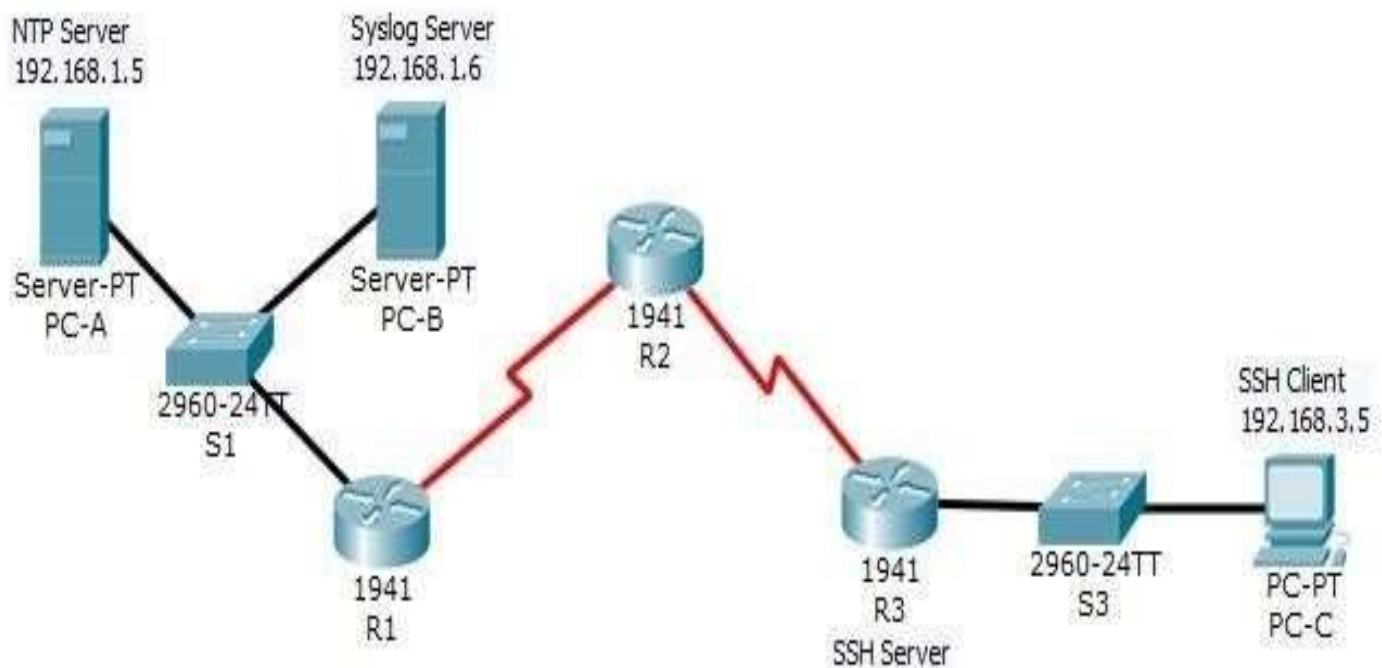


Practical 1: Configure Routers for Syslog, NTP and SSH operation

Topology:



Addressing Table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	gig0/0	192.168.1.1	255.255.255.0	N/A
	Se0/1/0	10.1.1.1	255.255.255.252	N/A
R2	Se0/1/0	10.1.1.2	255.255.255.252	N/A
	Se0/1/1	10.2.2.2	255.255.255.252	N/A
R3	gig0/0	192.168.3.1	255.255.255.0	N/A
	Se0/1/0	10.2.2.1	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

Objectives:

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

■ Configure Router with password

Step 1: Configure password for vty lines

Execute Command on all routers

```
R(config) # line vty 0 4
```

```
R(config-line) #password vtyp55
```

```
R(config-line) #login
```

Step 2: Configure secret on router

Execute Command on all routers

```
R(config) # enable secret enpa55
```

Step 3: Configure OSPF on routers

```
R1(config) #router ospf 1
```

```
R1(config-router) #network 192.168.1.0 0.0.0.255 area 0
```

```
R1(config-router) #network 10.1.1.0 0.0.0.3 area 0
```

```
R2(config) #router ospf 1
```

```
R2(config-router) #network 10.1.1.0 0.0.0.3 area 0
```

```
R2(config-router) #network 10.2.2.0 0.0.0.3 area 0
```

```
R3(config) #router ospf 1
```

```
R3(config-router) #network 192.168.3.0 0.0.0.255 area 0
```

```
R3(config-router) #network 10.2.2.0 0.0.0.3 area 0
```

Step 4: Test Connectivity

```
PC-A > ping 192.168.3.5
```

Successful

```
PC-B > ping 192.168.3.5
```

Successful

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. Configure an MD5 key on the serial interfaces on R1, R2 and R3. Use the password MD5pa55 for key 1.

```
R1(config)# interface s0/1/0
```

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

```
R2(config)# interface s0/1/0
```

```
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55
```

```
R2(config-if)# interface s0/1/1
```

```
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55
```

```
R3(config)# interface s0/1/0
```

```
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.

Output should be shown in all the routers :

```
R# show ip ospf interface
```

```
Message-digest Authentication Enabled
```

```
Youngest key ID is 1
```

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
```

Verify that the hardware Clock was Updated

```
R# 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.

Execute commands on all 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

R# 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 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 sshpa55.

```
R3(config)# username SSHadmin privilege 15 secret sshpa55
```

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.

```
R3# show ip ssh
```

```
SSH enabled-version 1.99
```

```
Authentication time out: 120 secs; Authentication retries : 3
```

```
R#
```

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
```

Verify the SSH configuration

```
R3# show ip ssh
```

```
SSH enabled-version 2.0
```


Authentication time out: 90 secs; Authentication retries : 2

R#

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 shpa55.

```
PC> ssh -l SSHadmin 192.168.3.1
```

Password: sshpa55

Step 10: Connect to R3 using SSH on R2.

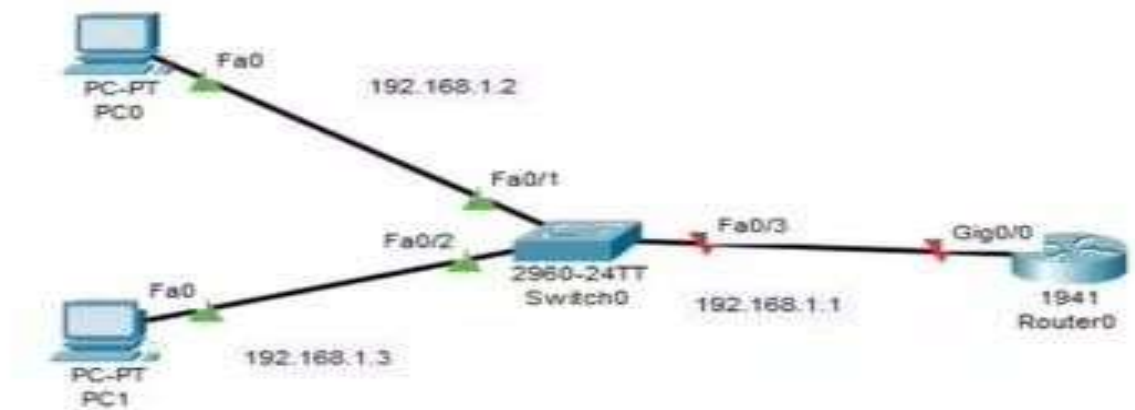
To troubleshoot and maintain R3, the administrator at the ISP must use SSH to access the router CLI. From the CLI of R2, enter the command to connect to R3 via SSH version 2 using the SSHadmin user account. When prompted for the password, enter the password configured for the administrator: ciscosshpa55.

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

Password: sshpa55

Practical 2: Configure AAA Authentication on Cisco routers

Topology:



Addressing Table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	gig0/0	192.168.1.1	255.255.255.0	N/A
PC0	NIC	192.168.1.2	255.255.255.0	192.168.1.1
PC1	NIC	192.168.1.3	255.255.255.0	192.168.1.1

Objectives:

- Configure a local user account on R1 and configure authentication on the console and vty lines using local AAA.
- Verify local AAA authentication from the R1 console and the PC0 client and PC1 Client.

■ Configure Router:

Step 1: Configure password for vty lines

```
R1(config) # line vty 0 4
```

```
R1(config-line) #password vtyp55
```

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

Step 2: Configure secret on router

```
R1(config) # enable secret enpa55
```

Step 3: Configure OSPF on routers

```
R1(config) #router ospf 1
```

```
R1(config-router) #network 192.168.1.0 0.0.0.255 area 0
```

Step 4: Configure OSPF MD5 authentication for all router in area 0

```
R1(config) #router ospf 1
```

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

Step 5: Configure MD5 key for all routers in area 0

```
R1(config)# int gig0/0
```

```
R1(config-if)# ip ospf message-digest-key 1 md5 pa55
```

Step 6: Verify configurations.

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

b. Verify end-to-end connectivity.

Output should be shown in all the routers :

```
R1# show ip ospf interface
```

```
Message-digest Authentication Enabled
```

```
Youngest key ID is 1
```

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

Step 1: Test Connectivity

PC0 > ping 192.168.1.3

Successful

PC1 > ping 192.168.1.2

Successful

Step 2: Configure Local username on R1

R1(config)# username admin secret adminpa55

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

User Access Verification

Username: admin

Password: adminpa55

R1>

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.

```
PC0> ssh -l Admin 192.168.1.1
```

```
Password: adminpa55
```

```
R1>
```

```
PC1> ssh -l Admin 192.168.1.1
```

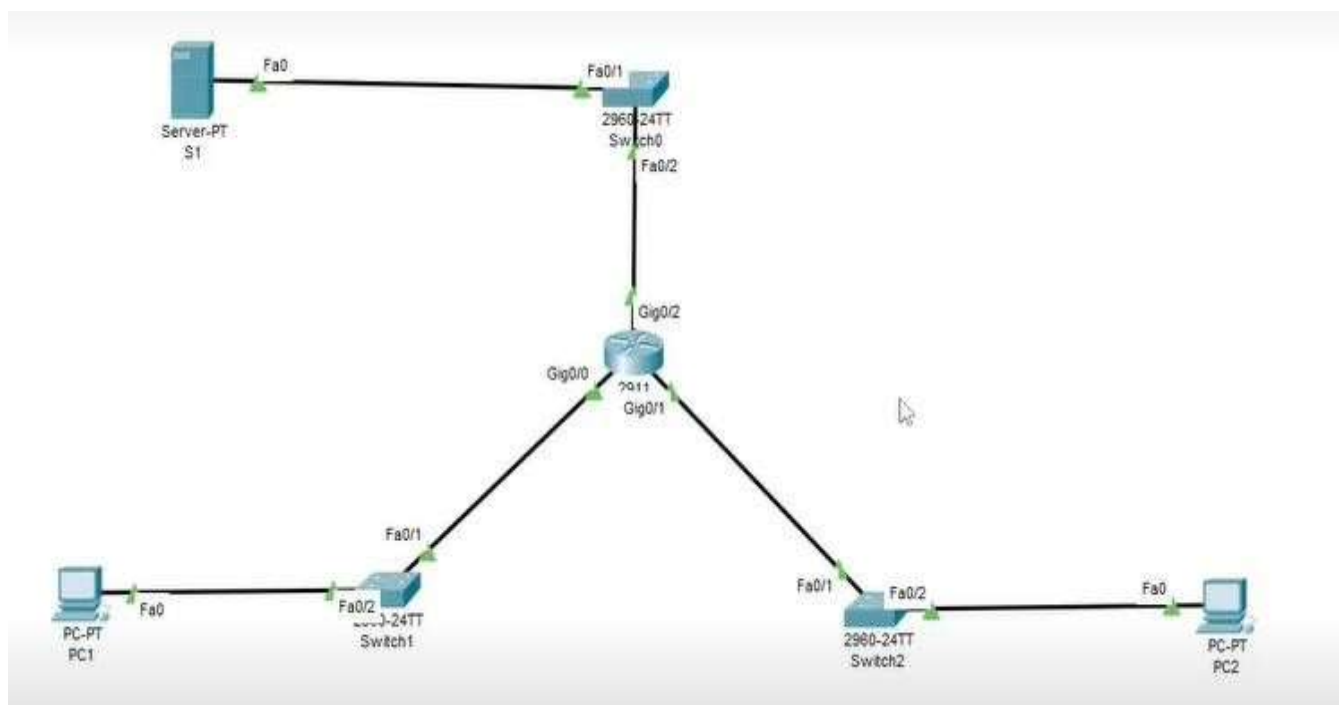
```
Password: adminpa55
```

```
R1>
```

Practical 3: Configuring Extended ACLs

A]

Topology:



Addressing Table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	gig0/0	172.22.34.65	255.255.255.224	N/A
	gig0/1	172.22.34.97	255.255.255.240	N/A
	gig0/2	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:

- **Configure, Apply and Verify an Extended Numbered ACL**
- **Configure, Apply and Verify an Extended Named ACL**

Scenario:

- PC1 Should be allowed only FTP access
- PC2 Should be allowed only web access
- Both PCs must ping server but not each other's

■ Configure Router:

Step 1: Configure password for vty lines

```
R1(config) # line vty 0 4
```

```
R1(config-line) #password vtyp55
```

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

Step 2: Configure secret on router

```
R1(config) # enable secret enp55
```

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

Step 1: Configure an ACL to permit FTP and ICMP. (Use Router 2911)

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

```
172.22.34.62 eq ftp
```

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

```
172.22.34.62
```

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

```
R1(config)# int gig 0/0
```

```
R1(config-if)# ip access-group 100 in
```

Step 3: Verify the ACL implementation.

a. Ping from PC1 to Server.

```
PC1> ping 172.22.34.62
```

(Successful)

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

```
PC1> ftp 172.22.34.62
```

c. Exit the FTP service of the Server.

```
ftp> quit
```

d. Ping from PC1 to PC2.

```
PC1> ping 172.22.34.98
```

(Unsuccessful) destination host unreachable

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

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

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

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

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

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

```
R1(config)# int gig0/1
```

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


Step 3: Verify the ACL implementation.

a. Ping from PC2 to Server.

PC2> ping 172.22.34.62

(Successful)

b. FTP from PC2 to Server

PC2> ftp 172.22.34.62

(Unsuccessful)

c. Open the web browser on PC2.

URL -> http://172.22.34.62

(Successful)

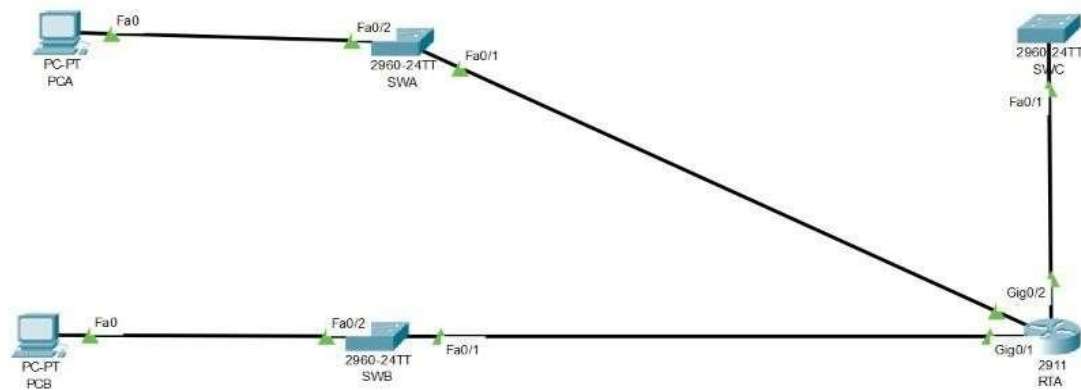
d. Ping from PC2 to PC1.

PC> ping 172.22.34.66

(Unsuccessful)

B]

Topology:



Addressing Table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
RTA	gig0/0	10.101.117.49	255.255.255.248	N/A
	gig0/1	10.101.117.33	255.255.255.240	N/A
	gig0/2	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:

- **Configure, Apply and Verify an Extended Numbered ACL**

Scenario:

- Device on one LAN are allowed to remotely access device in another LAN using SSH protocol
- Besides ICMP all traffic from other network is denied

■ Configure Switch and Router:

Step 1: Configure the IP address on switch

```
SWA(config)# int vlan 1
```

```
SWA(config-if)# ip address 10.101.117.50 255.255.255.248
```

```
SWA(config-if)# no shut
```

```
SWA(config-if)# ip default-gateway 10.101.117.49
```

```
SWB(config)# int vlan 1
```

```
SWB(config-if)# ip address 10.101.117.34 255.255.255.240
```

```
SWB(config-if)# no shut
```

```
SWB(config-if)# ip default-gateway 10.101.117.33
```

```
SWC(config)# int vlan 1
```

```
SWC(config-if)# ip address 10.101.117.2 255.255.255.224
```

```
SWC(config-if)# no shut
```

```
SWC(config-if)# ip default-gateway 10.101.117.1
```

Step 2: Configure the secret on router and switch

```
RTA/SW(config)# enable secret enpa55
```

Step 3: Configure the console password on router and switch

```
RTA/SW(config)# line console 0
```

```
RTA/SW(config)# password tyit
```

```
RTA/SW(config)# login
```

Step 4: Test connectivity

Ping from PCA to PC-B.

PCA>ping 10.101.117.35

(Successful)

Ping from PCA to SWC.

PCA>ping 10.101.117.2

(Successful)

Ping from PCB to SWC.

PCB>ping 10.101.117.2

(Successful)

Part 1: Configure Switch and Router to support SSH Connection

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

RTA/SW(config)# ip domain-name ccnasecurity.com

Step 2: Configure users to login to SSH

RTA/SW(config)# username admin secret adminpa55

Step 3: Configure incoming vty lines

RTA/SW(config)# line vty 0 4

RTA/SW(config-line)# login local

RTA/SW(config)# crypto key generate rsa

How many bits in the modulus [512]: 1024

Step 4: Verify the SSH Connection

PCA> ssh -l Admin 10.101.117.34

Password: adminpa55

SWB>

PCA> ssh -l Admin 10.101.117.2

Password: adminpa55

SWC>

PCB> ssh -l Admin 10.101.117.50

Password: adminpa55

SWA>

PCB> ssh -l Admin 10.101.117.2

Password: adminpa55

SWC>

SWC> ssh -l Admin 10.101.117.50

Password: adminpa55

SWA>

SWC> ssh -l Admin 10.101.117.34

Password: adminpa55

SWB>

SWB> exit

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

Step 1: Configure the extended ACL.

