168.2.1 from **PC-C** (should be successful). PC> **ssh –lSSHadmin192.168.2.1**

- Establish an SSH session to 192.168.2.1 from **PC-A**(shouldfail).

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 onserver**PC-A.**
- Deny any outside host access to HTTPS servicesonPC-A.
- Permit **PC-C** to access **R1** viaSSH.

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-

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 30.1.1.1 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-group120in
```

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

Part 4: Modify an Existing ACLonR1

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.

Go to command prompt **PC-A** and ping **Lo0 (i.e 192.168.2.1)**

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**.

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.

```
R3(config)# access-list 100 permit tcp 10.0.0.0 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 0.15.255.255any

R3(config)# access-list 100 deny ip 192.168.0.0 0.0.255.255any

R3(config)# access-list 100 deny ip 127.0.0.0 0.255.255.255any

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.

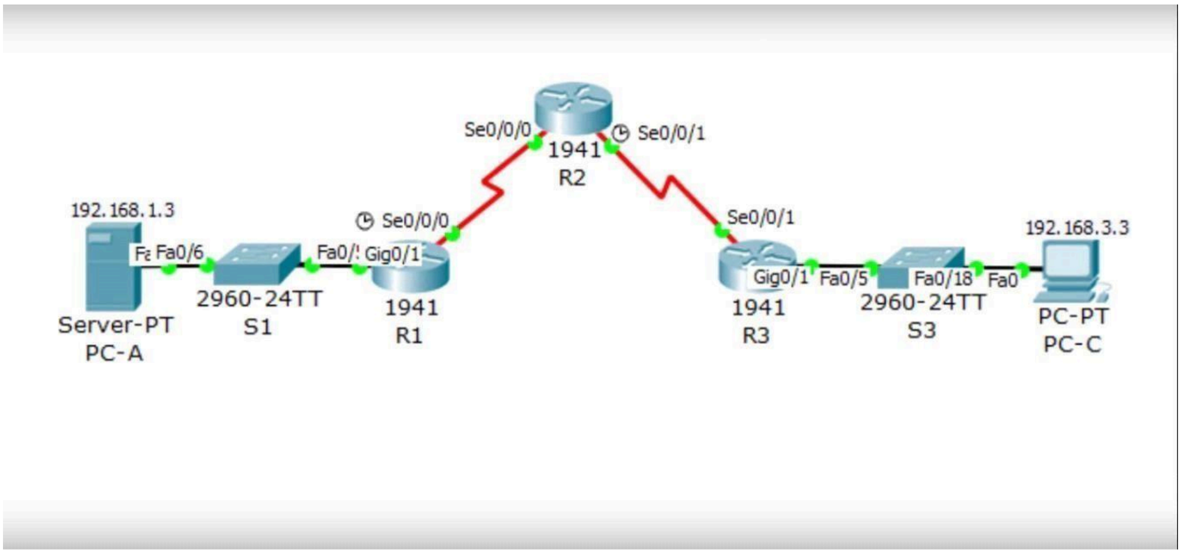
- From the PC-C command prompt, ping the PC-A server. The ICMP echo replies are blocked by the

ACL since they are sourced from the 192.168.0.0/16addressspace.

PRACTICAL 5

Aim: Configuring a Zone – Based policy firewall

Topology



Addressing table

Device	Interfaces	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0	30.1.1.1	255.255.255.252	N/A
R2	S0/0/0	30.1.1.2	255.255.255.252	N/A
	S0/0/1	40.2.2.2	255.255.255.252	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	40.2.2.1	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

Part 1: Verify Basic Network Connectivity

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

Step 2: Access R2 using SSH.

- 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 login.
  PC>**ssh -l Admin 10.2.2.2**

- Exit theSSHsession.

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

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

- Close the browseron**PC-C**.

Part 2: Create the Firewall Zones on R3

Step 1: Enable the Security Technology package.

- On **R3**, issue the **show version** command to view the Technology Packagelicenseinformation.

- If the Security Technology package has not been enabled, use the following command to enablethe package.
  R3(config)# **license boot module c1900 technology-package securityk9**

- Accept the end-user licenseagreement.

- Save the running-config and reload the router to enable the security license. R3# **write**

  R3# **reload**

- Verify that the Security Technology package has been enabled by using the**showversion** 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 toanydestination.