RTA(config)# access-list 199 permit tcp 10.101.117.32 0.0.0.15 10.101.117.0
0.0.0.31 eq 22

RTA(config)# access-list 199 permit icmp any any

Step 2: Apply the extended ACL.

```
RTA(config)# int gig0/2
```

```
RTA(config-if)# 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.

```
PCB> ping 10.101.117.51
```

(Successful)

```
PCB> ping 10.101.117.2
```

(Successful)

b. SSH from PCB to SWC.

```
PCB> ssh -l Admin 10.101.117.2
```

```
Password:adminpa55
```

```
SWC>
```

c. Exit the SSH session to SWC.

```
SWC>exit
```

d. Ping from PCA to all of the other IP addresses in the network.

```
PCA> ping 10.101.117.35
```

(Successful)

```
PCA> ping 10.101.117.2
```

(Successful)

e. SSH from PCA to SWC

```
PCA> ssh -l Admin 10.101.117.2
```

Connection timed out. Remote host not responding

f. SSH from PCA to SWB.

PCA> ssh -l Admin 10.101.117.34

Password: adminpa55

SWB>

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

SWB# ssh -l Admin 10.101.117.2

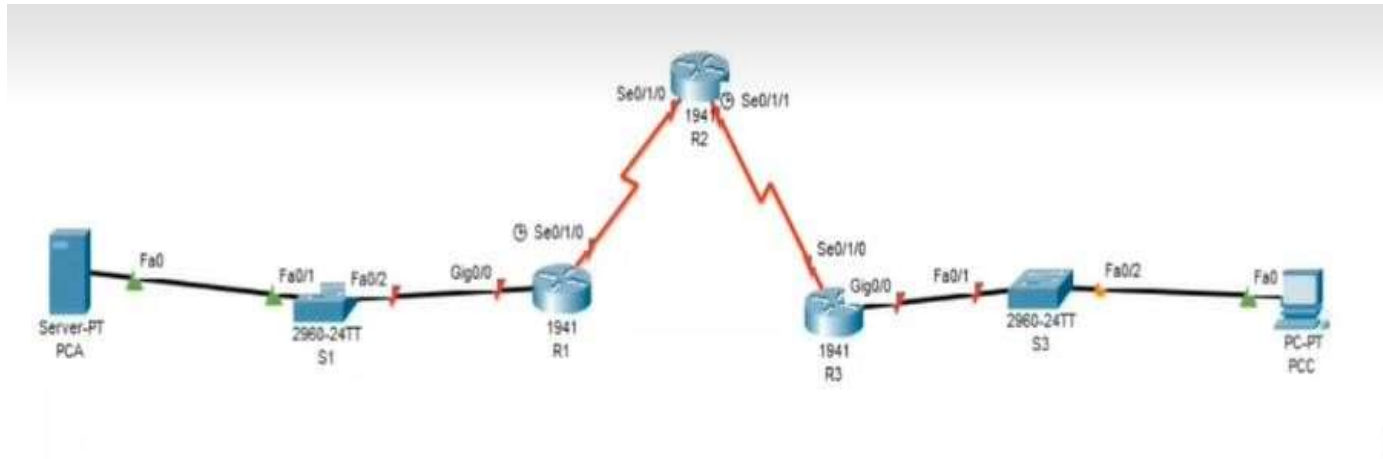
Password: adminpa55

SWC>

Practical 4: Configure IP ACLs to Mitigate Attacks

A]

Topology:



Addressing Table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	gig0/0	192.168.1.1	255.255.255.0	N/A
	Se0/1/0 (DCE)	10.1.1.1	255.255.255.252	N/A
R2	Se0/1/0	10.1.1.2	255.255.255.252	N/A
	Se0/1/1 (DCE)	10.2.2.2	255.255.255.252	N/A
	Lo0	192.168.2.1	255.255.255.0	N/A
R3	gig0/0	192.168.3.1	255.255.255.0	N/A
	Se0/1/0	10.2.2.1	255.255.255.252	N/A
PC-A	Fa0	192.168.1.3	255.255.255.0	192.168.1.1
PC-C	Fa0	192.168.3.3	255.255.255.0	192.168.3.1

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.

■ Configure Router:

Step 1: Configure secret on router

```
R(config) # enable secret enpa55
```

Step 2: Configure console password on router

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

```
R(config-line) #password conpa55
```

```
R(config-line) #login
```

Step 3: Configure SSH login on router

Execute command on all routers

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

```
R(config)# username admin secret adminpa55
```

```
R(config)# line vty 0 4
```

```
R(config-line)# login local
```

```
R(config-line)# crypto key generate rsa
```

```
How many bits in the modulus [512]: 1024
```

Step 4: Configure loop back address on Router 2

```
R2(config)#int loopback 0
```

```
R2(config-if)#ip address 192.168.2.1 255.255.255.0
```

```
R2(config-if)# no shut
```

Step 5: Configure static routing on routers

Execute command on all routers

```
R1(config)#ip route 192.168.3.0 255.255.255.0 10.1.1.2
```

```
R1(config)#ip route 10.2.2.0 255.255.255.252 10.1.1.2
R1(config)#ip route 192.168.2.0 255.255.255.0 10.1.1.2
```

```
R2(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.1
```

```
R3(config)#ip route 192.168.1.0 255.255.255.0 10.2.2.2
R3(config)#ip route 192.168.2.0 255.255.255.0 10.2.2.2
R3(config)#ip route 10.1.1.0 255.255.255.0 10.2.2.2
```

Part 2: Verify Basic Network Connectivity

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

```
PCA> ping 192.168.3.3
(Successful)
PCA> ping 192.168.2.1
(Successful)
PCA> ssh -l admin 192.168.2.1
Password: adminpa55
R2>exit
```

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

```
PCC> ping 192.168.1.3
(Successful)
PCC> ping 192.168.2.1
(Successful)
PCC> ssh -l admin 192.168.2.1
```

Password: adminpa55

R2>exit

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

Close the browser when done.

Desktop->Web Browser->192.168.1.3

(Successful)

Part 3: Secure Access to Routers

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

Execute command on all routers

R(config)# access-list 10 permit host 192.168.3.3

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

Execute command on all routers

R(config)# line vty 0 4

R(config-line)# access-class 10 in

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

PCC> ssh -l admin 192.168.2.1

Password: adminpa55

R2>exit

Step 4: Verify denial from PC-A.

PCA> ssh -l admin 192.168.2.1

Connection refused by remote host

Part 4: Create a Numbered IP ACL 120 on R1

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 in Services tab.

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

```
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

```
R1(config)# int se0/1/0
```

```
R1(config-if)# ip access-group 120 in
```

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

Desktop->Web Browser->192.168.1.3

(Unsuccessful) Request timed out

Part 5: Modify an Existing ACL on R1

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

```
PCA> ping 192.168.2.1
```

(Unsuccessful) Request timed out

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

```
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.

```
PCA> ping 192.168.2.1 (Successful)
```

Part 6: Create a Numbered IP ACL 110 on R3

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

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

Step 2: Apply the ACL to interface

```
R3(config)# int gig0/1
```

```
R3(config-if)# ip access-group 110 in
```

Part 7: Create a Numbered IP ACL 100 on R3

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

```
R3(config)# access-list 100 permit tcp 10.0.0.0 0.255.255.255 host 192.168.3.3 eq 22
```

```
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

```
R3(config)# interface se0/1/0
```

```
R3(config-if)# ip access-group 100 in
```

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

```
PCC> ping 192.168.1.3
```

(Unsuccessful)

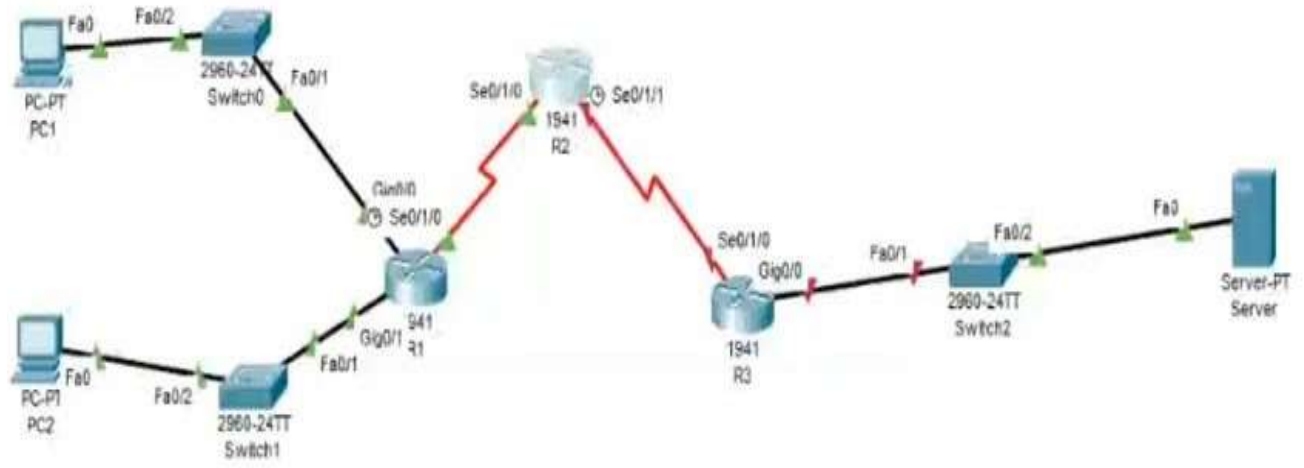
```
PCC> ssh -l admin 192.168.2.1
```

Password: adminpa55

```
R2>exit
```

B]

Topology:



Addressing Table:

Device	Interface	IPv6 Address/Prefix	Default Gateway
PC1	NIC	2001:DB8:1:10::10/64	FE80::1
PC2	NIC	2001:DB8:1:11:11/64	FE80::1
R1	gig0/0	2001:DB8:1:10::1/64	FE80::1
	se0/1/0	2001:DB8:1:1::1/64	FE80::1
	gig0/1	2001:DB8:1:11::1/64	FE80::1
R3	se0/1/0	2001:DB8:1:1::2/64	FE80::2
	se0/1/1	2001:DB8:1:2::2/64	FE80::2
R3	gig0/0	2001:DB8:1:30::1/64	FE80::3
	se0/1/0	2001:DB8:1:2::1/64	FE80::3
Server	NIC	2001:DB8:1:30::30/64	FE80::3

Objective:

- Configure, Apply, and Verify an IPv6 ACL
- Configure, Apply, and Verify a Second IPv6 ACL

■ Configure Router:

Step 1: Configure secret on router

Execute command on all routers

```
R(config)# enable secret enpa55
```

Step 2: Assign static ipv6 address

```
R1(config)# int gig0/0
```

```
R1(config-if)# ipv6 address 2001:DB8:1:10::1/64
```

```
R1(config-if)# ipv6 address FE80::1 link-local
```

```
R1(config-if)# no shut
```

```
R1(config)# int gig0/1
```

```
R1(config-if)# ipv6 address 2001:DB8:1:11::1/64
```

```
R1(config-if)# ipv6 address FE80::1 link-local
```

```
R1(config-if)# no shut
```

```
R1(config)# int se0/1/0
```

```
R1(config-if)# ipv6 address 2001:DB8:1:1::1/64
```

```
R1(config-if)# ipv6 address FE80::1 link-local
```

```
R1(config-if)# no shut
```

```
R2(config)# int se0/1/0
```

```
R2(config-if)# ipv6 address 2001:DB8:1:1::2/64
```

```
R2(config-if)# ipv6 address FE80::2 link-local
```

```
R2(config-if)# no shut
```

```
R2(config)# int se0/1/1
```



```
R2(config-if)# ipv6 address 2001:DB8:1:2::2/64
```

```
R2(config-if)# ipv6 address FE80::2 link-local
```

```
R2(config-if)# no shut
```

```
R3(config)# int gig0/0
```

```
R3(config-if)# ipv6 address 2001:DB8:1:30::1/64
```

```
R3(config-if)# ipv6 address FE80::3 link-local
```

```
R3(config-if)# no shut
```

```
R3(config)# int se0/1/0
```

```
R3(config-if)# ipv6 address 2001:DB8:1:2::1/64
```

```
R3(config-if)# ipv6 address FE80::3 link-local
```

```
R3(config-if)# no shut
```

Step 3: Enable IPv6 routing

```
R1(config)# ipv6 unicast-routing
```

```
R1(config)# ipv6 route 2001:DB8:1:2::0/64 2001:DB8:1:1::2
```

```
R1(config)# ipv6 route 2001:DB8:1:30::0/64 2001:DB8:1:1::2
```

```
R2(config)# ipv6 unicast-routing
```

```
R2(config)# ipv6 route 2001:DB8:1:10::0/64 2001:DB8:1:1::1
```

```
R2(config)# ipv6 route 2001:DB8:1:11::0/64 2001:DB8:1:1::1
```

```
R2(config)# ipv6 route 2001:DB8:1:30::0/64 2001:DB8:1:2::1
```

```
R3(config)# ipv6 unicast-routing
```

```
R3(config)# ipv6 route 2001:DB8:1:10::0/64 2001:DB8:1:2::2
```

```
R3(config)# ipv6 route 2001:DB8:1:11::0/64 2001:DB8:1:2::2
```

```
R3(config)# ipv6 route 2001:DB8:1:1::0/64 2001:DB8:1:2::2
```

Step 4: Verify connectivity

```
PC1> ping 2001:DB8:1:30::30
```

(Successful)

```
PC2> ping 2001:DB8:1:30::30
```

(Successful)

Part 2: Configure, Apply, and Verify an IPv6 ACL

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

```
R1(config)# ipv6 access-list BLOCK_HTTP
```

```
R1(config-ipv6-acl)# deny tcp any host 2001:DB8:1:30::30 eq www
```

```
R1(config-ipv6-acl)# deny tcp any host 2001:DB8:1:30::30 eq 443
```

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

```
R1(config-ipv6-acl)# exit
```

Step 2: Apply the ACL to the correct interface.

```
R1(config)# int gig0/1
```

```
R1(config-if)# ipv6 traffic-filter BLOCK_HTTP in
```

Step 3: Verify the ACL implementation

Open a web browser to the PC1 to display the web page.

```
Desktop->Web Browser->http://2001:DB8:1:30::30
```

(Successful)

```
Desktop->Web Browser->https://2001:DB8:1:30::30
```

(Successful)

Open a web browser to the PC2 to display the web page.

```
Desktop->Web Browser->http://2001:DB8:1:30::30
```

(Unsuccessful) – Request Timeout

Desktop->Web Browser->https://2001:DB8:1:30::30

(Unsuccessful) – Request Timeout

PC2> ping 2001:DB8:1:30::30

(Successful)

Part 3: Configure, Apply, and Verify a Second IPv6 ACL

Step 1: Create an access list to block ICMP.

```
R3(config)# ipv6 access-list BLOCK_ICMP
```

```
R3(config-ipv6-acl)# deny icmp any any
```

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

```
R3(config-ipv6-acl)# exit
```

Step 2: Apply the ACL to the correct interface.

```
R3(config)# int gig0/0
```

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

Step 3: Verify that the proper access list functions.

PC2> ping 2001:DB8:1:30::30

(Unsuccessful) - Destination host unreachable

PC1> ping 2001:DB8:1:30::30

(Unsuccessful) - Destination host unreachable.

Open a web browser to the PC1 to display the web page.

Desktop->Web Browser-> http://2001:DB8:1:30::30

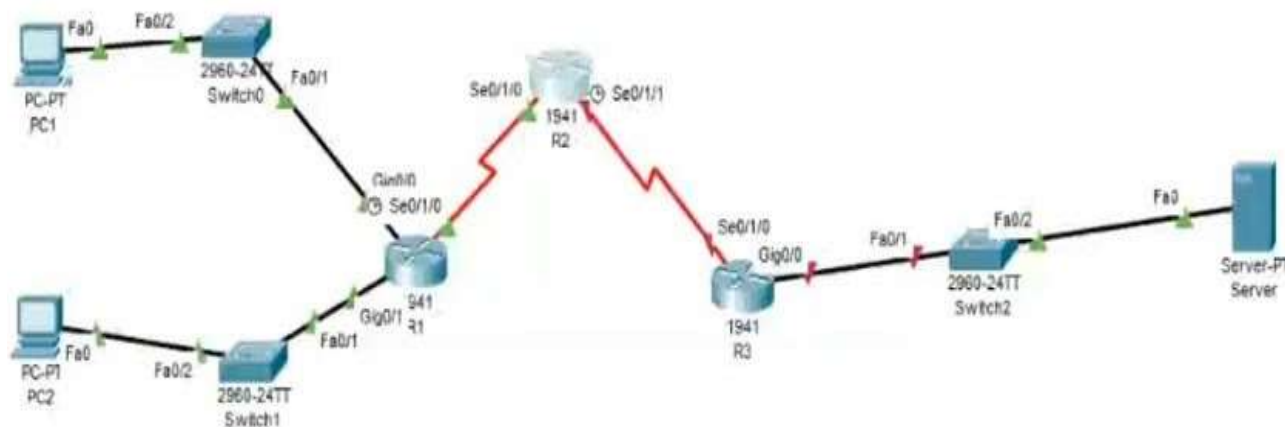
(Successful)

Desktop->Web Browser->https://2001:DB8:1:30::30

(Successful)

PRACTICAL 5

Topology:



Addressing Table:

Device	Interface	IPv6 Address/Prefix	Default Gateway
PC1	NIC	2001:DB8:1:10::10/64	FE80::1
PC2	NIC	2001:DB8:1:11:11/64	FE80::1
R1	gig0/0	2001:DB8:1:10::1/64	FE80::1
	se0/1/0	2001:DB8:1:1::1/64	FE80::1
	gig0/1	2001:DB8:1:11::1/64	FE80::1
R3	se0/1/0	2001:DB8:1:1::2/64	FE80::2
	se0/1/1	2001:DB8:1:2::2/64	FE80::2
R3	gig0/0	2001:DB8:1:30::1/64	FE80::3
	se0/1/0	2001:DB8:1:2::1/64	FE80::3
Server	NIC	2001:DB8:1:30::30/64	FE80::3

Objectives:

- Configure, Apply, and Verify an IPv6 ACL
- Configure, Apply, and Verify a Second IPv6 ACL

Configuring Router R1

```
Router>en
Router#conf t
Router(config)#host R1
R1(config)#ipv6 unicast-routing
R1(config)#int g0/0
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address 2001:DB8:1:10::1/64
R1(config-if)#no shut

R1(config-if)#int g0/1
R1(config-if)#ipv6 enable
```

```
R1(config-if)#ipv6 address 2001:DB8:1:11::1/64
R1(config-if)#no shut
R1(config-if)#
R1(config-if)#int s0/1/0
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address 2001:DB8:1:1::1/64
R1(config-if)#no shut
R1(config-if)#
```

Configuring Router R2

```
Router>enable
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
?Bad filename
%Error parsing filename (Bad file number)
Router#en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#host R2
R2(config)#ipv6 unicast-routing
R2(config)#int s0/1/0
R2(config-if)#ipv6 enable
R2(config-if)#ipv6 address 2001:DB8:1:1::2/64
R2(config-if)#no shut

R2(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up
```

```
R2(config-if)#int s0/1/1
R2(config-if)#ipv6 enable
R2(config-if)#ipv6 address 2001:DB8:1:2::2/64
R2(config-if)#no shut
```

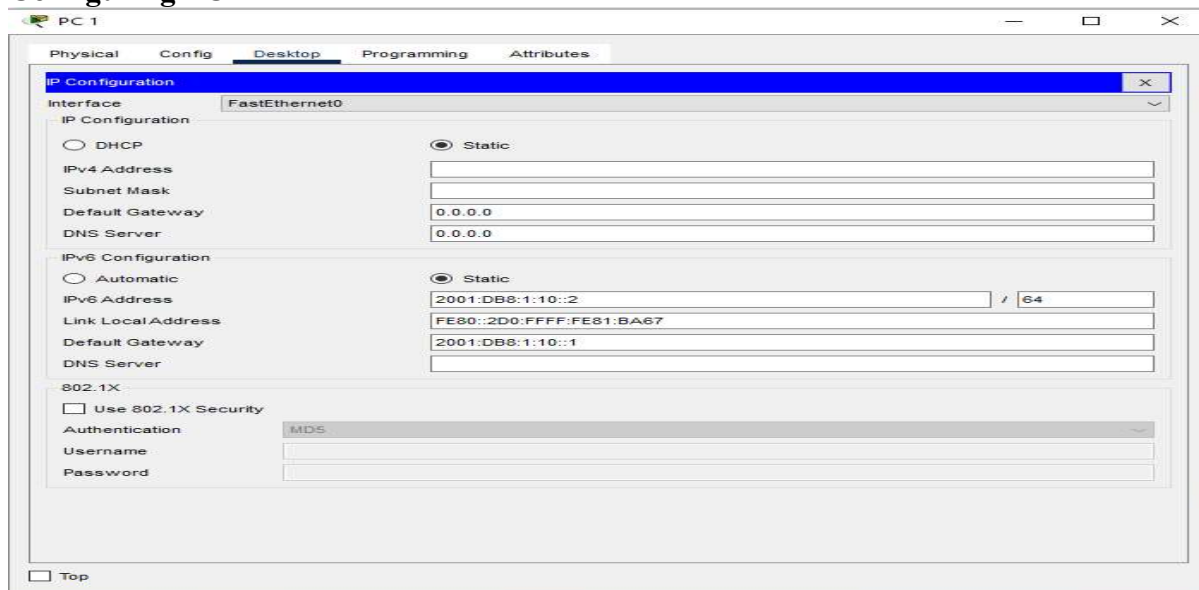
```
%LINK-5-CHANGED: Interface Serial0/1/1, changed state to down
R2(config-if)#
R2(config-if)#int s0/1/0
R2(config-if)#ipv6 enable
R2(config-if)#ipv6 address 2001:DB8:1:1::1/64
R2(config-if)#no shut
R2(config-if)#
```

Configuring Router R3

```
Router>en
Router#conf t
Router(config)#host R3
R3(config)#ipv6 unicast-routing
R3(config)#int g0/0
R3(config-if)#ipv6 enable
R3(config-if)#ipv6 address 2001:DB8:1:30::1/64
R3(config-if)#no shut
R3(config-if)#
```

```
R3(config-if)#int s0/1/1
R3(config-if)#ipv6 enable
R3(config-if)#ipv6 address 2001:DB8:1:2::1/64
R3(config-if)#no shut
R3(config-if)#
```

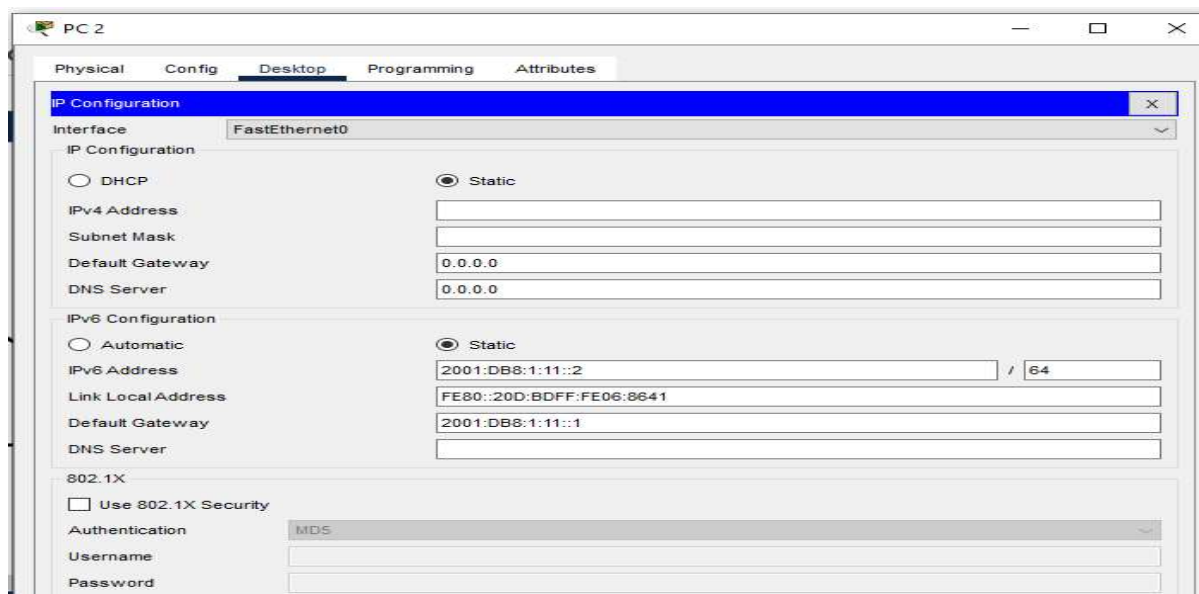
Configuring PC 1



The screenshot shows the 'PC 1' configuration window with the 'Desktop' tab selected. The 'IP Configuration' section is expanded, showing the 'FastEthernet0' interface. The 'IP Configuration' group has 'Static' selected. The 'IPv6 Configuration' group also has 'Static' selected. The IPv6 address is set to '2001:DB8:1:10::2' with a prefix length of '64'. The link local address is 'FE80::2D0:FFFF:FE81:BA67'. The default gateway is '2001:DB8:1:10::1'. The DNS server is empty. The '802.1X' section is collapsed.

Field	Value
Interface	FastEthernet0
IP Configuration	Static
IPv4 Address	
Subnet Mask	
Default Gateway	0.0.0.0
DNS Server	0.0.0.0
IPv6 Configuration	Static
IPv6 Address	2001:DB8:1:10::2 / 64
Link Local Address	FE80::2D0:FFFF:FE81:BA67
Default Gateway	2001:DB8:1:10::1
DNS Server	
802.1X	
Use 802.1X Security	<input type="checkbox"/>
Authentication	MDS
Username	
Password	

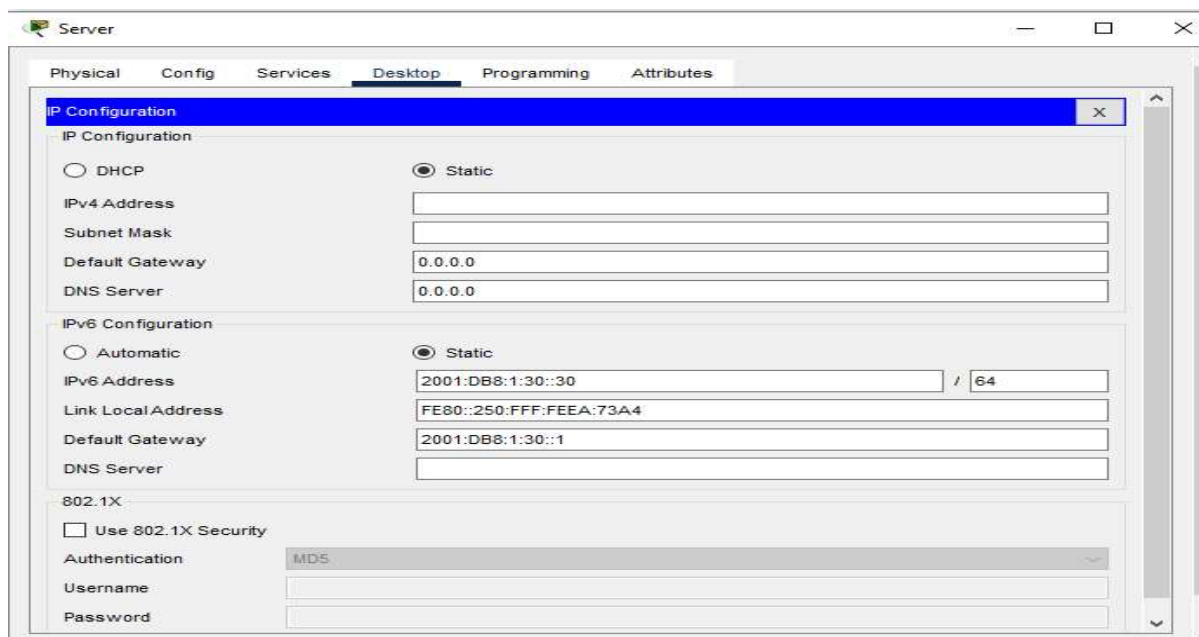
Configuring PC 2



The screenshot shows the 'PC 2' configuration window with the 'Desktop' tab selected. The 'IP Configuration' section is expanded, showing the 'FastEthernet0' interface. The 'IP Configuration' group has 'Static' selected. The 'IPv6 Configuration' group also has 'Static' selected. The IPv6 address is set to '2001:DB8:1:11::2' with a prefix length of '64'. The link local address is 'FE80::2D0:B0FF:FE06:8641'. The default gateway is '2001:DB8:1:11::1'. The DNS server is empty. The '802.1X' section is collapsed.

Field	Value
Interface	FastEthernet0
IP Configuration	Static
IPv4 Address	
Subnet Mask	
Default Gateway	0.0.0.0
DNS Server	0.0.0.0
IPv6 Configuration	Static
IPv6 Address	2001:DB8:1:11::2 / 64
Link Local Address	FE80::2D0:B0FF:FE06:8641
Default Gateway	2001:DB8:1:11::1
DNS Server	
802.1X	
Use 802.1X Security	<input type="checkbox"/>
Authentication	MDS
Username	
Password	

Configuring Server



Step 1: Configure secret and console password on router

R1>en

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#enable secret enpa55

R1(config)#line console 0

R1(config-line)#password conpa55

R1(config-line)#login

R1(config-line)#

R2>en

R2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R2(config)#enable secret enpa55

R2(config)#line console 0

R2(config-line)#password conpa55

R2(config-line)#login

R2(config-line)#

R3>en

R3#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R3(config)#enable secret enpa55

R3(config)#line console 0

R3(config-line)#password conpa55

R3(config-line)#login

R3(config-line)#

Step 2: Configure SSH login on router

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

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

R1(config)#username Admin secret Admin123

R1(config)#line vty 0 4

R1(config-line)#login local

R1(config-line)#crypto key generate rsa

The name for the keys will be: R1.ccnasecurity.com

Choose the size of the key modulus in the range of 360 to 4096 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]

R1(config)#

R2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R2(config)#ip domain-name ccnasecurity.com

R2(config)#username Admin secret Admin123

R2(config)#line vty 0 4

R2(config-line)#login local

R2(config-line)#crypto key generate rsa

The name for the keys will be: R2.ccnasecurity.com

Choose the size of the key modulus in the range of 360 to 4096 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]

R2(config)#

R3>en

Password:

R3#conf t

Enter configuration commands, one per line. End with CNTL/Z.

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

R3(config)#username Admin secret Admin123

R3(config)#line vty 0 4

R3(config-line)#login local

R3(config-line)#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 4096 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]

R3(config)#

Step 3: Configure loop back address on Router 2

R2(config)#int loopback 0

R2(config-if)#ip address 192.168.2.1 255.255.255.0

R2(config-if)# no shut

Step 3: Configure static routing on Routers

R1>en

Password:

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#ip route 192.168.3.0 255.255.255.0 10.1.1.2

R1(config)#ip route 10.2.2.0 255.255.255.252 10.1.1.2

R1(config)#ip route 192.168.2.0 255.255.255.0 10.1.1.2

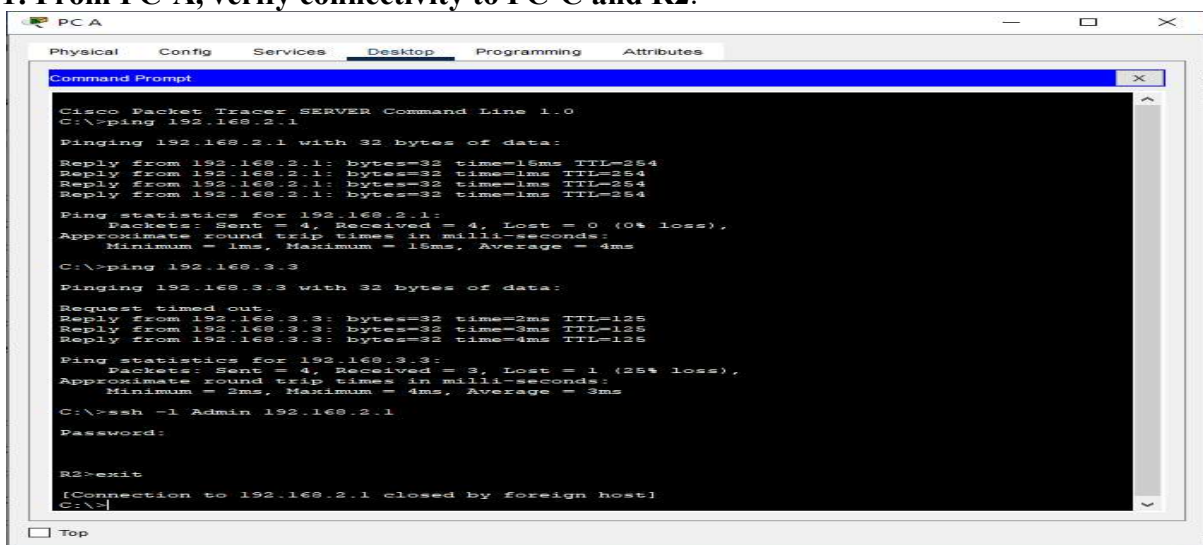
R1(config)#


```
R2(config-if)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.1
R2(config)#
R2#
```

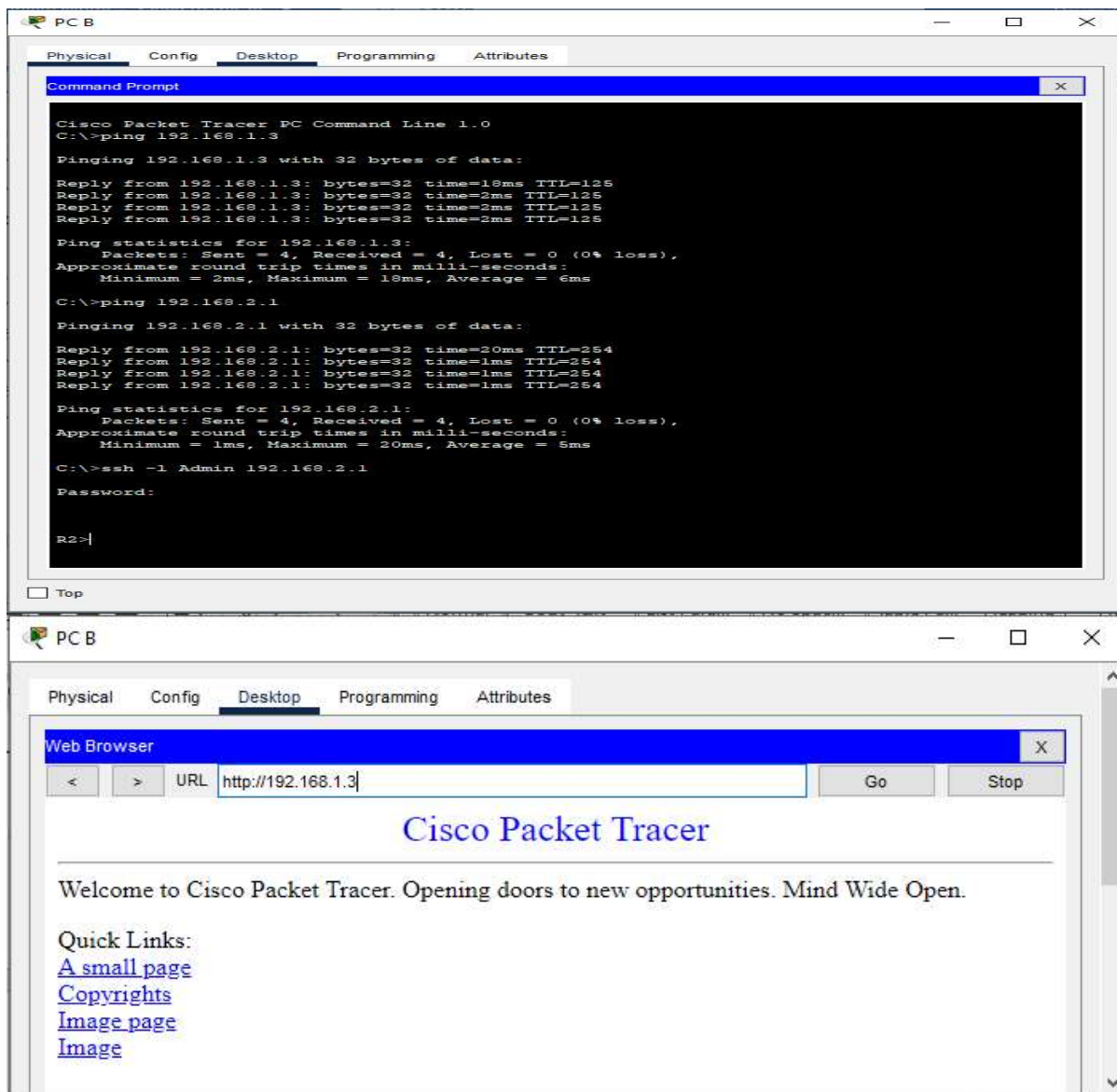
```
R3>en
R3#conf t
R3(config)#ip route 192.168.1.0 255.255.255.0 10.2.2.2
R3(config)#ip route 192.168.2.0 255.255.255.0 10.2.2.2
R3(config)#ip route 10.1.1.0 255.255.255.0 10.2.2.2
R3(config)#
R3#
```

Part 2: Verify Basic Network Connectivity

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



Step 2: From PC-B, verify connectivity to PC-A and R1.