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

```
R3(config-pmap-c)# inspect
    All protocols will be inspected.
R3(config-pmap-c)#
exit R3(config-pmap)#
exit Part 5: Apply
Firewall
```

Policies Step 1: Create a pair

ofzones.

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 OUT-ZONE
```

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

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
```

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 Step 1: From internal PC-C, ping the

externalPC-Aserver.

Step 2: From internal PC-C, SSH to the R2S0/0/1interface.

```
R3# show policy-map type inspect zone-pairsessions
```

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.

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

sessions Step 5: Close the browser on PC-C.

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

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.

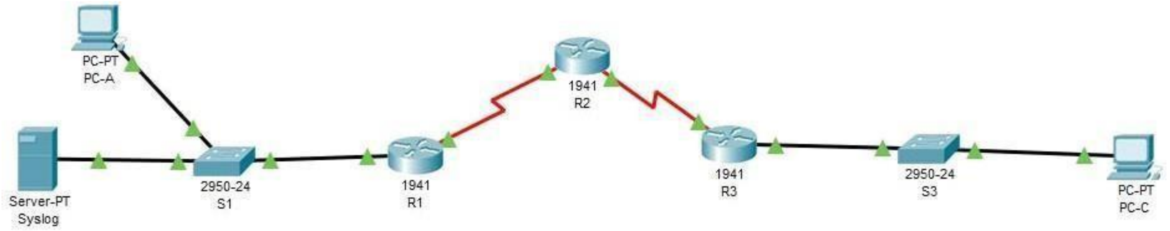
Step 2: From R2, ping PC-C.

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

PRACTICAL 6

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

Topology

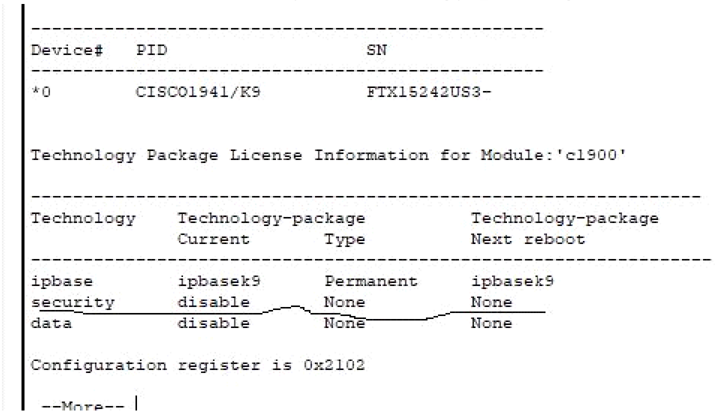


Addressing Table

Device	Interfaces	IP Address	Subnet Mask	Default
				Gateway
R1	G0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0	30.1.1.1	255.255.255.252	N/A
R2	S0/0/0	30.1.1.2	255.255.255.252	N/A
	S0/0/1	40.2.2.2	255.255.255.252	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A
	S0/0/1	40.2.2.1	255.255.255.252	N/A
Syslog	NIC	192.168.1.50	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.2	255.255.255.0	192.168.1.1
PC-C	NIC	192.168.3.2	255.255.255.0	192.168.3.1