Part 3: Secure Access to Routers

Step 1: Configure ACL 10 to block all remote access to routers except from PC B

```
R1>en
R1#conf t
R1(config)#access-list 10 permit host 192.168.3.3
```

```
R2>en
R2#conf t
R2(config)#access-list 10 permit host 192.168.3.3
```

```
R3>en
R3#conf t
R3(config)#access-list 10 permit host 192.168.3.3
```

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

```
R1(config)#line vty 0 4
R1(config-line)#access-class 10 in
```

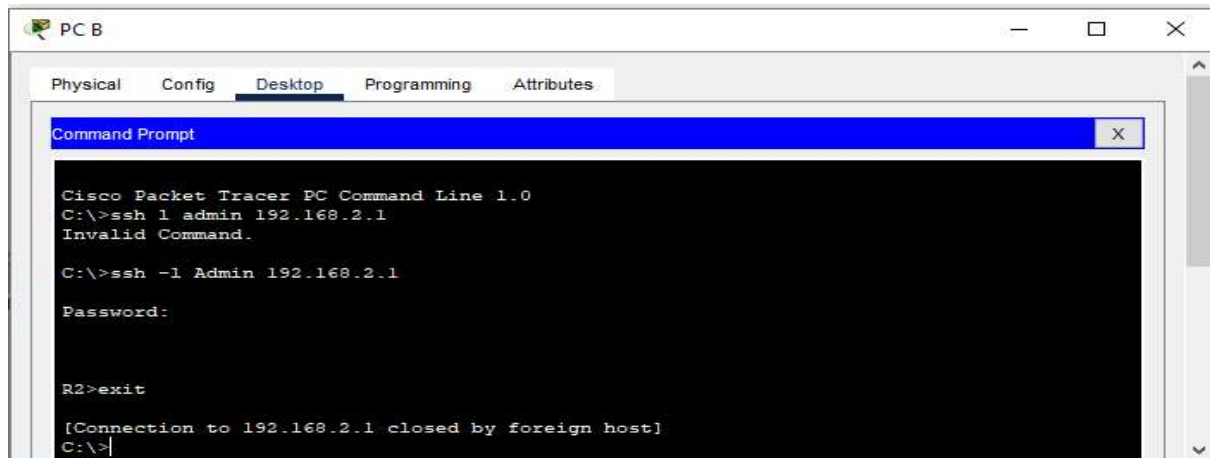
```
R2(config)#line vty 0 4
```

R2(config-line)#access-class 10 in

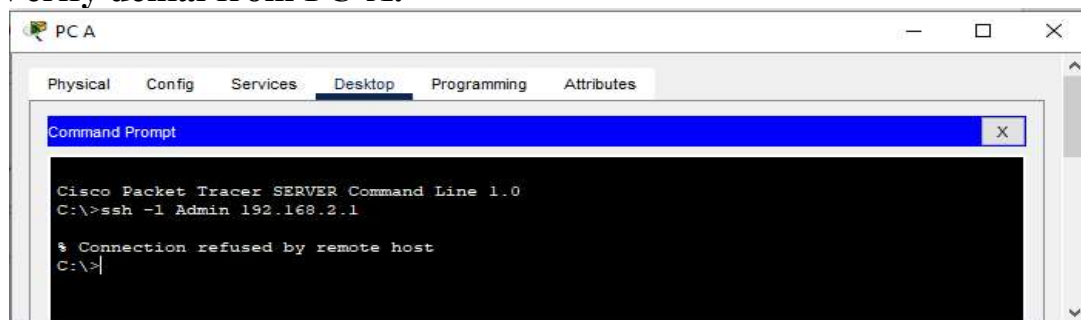
R3(config)#line vty 0 4

R3(config-line)#access-class 10 in

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



Step 4: Verify denial from PC-A.

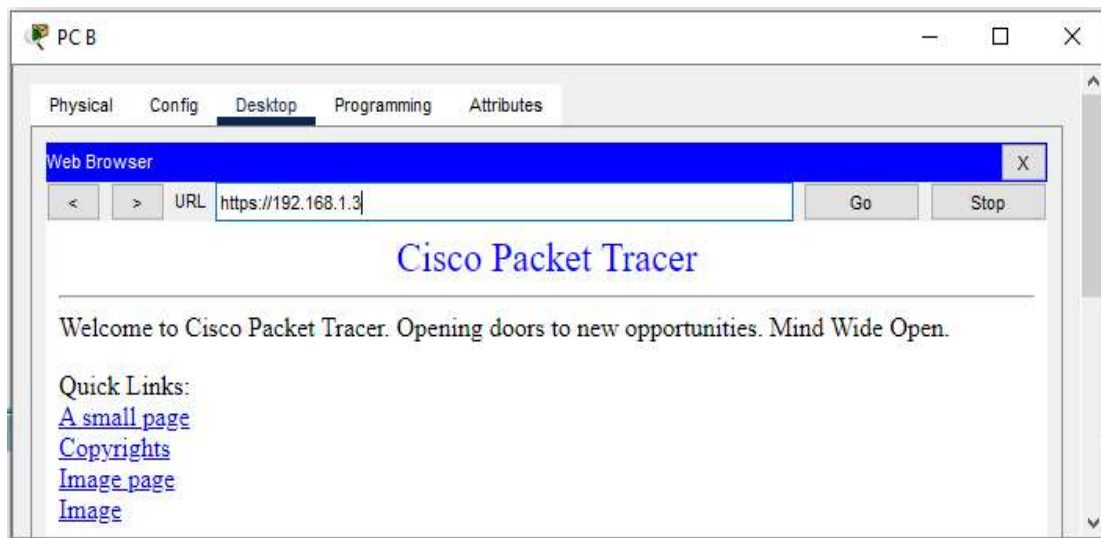


Part 4: Create a Numbered IP ACL 120 on R1

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

Be sure to disable HTTP and enable HTTPS on server PC-A in Services tab.





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

```
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
R1(config)#
```

Step 3: Apply the ACL to interface

```
R1(config)#int se0/1/0
R1(config-if)#ip access-group 120 in
```

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

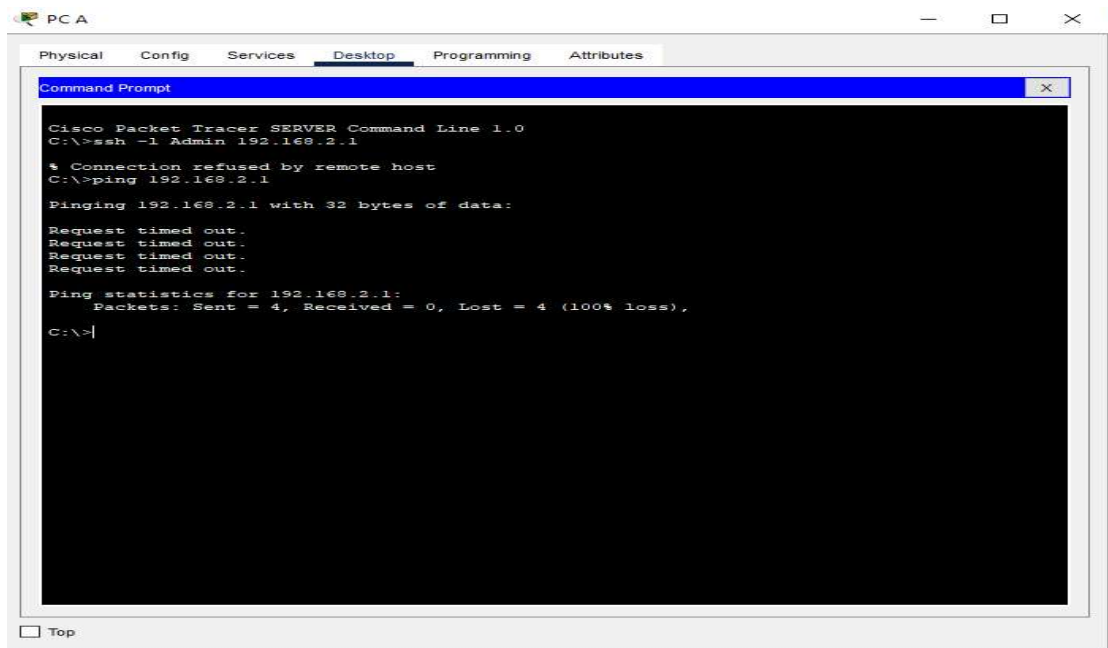
Desktop->Web Browser->192.168.1.3



Part 5: Modify an Existing ACL on R1

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

```
PCA> ping 192.168.2.1
```



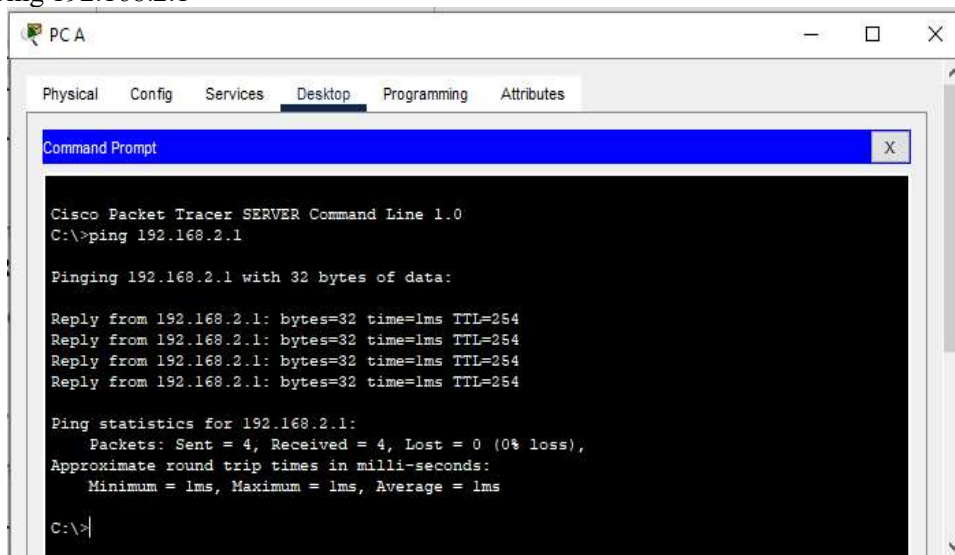
Step 2: Make any necessary changes to ACL 120 to permit and deny the specified traffic.

```

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
R1(config)#
  
```

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

PCA> ping 192.168.2.1



Part 6: Create a Numbered IP ACL 110 on R3

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

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

Step 2: Apply the ACL to interface

```
R3(config)#int gig0/1
```

```
R3(config-if)#ip access-group 110 in
```

Part 7: Create a Numbered IP ACL 100 on R3

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

```
R3(config-if)#access-list 100 permit tcp 10.0.0.0 0.255.255.255 host 192.168.3.3 eq 22
```

```
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

```
R3(config)#interface se0/1/0
```

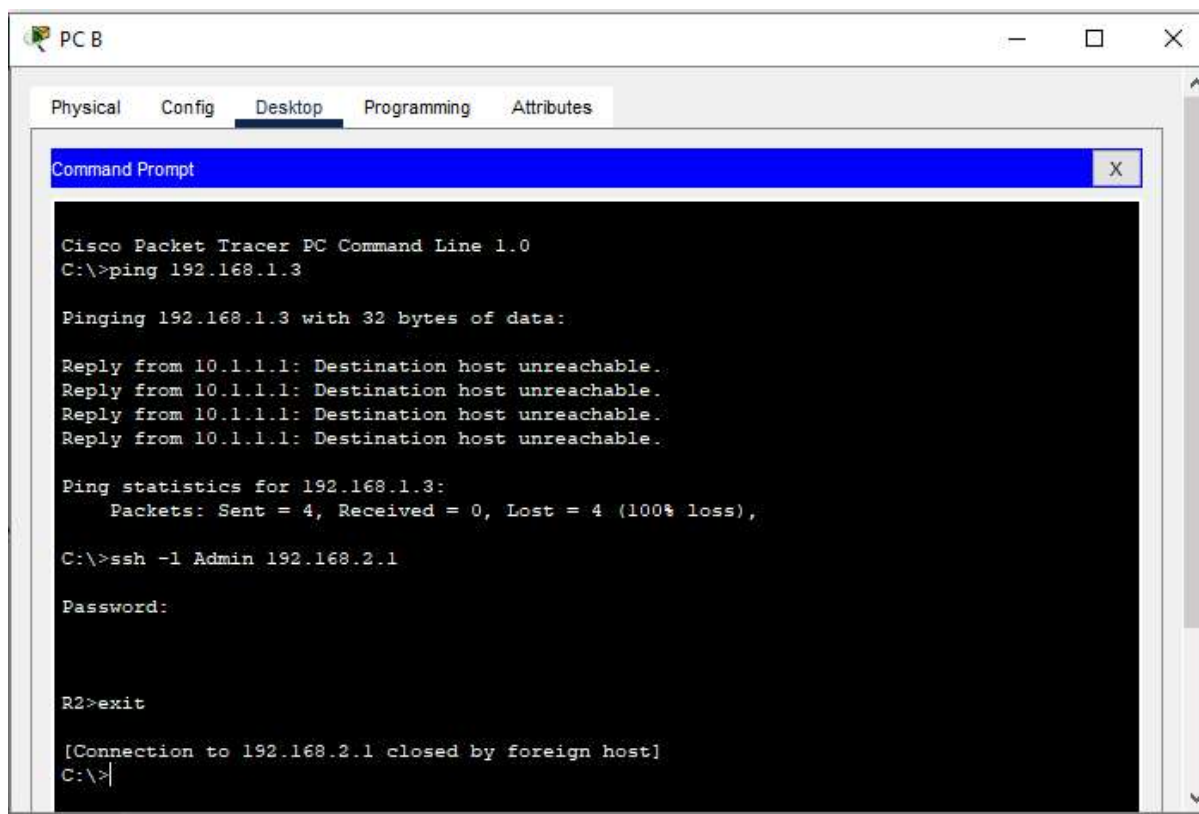
```
R3(config-if)#ip access-group 100 in
```

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

```
PC B> ping 192.168.1.3
```

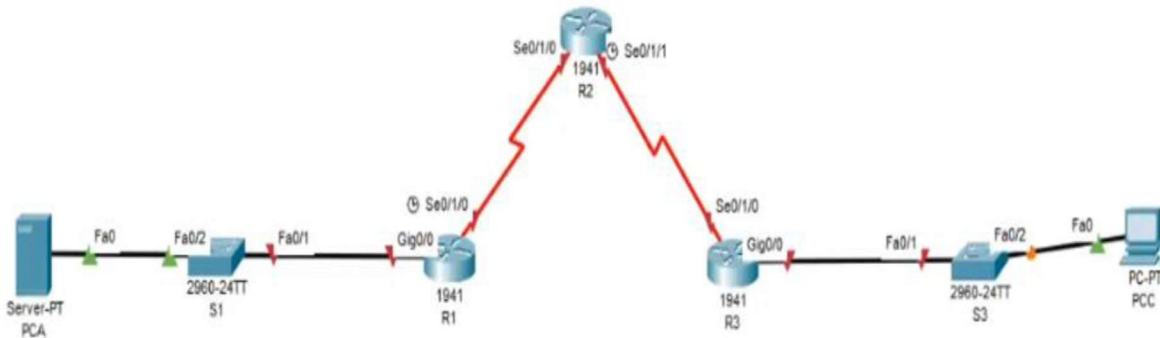
```
PC B> ssh -l admin 192.168.2.1 Password: Admin123
```

```
R2>exit
```



PRACTICAL 6

AIM: Configuring a Zone-Based Policy Firewall(ZPF)



Addressing Table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	gig0/0	192.168.1.1	255.255.255.0	N/A
	Se0/1/0 (DCE)	10.1.1.1	255.255.255.252	N/A
R2	Se0/1/0	10.1.1.2	255.255.255.252	N/A
	Se0/1/1 (DCE)	10.2.2.2	255.255.255.252	N/A
R3	gig0/0	192.168.3.1	255.255.255.0	N/A
	Se0/1/1	10.2.2.1	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

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.

Assigning IP address to R1

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#
R1(config)#end
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface GigabitEthernet0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface Serial0/1/0
R1(config-if)#ip address 10.1.1.1 255.0.0.0
```



```
R1(config-if)#ip address 10.1.1.1 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
```

Assigning IP address to R2

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R2
R2(config)#
R2(config)#end
R2#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R2#
%SYS-5-CONFIG_I: Configured from console by console
```

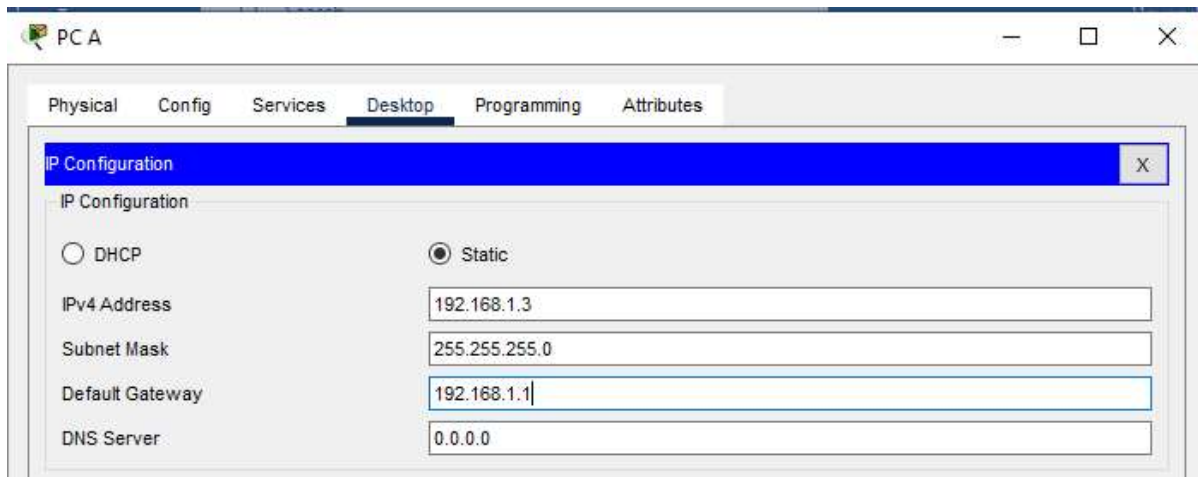
```
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface Serial0/1/0
R2(config-if)#ip address 10.1.1.2 255.0.0.0
R2(config-if)#ip address 10.1.1.2 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up
```

```
R2(config-if)#exit
R2(config)#interface Serial0/1/1
R2(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up
ip address 10.2.2.2 255.255.255.252
R2(config-if)#ip address 10.2.2.2 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#
```

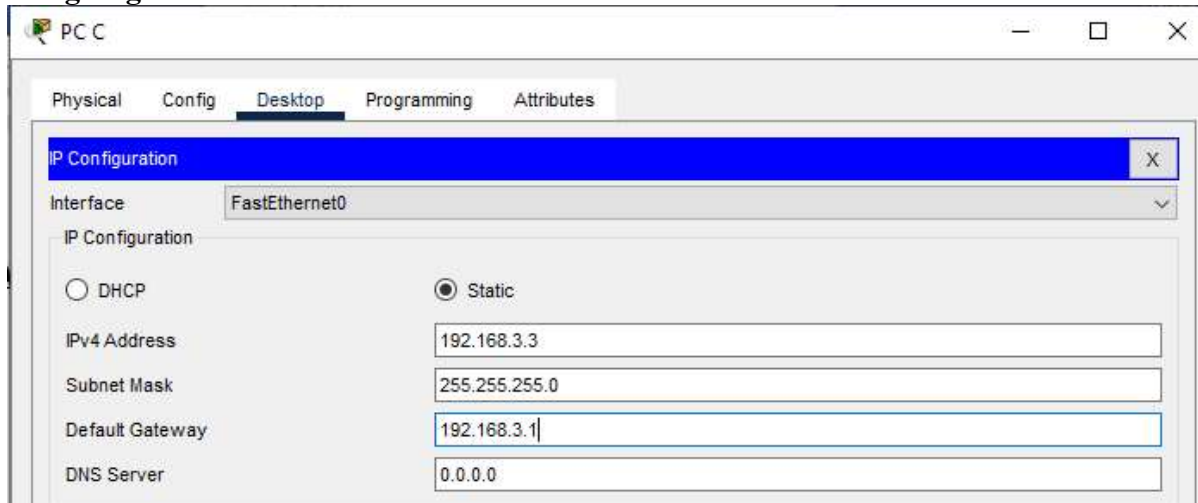
Assigning IP address to R3

```
Router>enable
Router#configure terminal
Router(config)#hostname R3
R3(config)#
R3(config)#end
R3#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface Serial0/1/1
R3(config-if)#ip address 10.2.2.1 255.0.0.0
R3(config-if)#ip address 10.2.2.1 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#
R3(config-if)#exit
R3(config)#interface GigabitEthernet0/0
R3(config-if)#
R3(config-if)#ip address 192.168.3.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#
```

Assigning IP address to PC A



Assigning IP address to PC C



Part 1: Configure Router:

Step 1: Configure console password on router

```
R1(config)#line console 0
R1(config-line)#password conAdmin123
R1(config-line)#login
```

```
R2(config)#line console 0
R2(config-line)#password conAdmin123
R2(config-line)#login
```

```
R3(config)#line console 0
R3(config-line)#password conAdmin123
R3(config-line)#login
```

Step 2: Configure password for vty lines

```
R1(config)#line vty 0 4
R1(config-line)#password vtyp123
R1(config-line)#login
R2(config)#line vty 0 4
R2(config-line)#password vtyp123
R2(config-line)#login
R3(config)#line vty 0 4
R3(config-line)#password vtyp123
R3(config-line)#login
```

Step 3: Configure secret on router

```
R1(config-line)#enable secret enpass123
R2(config-line)#enable secret enpass123
R3(config-line)#enable secret enpass123
```

Step 4: Configure SSH login on router

```
R1(config)#ip domain-name ccnasecurity.com
R1(config)#username admin secret adminpa55
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#crypto key generate rsa
The name for the keys will be: R1.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]

R1(config)#
R2(config)#ip domain-name ccnasecurity.com
R2(config)#username admin secret adminpa55
R2(config)#line vty 0 4
R2(config-line)#login local
R2(config-line)#crypto key generate rsaip domain-name ccnasecurity.com
^
% Invalid input detected at '^' marker.

R2(config-line)#username admin secret adminpa55
R2(config)#line vty 0 4
R2(config-line)#login local
R2(config-line)#crypto key generate rsa
The name for the keys will be: R2.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]

R3(config)#ip domain-name ccnasecurity.com
R3(config)#username admin secret adminpa55
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)#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 4096 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]
```

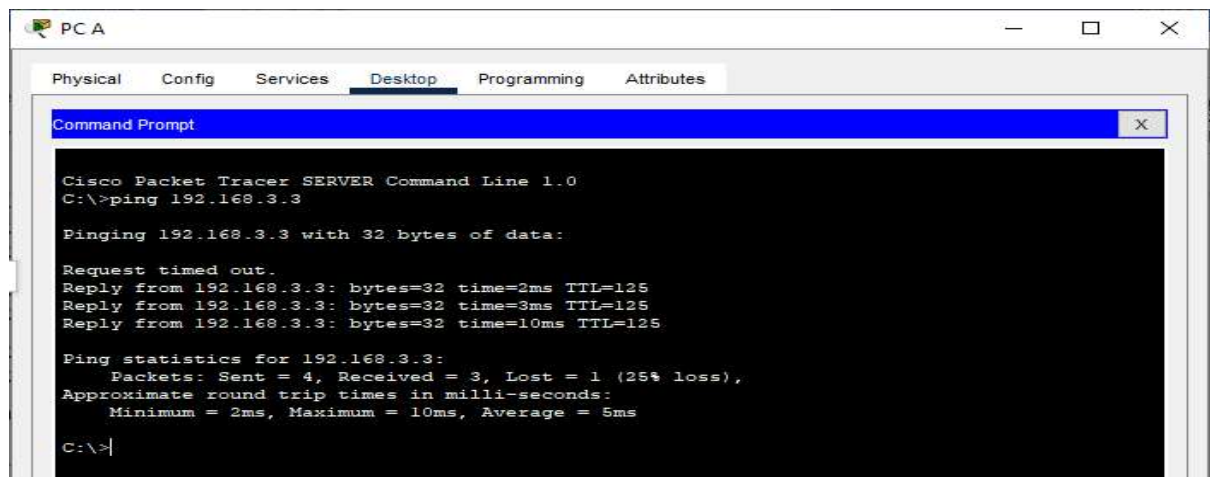
```
R3(config)#
```

Step 5: Configure static routing on routers

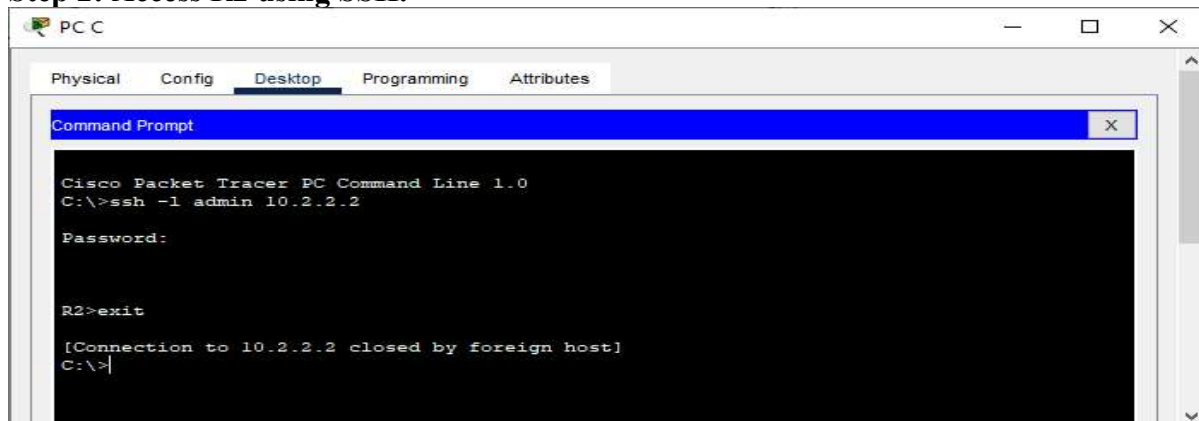
```
R1(config)#ip route 10.2.2.0 255.255.255.252 10.1.1.2
R1(config)#ip route 192.168.3.0 255.255.255.0 10.1.1.2
R2(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.1
R3(config)#ip route 192.168.1.0 255.255.255.0 10.2.2.2
R3(config)#ip route 10.1.1.0 255.255.255.252 10.2.2.2
```

Part 2: Verify Basic Network Connectivity

Step 1: Check connectivity from PCA to PCC



Step 2: Access R2 using SSH.



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



Part 3: Create the Firewall Zones on R3

Step 1: Verify that the Security Technology package

R3# show version

License UDI:

Device#	PID	SN
*0	CISCO2911/K9	FTX16249F23-

Technology Package License Information for Module:'c2900'

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

Configuration register is 0x2102

Step 2: Enable the Security Technology package

```
R3(config)#license boot module c1900 technology-package securityk9
R3(config)#exit
R3#
%SYS-5-CONFIG_I: Configured from console by console
```

Step 3: Reload the router

R3# reload

Step 4: Verify that the Security Technology package

Technology Package License Information for Module:'c1900'

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

Configuration register is 0x2102

Step 5: Create an internal zone and external zone.

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#zone security IN-ZONE
R3(config-sec-zone)#exit
R3(config)#zone security OUT-ZONE
R3(config-sec-zone)#exit
R3(config)#
```

Part 4: Identify Traffic Using a Class-Map

Step 1: Create an ACL that defines internal traffic.

```
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

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

Part 5: Specify Firewall Policies

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

```
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.

```
R3(config-pmap-c)#inspect
    %No specific protocol configured in class IN-NET-CLASS-MAP for inspection. All protocols will be
    inspected
R3(config-pmap-c)#exit
R3(config-pmap)#exit
```

Part 6: Apply Firewall Policies

Step 1: Create a pair of zones.

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

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

```
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.

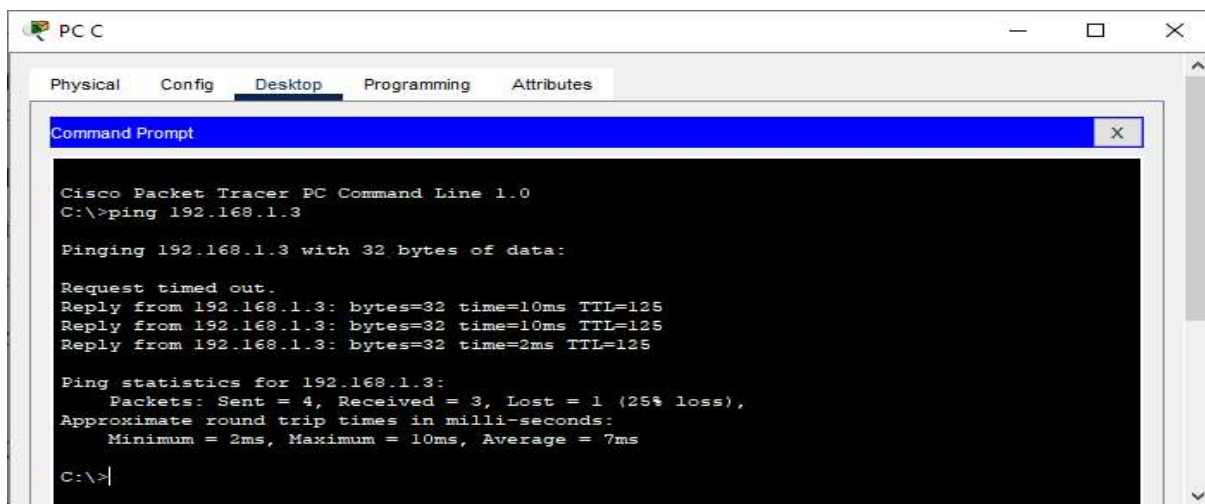
```
R3(config)#int gig0/0
R3(config-if)#zone-member security IN-ZONE
R3(config-if)#exit
R3(config)#int se0/1/1
R3(config-if)#zone-member security OUT-ZONE
R3(config-if)#exit
R3(config)#
```

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

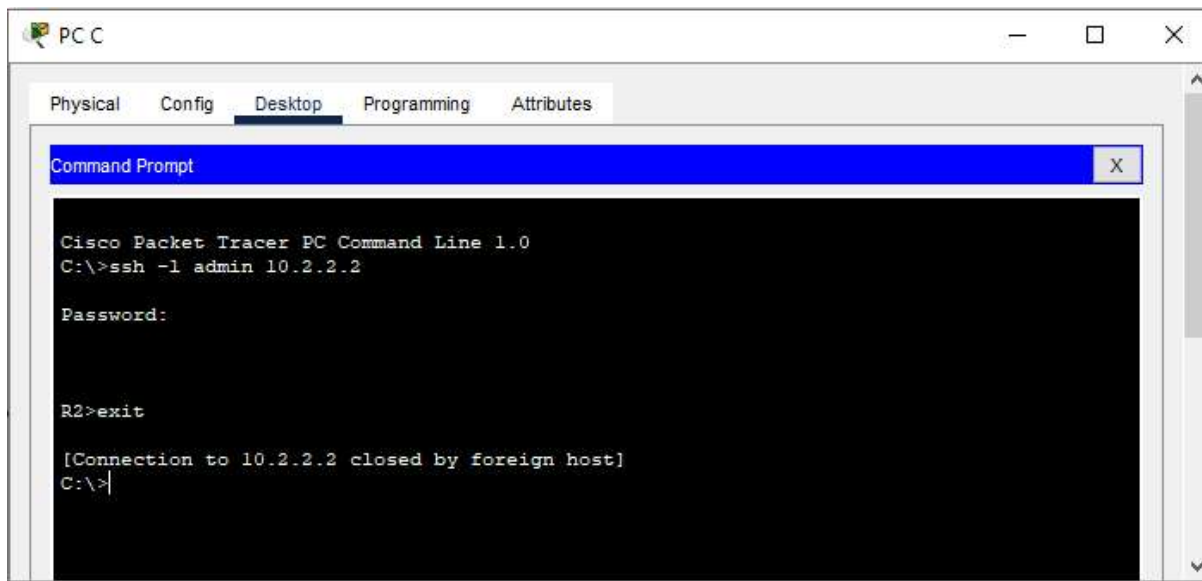
```
R3# reload
```

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

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



Step 2: Access R2 using SSH.



Step 3: View established sessions

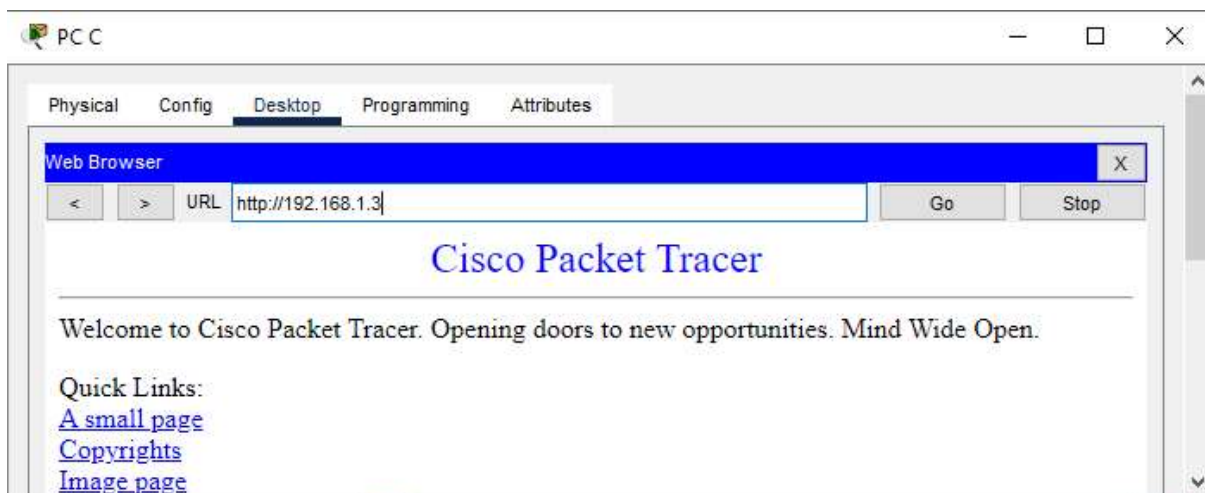
```
R3#show policy-map type inspect zone-pair sessions
policy exists on zp IN-2-OUT-ZPAIR
Zone-pair: IN-2-OUT-ZPAIR