Part 1: Enable IOS IPS

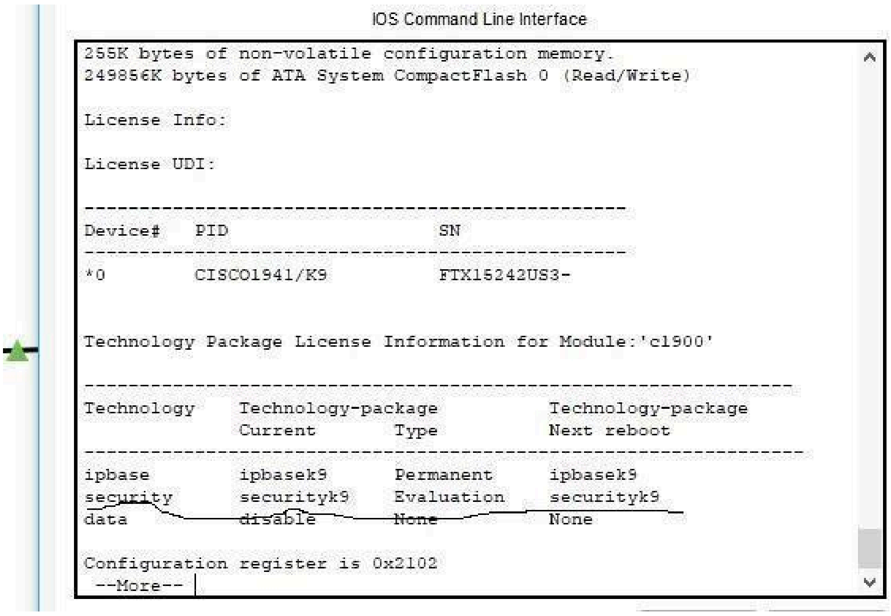
Step 1: Enable the Security Technology package



```
-------------------------------------------------
Device#   PID                     SN
-------------------------------------------------
*0        CISCO1941/K9            FTX15242US3-


Technology Package License Information for Module:'c1900'

----------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent      ipbasek9
security      disable       None           None
data          disable       None           None

Configuration register is 0x2102

 --More--
```

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

R1# **write** R1# **reload**



```
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)

License Info:

License UDI:

-------------------------------------------------
Device#   PID                   SN
-------------------------------------------------
*0        CISCO1941/K9          FTX15242US3-


Technology Package License Information for Module:'c1900'

----------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent      ipbasek9
security      securityk9    Evaluation     securityk9
data          disable       None           None

Configuration register is 0x2102
 --More--
```

Step 2: Verify network connectivity.

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

Step 3: Create an IOS IPS configuration directory in flash

Router# **mkdir ipsdir**

Step 4: Configure the IPS signature storage location

R1(config)# **ip ips config location**

flash:ipsdir Step 5: Create an IPS rule.R1(config)#

ip ips name iosips Step 6: Enable logging

R1(config)# **ip ips notify log**
R1# **clock set 10:20:00 10 january 2014**
R1(config)# **service timestamps log datetime**
msec R1(config)# **logging host 192.168.1.50**

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

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**

Step 8: Apply the IPS rule to an 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.

```
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)#exi
t R1(config-sigdef)# exit
```

Step 2: Use show commands to verify IPS.

```
R1(config)# show ip ips all
```

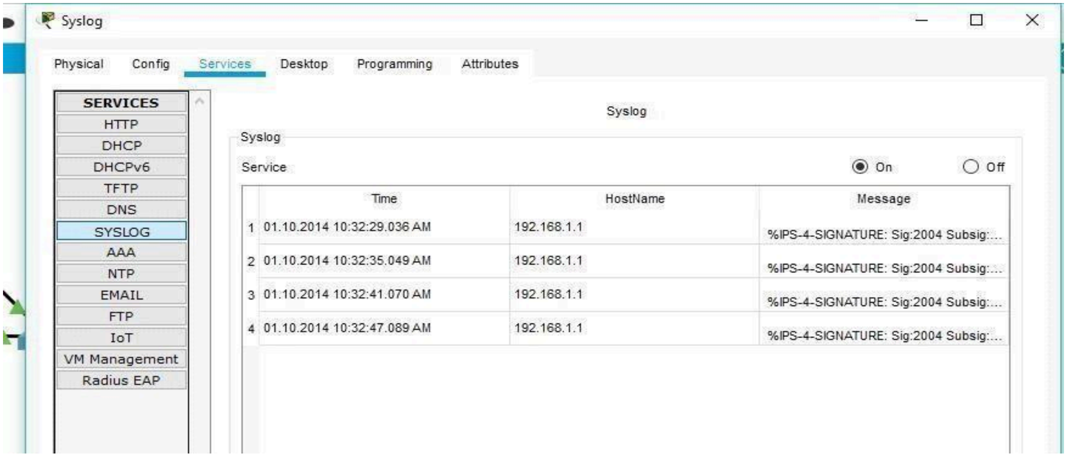
Step 3: Verify that IPS is working properly

Again,

Ping **PC-A** to**PC-C**

Ping **PC-C** to**PC-A**

Step 4: View the syslog messages



Syslog

Syslog

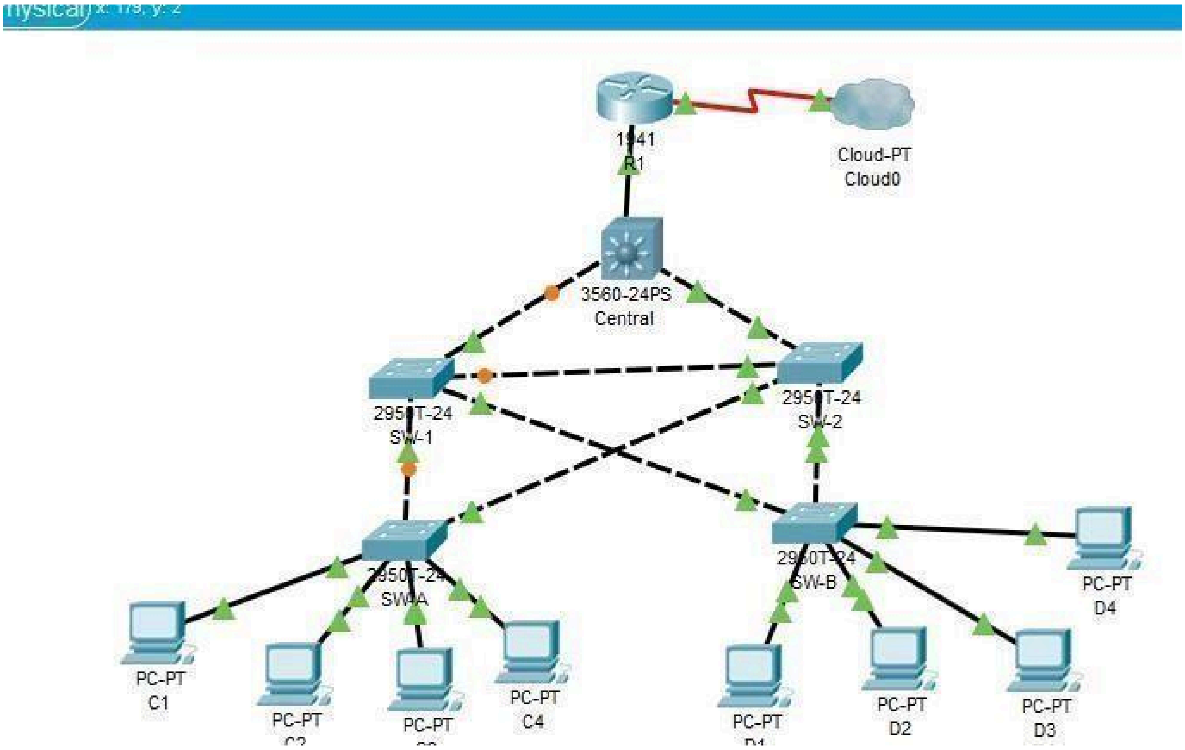
Service ● On ○ Off

	Time	HostName	Message
1	01.10.2014 10:32:29.036 AM	192.168.1.1	%IPS-4-SIGNATURE: Sig:2004 Subsig:...
2	01.10.2014 10:32:35.049 AM	192.168.1.1	%IPS-4-SIGNATURE: Sig:2004 Subsig:...
3	01.10.2014 10:32:41.070 AM	192.168.1.1	%IPS-4-SIGNATURE: Sig:2004 Subsig:...
4	01.10.2014 10:32:47.089 AM	192.168.1.1	%IPS-4-SIGNATURE: Sig:2004 Subsig:...

PRACTICAL 7

Aim: Packet Tracer - Layer 2 Security

Topology



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.

Current root is SW-1.

Step 2: Assign Central as the primary root bridge

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

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

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

Step 4: Verify the spanning-tree configuration

Central# **show spanning-tree**

```
        LAN0001
    Spanning tree enabled protocol
    ieee RootIDPriority    24577
             Address       00D0.D31C.634C

             This bridge is the root

                               MaxAge   20sec    ForwardDelay   15
             Hello Time   2 sec   sec
```

Part 2: Protect AgainstSTPAttacks

Step 1: Enable PortFast on allaccessports

```
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.