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

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

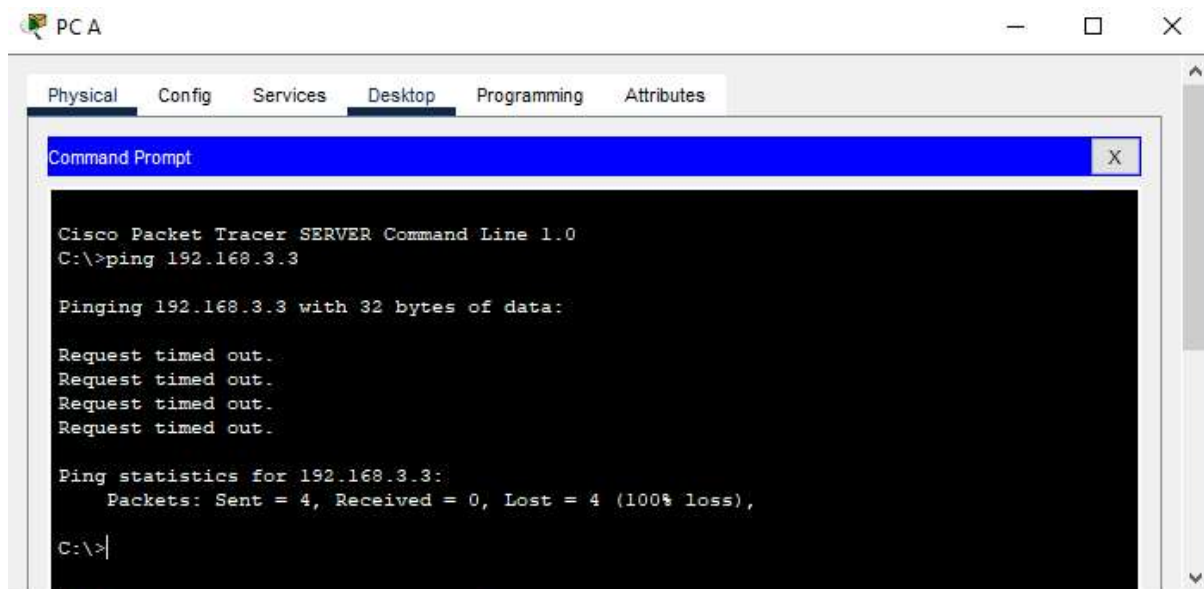
Class-map: class-default (match-any)
Match: any
Drop (default action)
0 packets, 0 bytes
R3#
```

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



Part 8: Test Firewall Functionality from OUT-ZONE to IN-ZONE

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



Step 2: From R2, ping PC-C.

```
R2#ping 192.168.3.3
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.3.3, timeout is 2 seconds:
```

```
.....
```

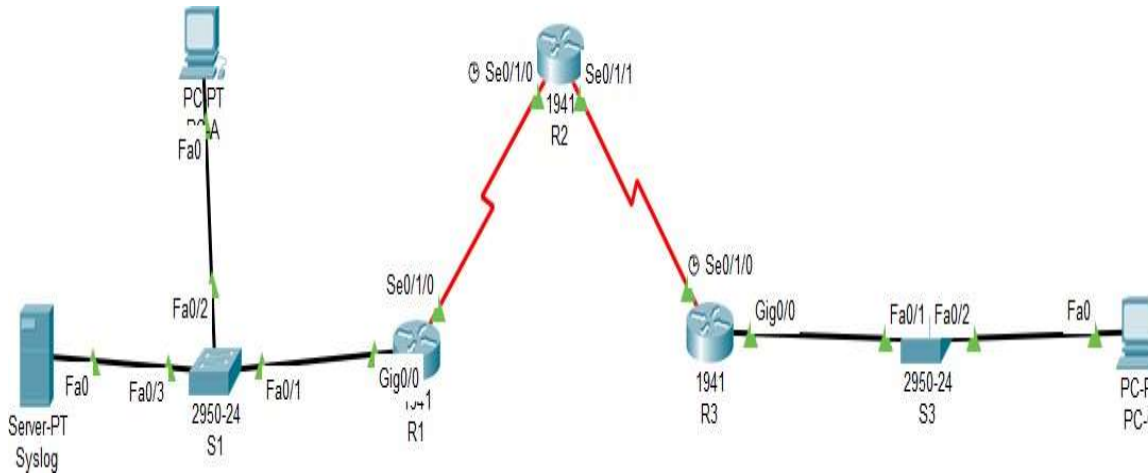
```
Success rate is 0 percent (0/5)
```

```
R2#
```

Practical 7

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

Topology:



Addressing Table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	gig0/0	192.168.1.1	255.255.255.0	N/A
	Se0/1/0	10.1.1.1	255.255.255.252	N/A
R2	Se0/1/0	10.1.1.2	255.255.255.252	N/A
	Se0/1/1	10.2.2.2	255.255.255.252	N/A
R3	gig0/0	192.168.3.1	255.255.255.0	N/A
	Se0/1/0	10.2.2.1	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

Objectives

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

Assigning IP address to R1

```
Router>enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#
R1(config)#end
R1(config)#interface GigabitEthernet0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface Serial0/0/0
R1(config-if)#ip address 10.1.1.1 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
```

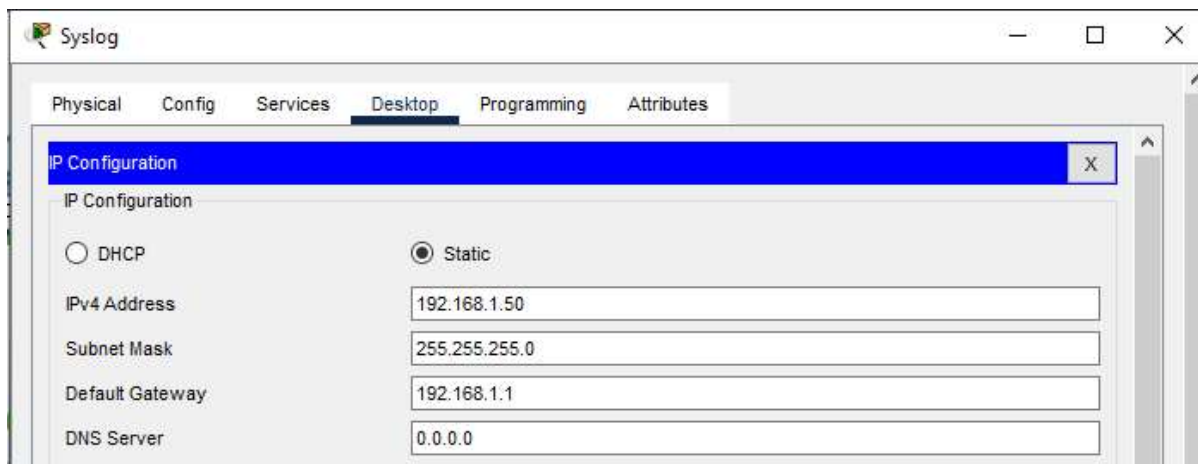
Assigning IP address to R2

```
Router>enable
Router#configure terminal
Router(config)#hostname R2
R2(config)#
R2(config)#end
R2(config)#interface Serial0/0/0
R2(config-if)#ip address 10.1.1.2 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#
R2(config-if)#exit
R2(config)#interface Serial0/0/1
R2(config-if)#ip address 10.2.2.2 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
```

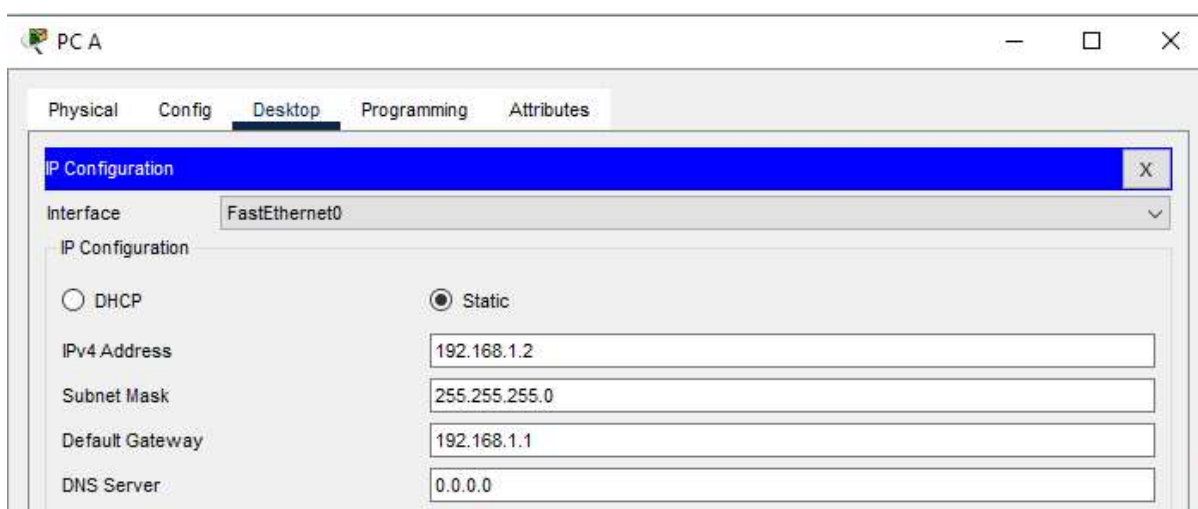
Assigning IP address to R3

```
Router>enable
Router#configure terminal
Router(config)#hostname R3
R3(config)#interface GigabitEthernet0/0
R3(config-if)#ip address 192.168.3.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#interface Serial0/0/1
R3(config-if)#ip address 10.2.2.1 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#
```

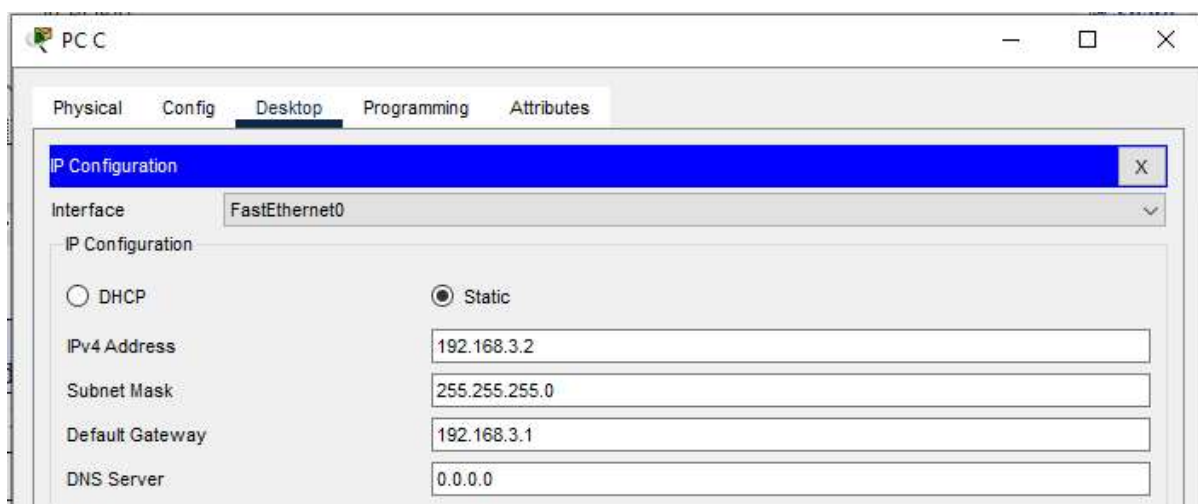
Assigning IP address to Syslog



Assigning IP address to PC A



Assigning IP address to PC C



Part 1: Configure router

Step 1: Configure secret on router

```
R1(config)#enable secret enpass123
R2(config)#enable secret enpass123
R3(config)#enable secret enpass123
```

Step 2: Configure console password on router

Execute command on all routers

```
R1(config)#line console 0
R1(config-line)#password compass123
R1(config-line)#login
R1(config-line)#exit
R1(config)#
R2(config)#line console 0
R2(config-line)#password compass123
R2(config-line)#login
R2(config-line)#exit
R2(config)#
R3(config)#line console 0
R3(config-line)#password compass123
R3(config-line)#login
R3(config-line)#exit
R3(config)#
```

Step 3: Configure SSH login on router

Execute command on all routers

```
R1(config)#ip domain-name ccnasecurity.com
R1(config)#username Admin secret Admin123
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#exit
R1(config)#crypto key generate rsa
The name for the keys will be: R1.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]
```

```
R1(config)#
```

```
R2(config)#ip domain-name ccnasecurity.com
R2(config)#username Admin secret Admin123
R2(config)#line vty 0 4
R2(config-line)#login local
R2(config-line)#exit
R2(config)#crypto key generate rsa
The name for the keys will be: R2.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]
```

```
R2(config)#
```

```

R3(config)#ip domain-name ccnasecurity.com
R3(config)#username Admin secret Admin123
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)#exit
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 4096 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]

R3(config)#

```

Step 4: Configure OSPF on routers

Execute command on R1

```

R1(config)#router ospf 1
R1(config-router)#network 10.1.1.0 0.0.0.3 area 0
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0
R1(config-router)#

```

Execute command on R2

```

R2(config)#router ospf 1
*Mar 1 0:46:21.323: %SSH-5-ENABLED: SSH 1.99 has been enabled
R2(config-router)#network 10.1.1.0 0.0.0.3 area 0
R2(config-router)#network 10.2.2.0 0.0.0.3 area 0
00:54:55: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on Serial0/0/0 from LOADING to FULL,
Loading Done

R2(config-router)#

```

Execute command on R3

```

R3(config)#router ospf 1
*Mar 1 0:49:52.31: %SSH-5-ENABLED: SSH 1.99 has been enabled
R3(config-router)#network 10.2.2.0 0.0.0.3 area 0
R3(config-router)#network 192.168.3.0 0.0.0.255 area 0
00:56:30: %OSPF-5-ADJCHG: Process 1, Nbr 10.2.2.2 on Serial0/0/1 from LOADING to FULL,
Loading Done
|

```

Part 2: Enable IOS IPS

Step 1: Enable the Security Technology package

R1# show version

```

Technology Package License Information for Module:'c1900'

```

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

```

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

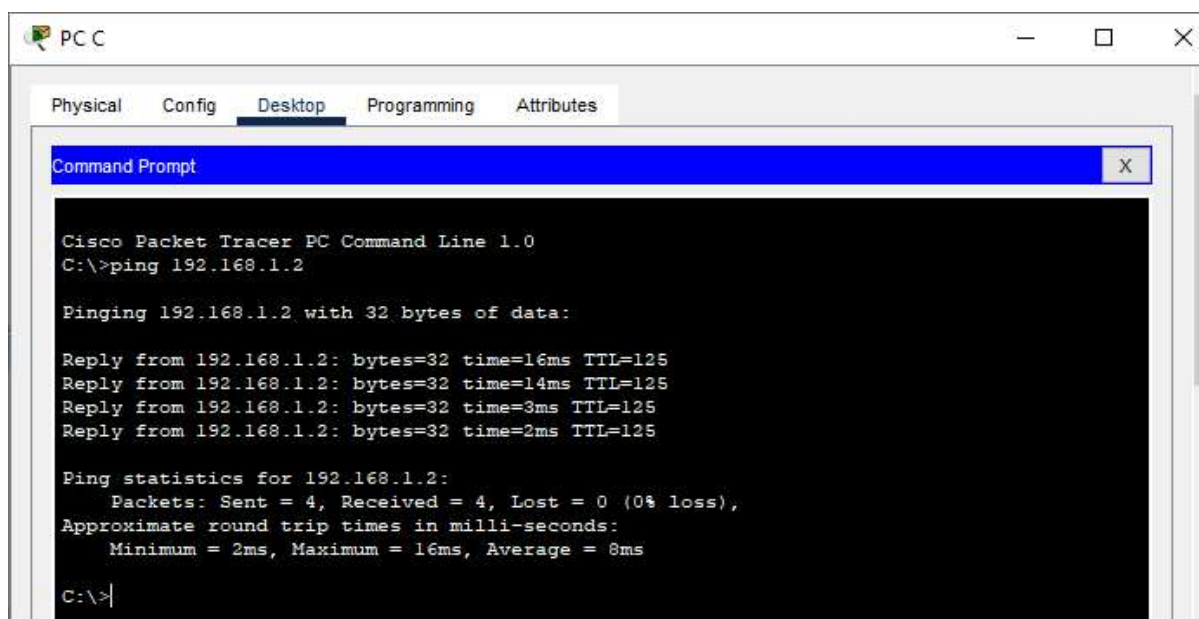
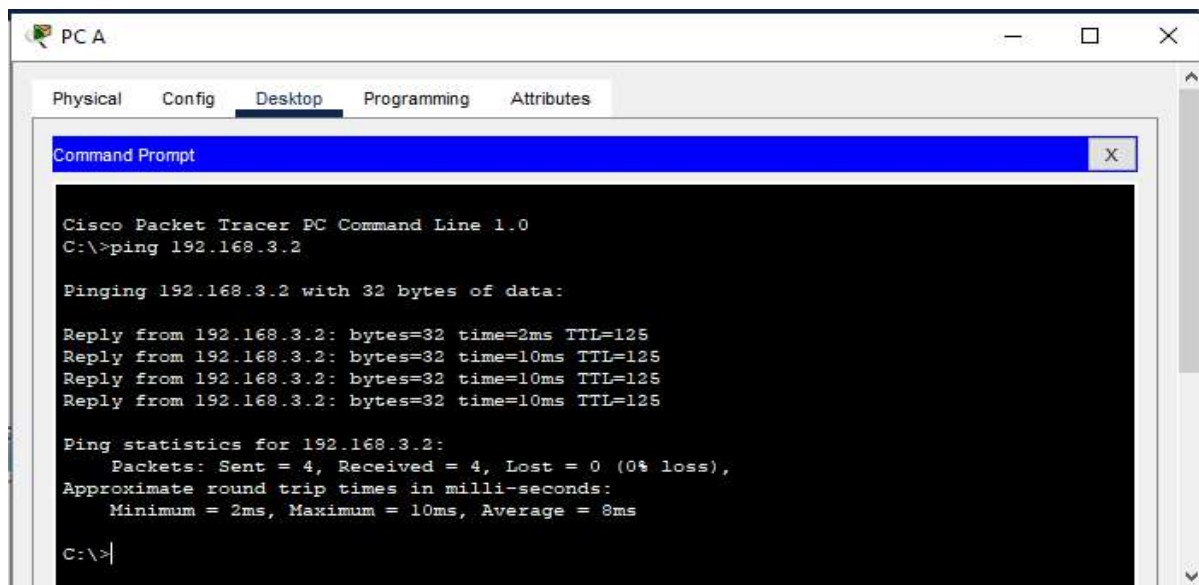
```

R1# reload
R1# show version

Technology Package License Information for Module:'c1900'

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

Step 2: Verify network connectivity



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


```
R1#mkdir ipsdir
Create directory filename [ipsdir]?
Created dir flash: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
R1(config)#
```

Step 6: Enable logging.