```
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
```

Step 3: Enable root guard

```
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.

```
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)# switchportmodeaccess

SW-B(config-if-range)#switchportport-security
SW-B(config-if-range)# switchport port-security maximum 2
```

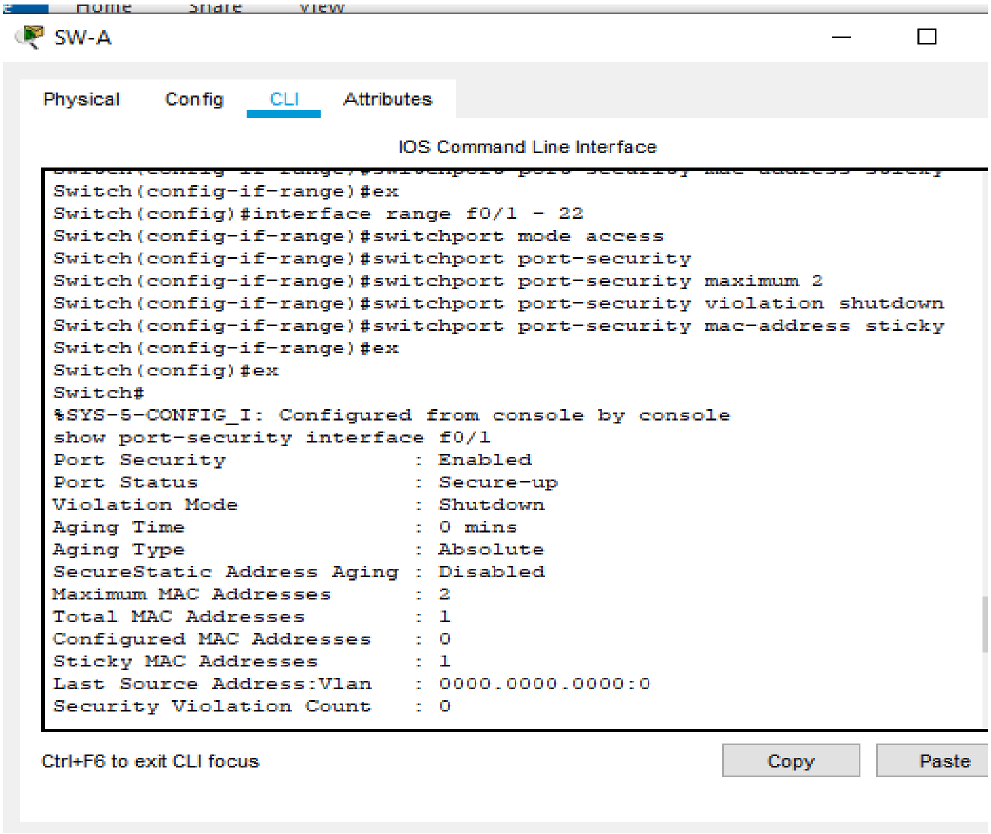
SW-B(config-if-range)# **switchport port-security violation shutdown**

SW-B(config-if-range)# **switchport port-security mac-address sticky**

Step 2: Verify port security.

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

SW-A# **show port-security interface f0/1**

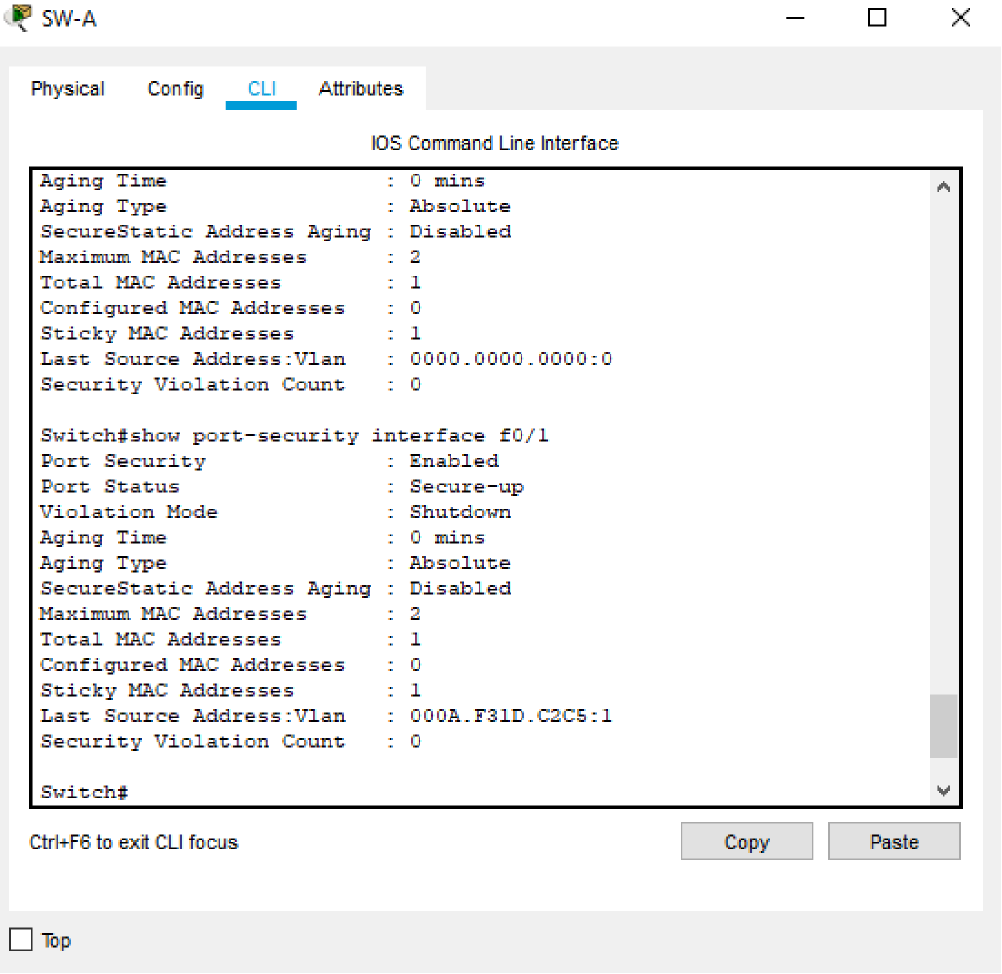


```
Switch(config-if-range)#switchport port-security mac-address sticky
Switch(config-if-range)#ex
Switch(config)#interface range f0/1 - 22
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport port-security
Switch(config-if-range)#switchport port-security maximum 2
Switch(config-if-range)#switchport port-security violation shutdown
Switch(config-if-range)#switchport port-security mac-address sticky
Switch(config-if-range)#ex
Switch(config)#ex
Switch#
%SYS-5-CONFIG_I: Configured from console by console
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        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
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 addressfor**C1**.

c. **After ping C1toC2:**

SW-A# **show port-security interface f0/1**



```
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 2
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0

Switch#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        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 000A.F31D.C2C5:1
Security Violation Count   : 0