```
R1(config-if)#ip ips notify log
R1(config)#^Z
R1#

R1#clock set 04:07:00 06 April 2024
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#service timestamps log datetime msec
R1(config)#logging host 192.168.1.50
R1(config)#|
```

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?
category
R1(config-ips-category)#category ios_ips basic
R1(config-ips-category-action)#retired false
R1(config-ips-category-action)#exit
R1(config-ips-category)#exit
Do you want to accept these changes? [confirm]y
```

Step 8: Apply the IPS rule to an interface.

```
R1(config)#int gig0/0
R1(config-if)#ip ips iosips out
R1(config-if)#
%IPS-6-ENGINE_BUILDS_STARTED: 00:44:38 UTC Mar 01 1993

%IPS-6-ENGINE_BUILDING: atomic-ip - 3 signatures - 1 of 13 engines

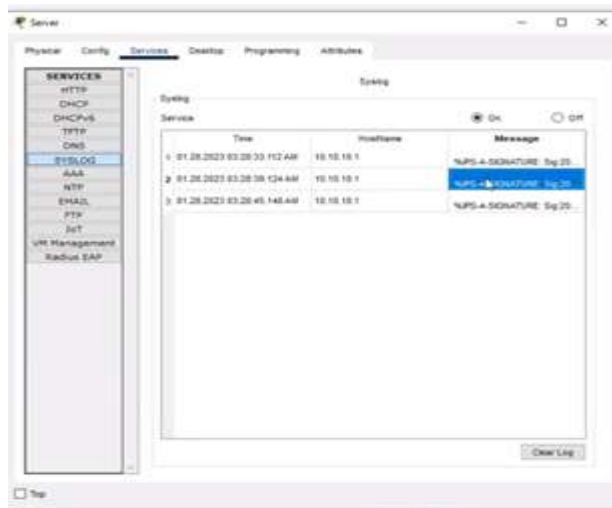
%IPS-6-ENGINE_READY: atomic-ip - build time 8 ms - packets for this engine will be
scanned

%IPS-6-ALL_ENGINE_BUILDS_COMPLETE: elapsed time 8 ms
R1(config-if)#|
```

Step 9: Use show commands to verify IPS.

```
R1#show ip ips all
```

Step 10: View the syslog messages.



Part 3: Modify the Signature

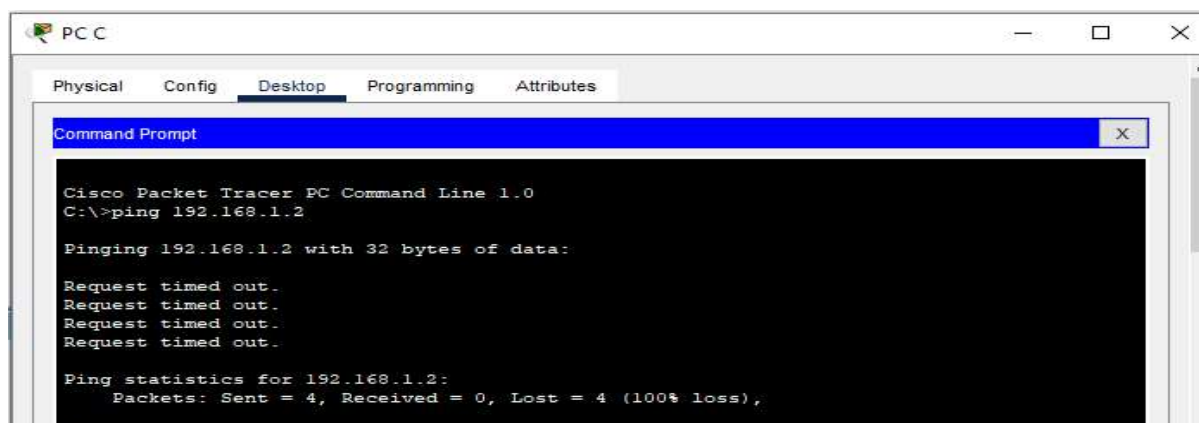
Step 1: Change the event-action of a signature.

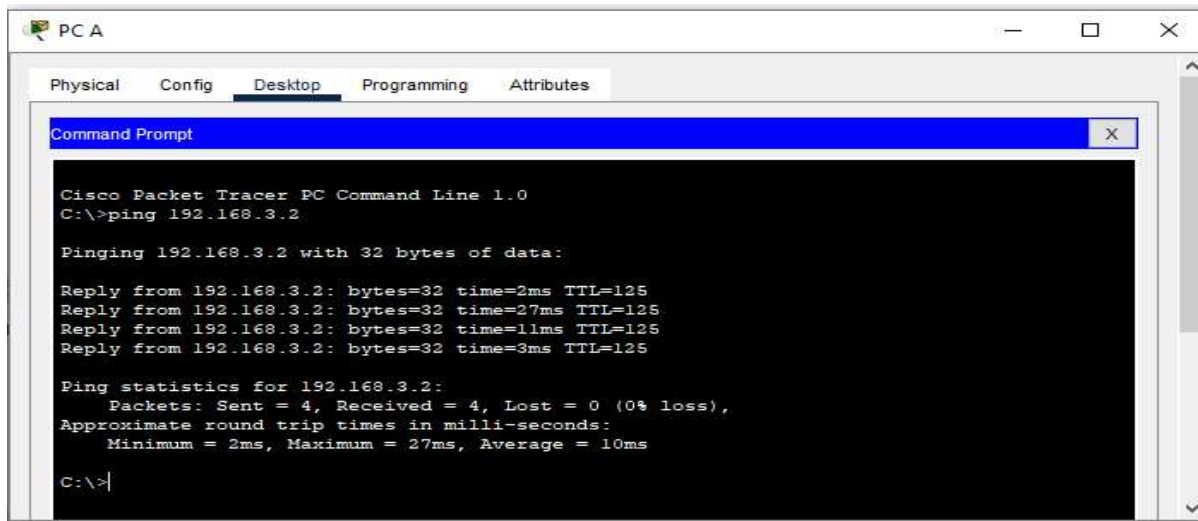
```
R1(config)#ip ips signature-definition
R1(config-sigdef)#signature 2004 0
R1(config-sigdef-sig)#status
R1(config-sigdef-sig-status)#retired false
R1(config-sigdef-sig-status)#enabled true
R1(config-sigdef-sig-status)#exit
R1(config-sigdef-sig)#engine
R1(config-sigdef-sig-engine)#event-action produce-alert
R1(config-sigdef-sig-engine)#event-action deny-packet-inline
R1(config-sigdef-sig-engine)#exit
R1(config-sigdef-sig)#exit
R1(config-sigdef)#exit
Do you want to accept these changes? [confirm]
%IPS-6-ENGINE_BUILDS_STARTED:
%IPS-6-ENGINE_BUILDING: atomic-ip - 303 signatures - 3 of 13 engines
%IPS-6-ENGINE_READY: atomic-ip - build time 480 ms - packets for this engine will be
scanned
%IPS-6-ALL_ENGINE_BUILDS_COMPLETE: elapsed time 648 ms
R1(config)#
```

Step 2: Use show commands to verify IPS.

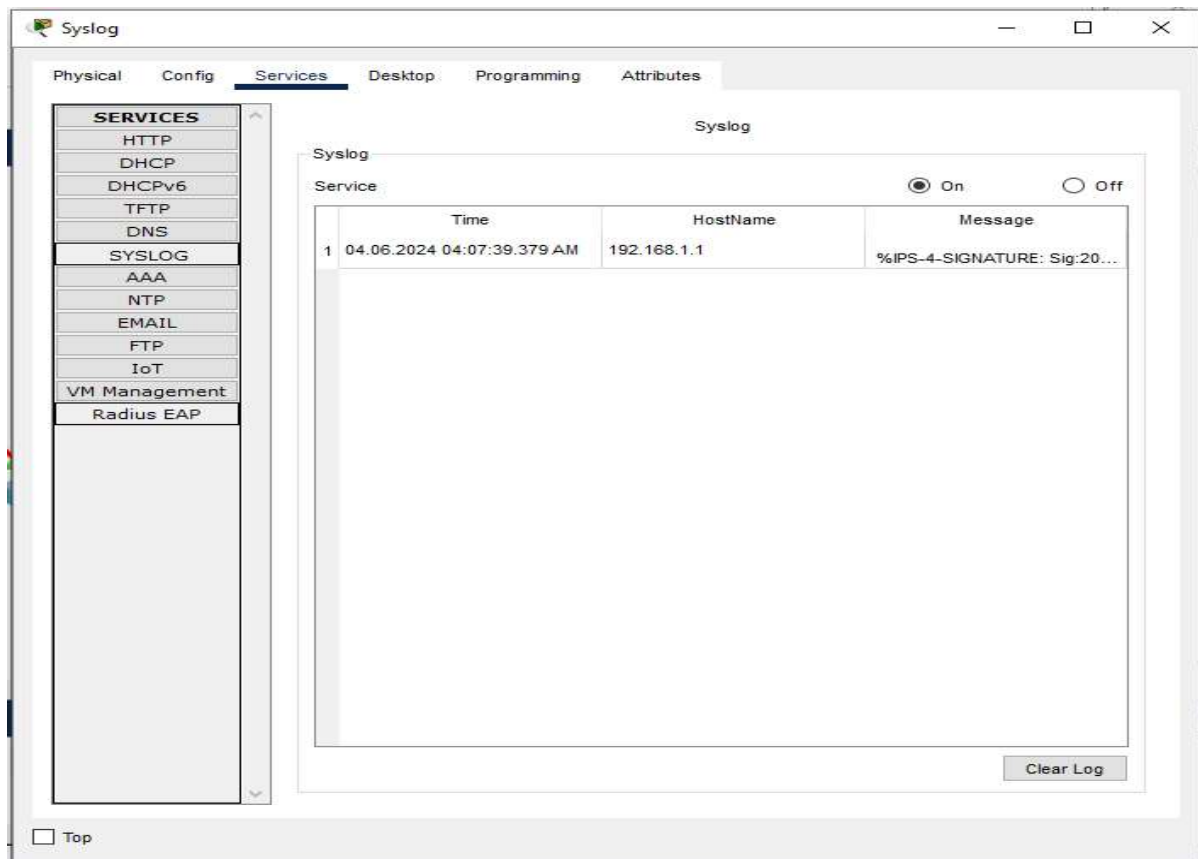
R1#show ip ips all(Running)

Step 3: Verify that IPS is working properly.



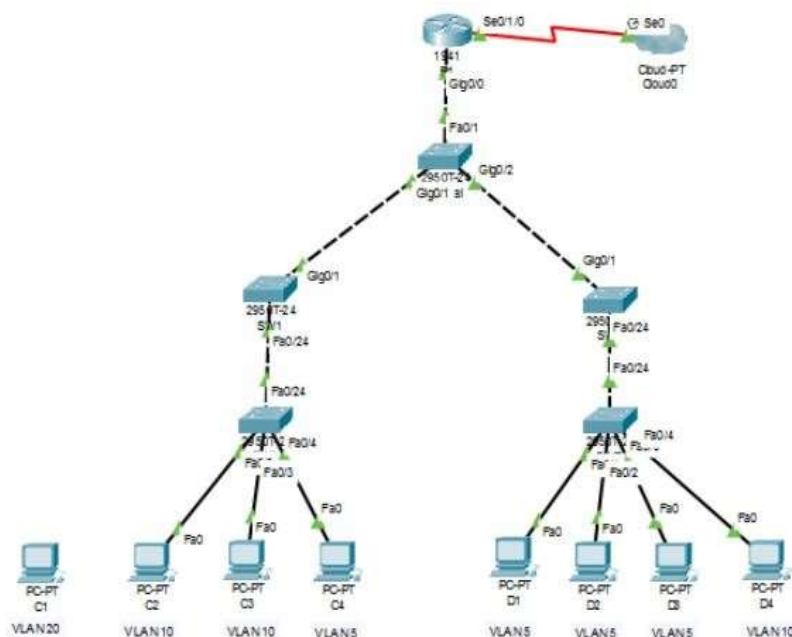


Step 4: View the syslog messages.



PRACTICAL 8

Aim: Layer 2 VLAN Security

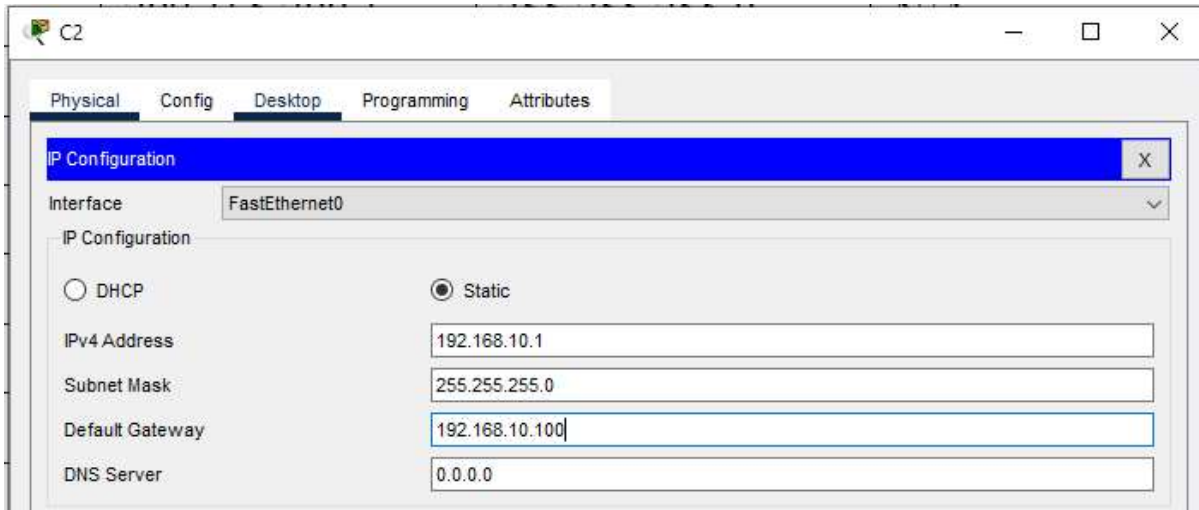


Assigning IP address to R1

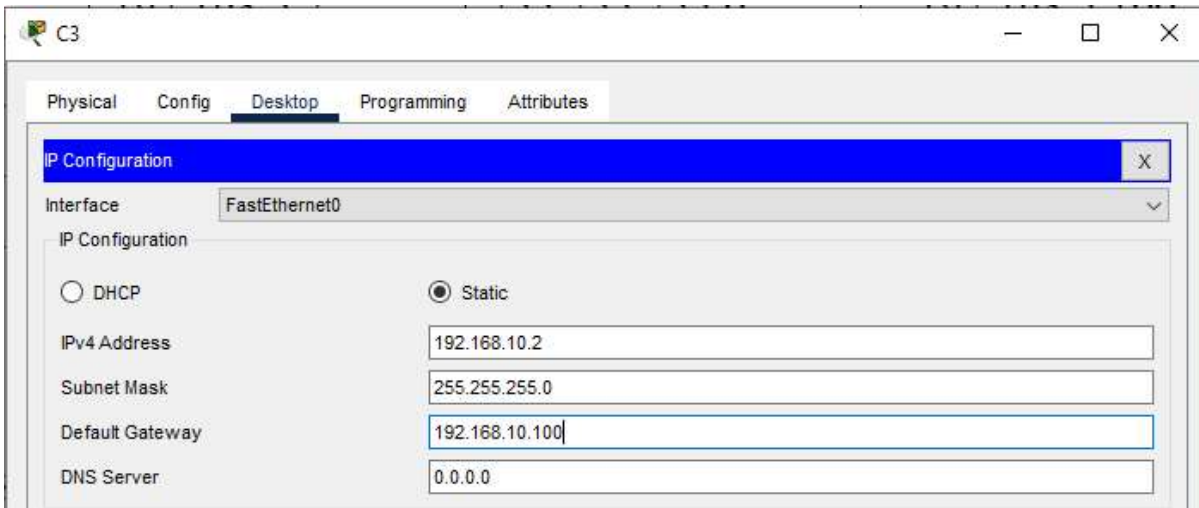
```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#
R1(config)#end
R1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface Serial0/0/0
R1(config-if)#ip address 209.109.100.1 255.255.255.0
R1(config-if)#ip address 209.109.100.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
```