Switch#
```

Step 3: Disable unused ports.

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

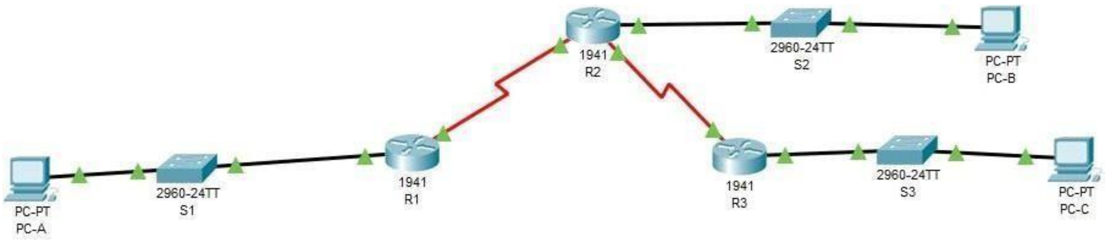
SW-B(config)# **interface range f0/5 - 22**
SW-B(config-if-range)# **shutdown**

Step 4: Check results.

PRACTICAL 8

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

Topology



Addressing Table

Device	Interfaces	IP Address	Subnet Mask	Default
				Gateway
R1	G0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	30.1.1.2	255.255.255.252	N/A
R2	S0/0/0	30.1.1.1	255.255.255.252	N/A
	S0/0/1	40.2.2.1	255.255.255.252	N/A
	G0/0	192.168.2.1	255.255.255.0	N/A
R3	G0/0	192.168.3.1	255.255.255.0	N/A
	S0/0/1	40.2.2.2	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

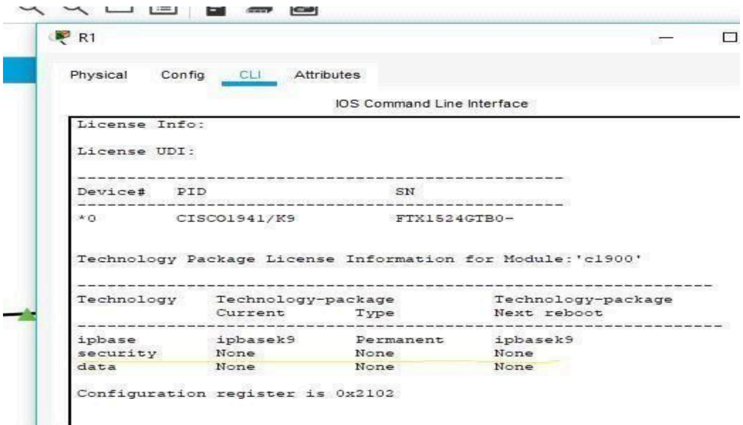
Part 1: Configure IPsec Parameters on R1

Step 1: Test connectivity.

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

Step 2: Enable the Security Technology package.

R1# **show version**



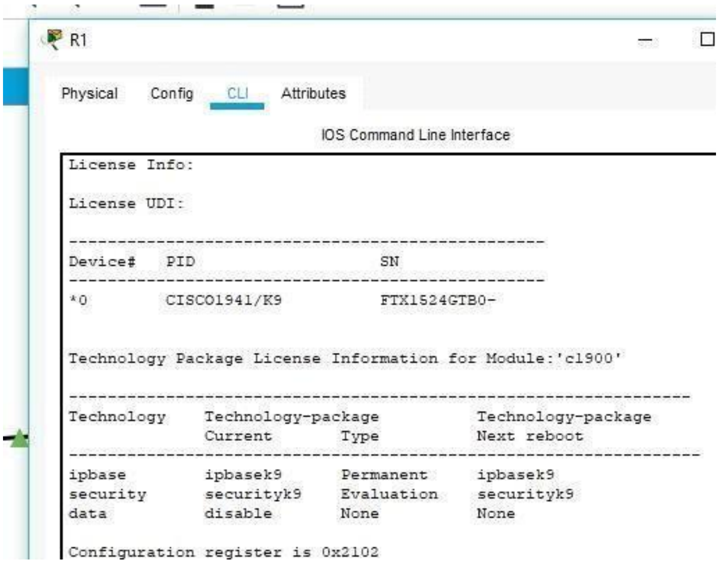
```
License Info:

License UDI:

-------------------------------------------------
Device#   PID                    SN
-------------------------------------------------
*0        CISCO1941/K9           FTX1524GTB0-


Technology Package License Information for Module:'c1900'

-----------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
------------------------------------------------------------------
ipbase        ipbasek9      Permanent      ipbasek9
security      None          None           None
data          None          None           None

Configuration register is 0x2102
```

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

R1(config)# **do write**

R1(config)# **do**

reload R1# **show**



```
License Info:

License UDI:

-------------------------------------------------
Device#   PID                    SN
-------------------------------------------------
*0        CISCO1941/K9           FTX1524GTB0-


Technology Package License Information for Module:'c1900'

----------------------------------------------------------------
Technology    Technology-package          Technology-package
              Current       Type          Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent     ipbasek9
security      securityk9    Evaluation    securityk9
data          disable       None          None

Configuration register is 0x2102
```

version

Step 3: Identify interesting traffic on R1.

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

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 40.2.2.2

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

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

```
R1(config)# crypto map VPN-MAP 10 ipsec-isakmp
```

```
R1(config-crypto-map)# description VPN connection to R3

R1(config-crypto-map)# set peer 40.2.2.2

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.

```
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

Technologypackage.

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

a. 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.

```
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

```
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
```

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

```
R3(config)# crypto ipsec transform-set VPN-SET esp-aes
esp-sha-hmac R3(config)# crypto map VPN-MAP 10
ipsec-isakmp R3(config-crypto-map)# description VPN
connection to R1
```

R3(config-crypto-map)# **set peer 30.1.1.2**

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

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