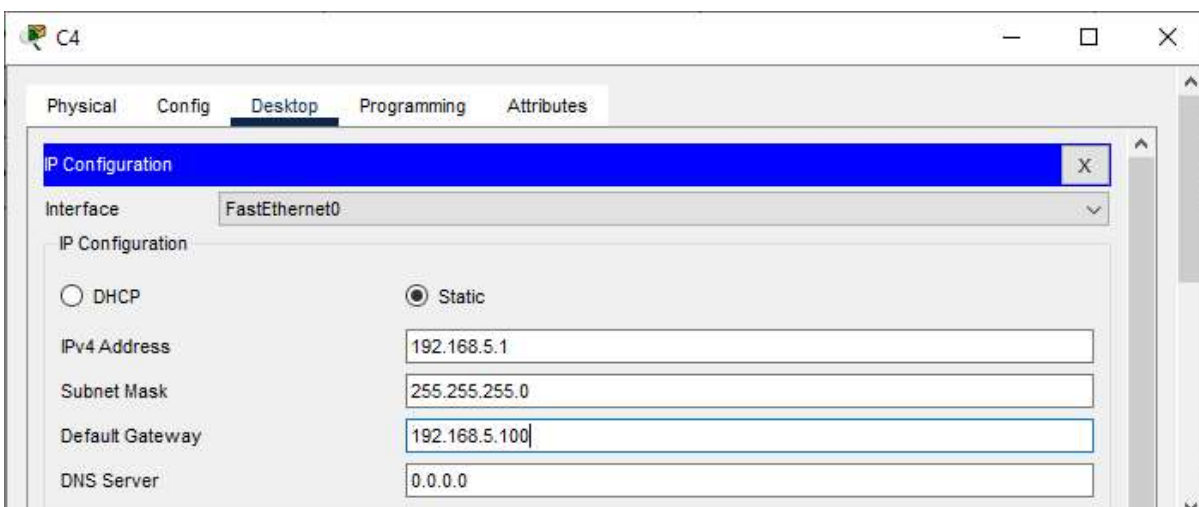
Assigning IP address to C2



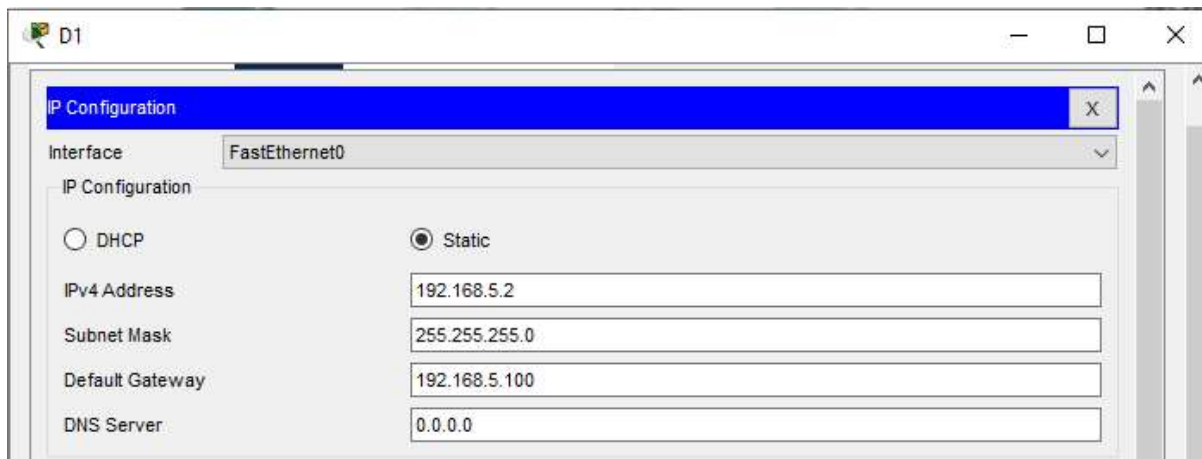
Assigning IP address to C3



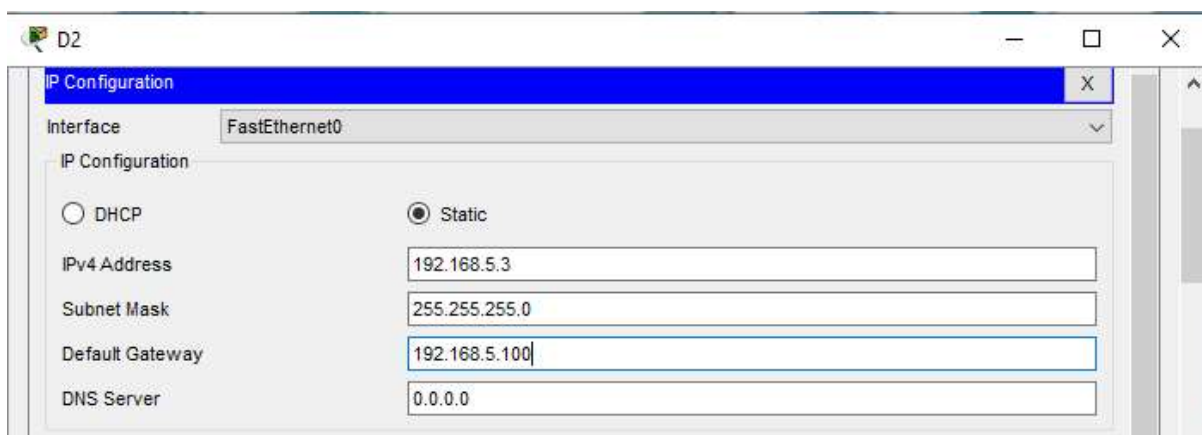
Assigning IP address to C4



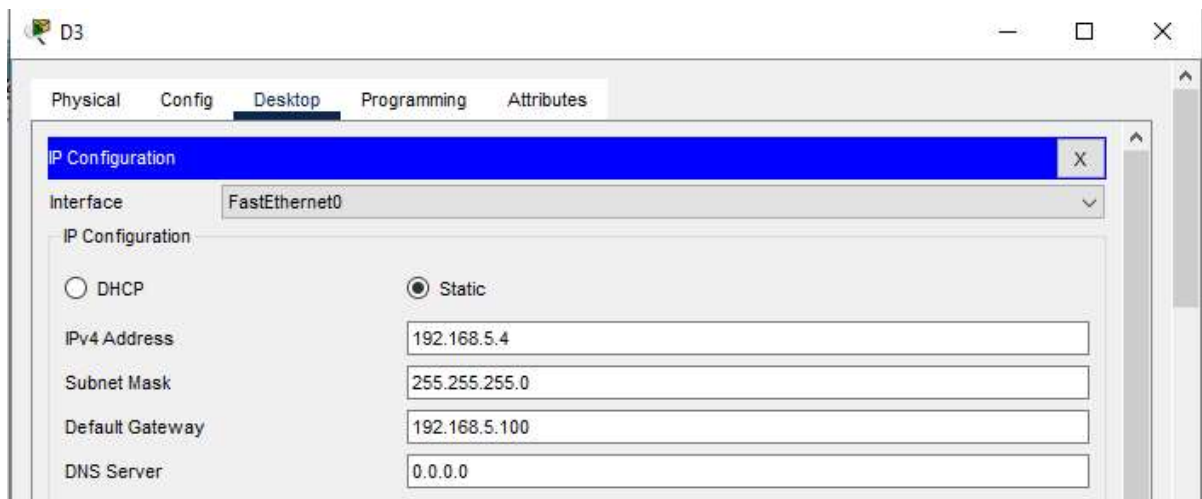
Assigning IP address to D1



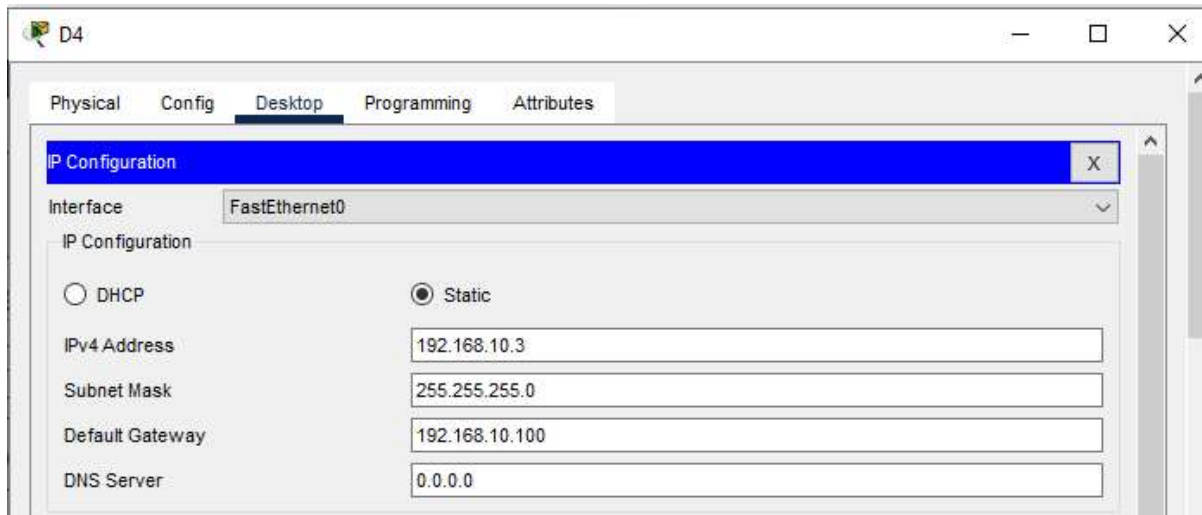
Assigning IP address to D2



Assigning IP address to D3



Assigning IP address to D4



Part 1: Configure Switch/Router

Step 1: Configure secret, console password and ssh login

```
R1(config)#enable secret enpass123
R1(config)#line console 0
R1(config-line)#password conpass123
R1(config-line)#login
R1(config-line)#exit
R1(config)#ip domain-name ccnasecurity.com
R1(config)#username admin secret Admin123
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#exit
R1(config)#crypto key generate rsa
The name for the keys will be: R1.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]

R1(config)#

SW1(config)#enable secret enpass123
SW1(config)#line console 0
SW1(config-line)#password conpass123
SW1(config-line)#login
SW1(config-line)#exit
SW1(config)#ip domain-name ccnasecurity.com
SW1(config)#username admin secret Admin123
SW1(config)#line vty 0 4
SW1(config-line)#login local
SW1(config-line)#exit
SW1(config)#crypto key generate rsa
The name for the keys will be: SW1.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]

SW1(config)#
```

```
SW2(config)#enable secret enpass123
SW2(config)#line console 0
SW2(config-line)#password conpass123
SW2(config-line)#login
SW2(config-line)#exit
SW2(config)#ip domain-name ccnasecurity.com
SW2(config)#username admin secret Admin123
SW2(config)#line vty 0 4
SW2(config-line)#login local
SW2(config-line)#exit
SW2(config)#crypto key generate rsa
The name for the keys will be: SW2.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]
```

```
SW2(config)#
```

```
SWA(config)#enable secret enpass123
SWA(config)#line console 0
SWA(config-line)#password conpass123
SWA(config-line)#login
SWA(config-line)#exit
SWA(config)#ip domain-name ccnasecurity.com
SWA(config)#username admin secret Admin123
SWA(config)#line vty 0 4
SWA(config-line)#login local
SWA(config-line)#exit
SWA(config)#crypto key generate rsa
The name for the keys will be: SWA.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]
```

```
SWA(config)#
```

```
SWB(config)#enable secret enpass123
SWB(config)#line console 0
SWB(config-line)#password conpass123
SWB(config-line)#login
SWB(config-line)#exit
SWB(config)#ip domain-name ccnasecurity.com
SWB(config)#username admin secret Admin123
SWB(config)#line vty 0 4
SWB(config-line)#login local
SWB(config-line)#exit
SWB(config)#crypto key generate rsa
The name for the keys will be: SWB.ccnasecurity.com
Choose the size of the key modulus in the range of 360 to 4096 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]
```

```
SWB(config)#
```

Part 2: Create VLAN and assign access mode and trunk mode to Interfaces

Step 1: Check existing VLAN

Central#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gig0/1, Gig0/2
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

Central#

SW1#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Gig0/1 Gig0/2
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SW1#

SW2#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Gig0/1 Gig0/2
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SW2#

SWA#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/5, Fa0/6, Fa0/7 Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SWA#

SWB#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Gig0/1 Gig0/2
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SWB#

Step 2: Create new VLAN

```
Central(config)#vlan 5
Central(config-vlan)#exit
Central(config)#vlan 10
Central(config-vlan)#exit
Central(config)#vlan 15
Central(config-vlan)#exit
Central(config)#exit
Central#
%SYS-5-CONFIG_I: Configured from console by console
|
SW1(config)#vlan 5
SW1(config-vlan)#exit
SW1(config)#vlan 10
SW1(config-vlan)#exit
SW1(config)#vlan 15
SW1(config-vlan)#exit
SW1(config)#exit
SW1#
%SYS-5-CONFIG_I: Configured from console by console

SW2(config)#vlan 5
SW2(config-vlan)#exit
SW2(config)#vlan 10
SW2(config-vlan)#exit
SW2(config)#vlan 15
SW2(config-vlan)#exit
SW2(config)#exit
SW2#
%SYS-5-CONFIG_I: Configured from console by console

SWA(config)#vlan 5
SWA(config-vlan)#exit
SWA(config)#vlan 10
SWA(config-vlan)#exit
SWA(config)#vlan 15
SWA(config-vlan)#exit
SWA(config)#exit
SWA#
%SYS-5-CONFIG_I: Configured from console by console

SWB(config)#vlan 5
SWB(config-vlan)#exit
SWB(config)#vlan 10
SWB(config-vlan)#exit
SWB(config)#vlan 15
SWB(config-vlan)#exit
SWB(config)#exit
SWB#
%SYS-5-CONFIG_I: Configured from console by console
|
```

Step 3: Check the new VLAN

Central#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gig0/1, Gig0/2
5 VLAN0005	active	
10 VLAN0010	active	
15 VLAN0015	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SW1#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Gig0/1 Gig0/2
5 VLAN0005	active	
10 VLAN0010	active	
15 VLAN0015	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SW2#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Gig0/1 Gig0/2
5 VLAN0005	active	
10 VLAN0010	active	
15 VLAN0015	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SWA#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/5, Fa0/6, Fa0/7 Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23
5 VLAN0005	active	
10 VLAN0010	active	
15 VLAN0015	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SWB#show vlan brief

VLAN Name	Status	Ports
1 default	active	Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Gig0/1 Gig0/2
5 VLAN0005	active	
10 VLAN0010	active	
15 VLAN0015	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

Step 4: Assign access mode to VLAN switch interfaces

```
SWA(config)#int fa0/2
SWA(config-if)#switchport mode access
SWA(config-if)#switchport access vlan 10
SWA(config-if)#exit
SWA(config)#int fa0/3
SWA(config-if)#switchport mode access
SWA(config-if)#switchport access vlan 10
SWA(config-if)#exit
SWA(config)#int fa0/4
SWA(config-if)#switchport mode access
SWA(config-if)#switchport access vlan 5
SWA(config-if)#
SWB(config)#int fa0/1
SWB(config-if)#switchport mode access
SWB(config-if)#switchport access vlan 5
SWB(config-if)#exit
SWB(config)#int fa0/2
SWB(config-if)#switchport mode access
SWB(config-if)#switchport access vlan 5
SWB(config-if)#exit
SWB(config)#int fa0/3
SWB(config-if)#switchport mode access
SWB(config-if)#switchport access vlan 5
SWB(config-if)#exit
SWB(config)#int fa0/4
SWB(config-if)#switchport mode access
SWB(config-if)#switchport access vlan 10
SWB(config-if)#
```

Step 5: Check the access mode allocations

SWA#show vlan brief

5	VLAN0005	active	Fa0/4
10	VLAN0010	active	Fa0/2, Fa0/3

SWB#show vlan brief

5	VLAN0005	active	Fa0/1, Fa0/2, Fa0/3
10	VLAN0010	active	Fa0/4

Step 6: Assign trunk mode to other switch interfaces

```
Central(config)#int range gig0/1-2
Central(config-if-range)#switchport mode trunk
Central(config-if-range)#switchport trunk native vlan 15
Central(config-if-range)#exit
Central(config)#int fa0/1
Central(config-if)#switchport mode trunk
Central(config-if)#switchport trunk native vlan 15
Central(config-if)#exit
SW1(config)#int fa0/24
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk native vlan 15
SW1(config-if)#exit
```

```
SW1(config-if)#int gig0/1
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk native vlan 15
SW1(config-if)#exit
```

```
SW2(config)#int fa0/24
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk native vlan 15
SW2(config-if)#exit
SW2(config-if)#int gig0/1
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk native vlan 15
SW2(config-if)#exit
```

```
SWA(config)#int fa0/24
SWA(config-if)#switchport mode trunk
SWA(config-if)#switchport trunk native vlan 15
SWA(config-if)#exit
```

```
SWB(config)#int fa0/24
SWB(config-if)#switchport mode trunk
SWB(config-if)#switchport trunk native vlan 15
SWB(config-if)#exit
```

Step 7: Check the trunk mode allocations

```
Central#show int trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Gig0/1	on	802.1q	trunking	15
Gig0/2	on	802.1q	trunking	15

Port	Vlans allowed on trunk
Gig0/1	1-1005
Gig0/2	1-1005

Port	Vlans allowed and active in management domain
Gig0/1	1
Gig0/2	1

Port	Vlans in spanning tree forwarding state and not pruned
Gig0/1	1
Gig0/2	1

SW1#show int trunk

Port	Mode	Encapsulation	Status	Native vlan
Fa0/24	on	802.1q	trunking	15
Gig0/1	on	802.1q	trunking	1

Port Vlan allowed on trunk

Fa0/24	1-1005
Gig0/1	1-1005

Port Vlan allowed and active in management domain

Fa0/24	1
Gig0/1	1

Port Vlan in spanning tree forwarding state and not pruned

Fa0/24	1
Gig0/1	1

SW1#

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/24 (15), with SWA FastEthernet0/24 (1).

SW2#show int trunk

Port	Mode	Encapsulation	Status	Native vlan
Fa0/24	on	802.1q	trunking	15
Gig0/2	auto	n-802.1q	trunking	1

Port Vlan allowed on trunk

Fa0/24	1-1005
Gig0/2	1-1005

Port Vlan allowed and active in management domain

Fa0/24	1
Gig0/2	1

Port Vlan in spanning tree forwarding state and not pruned

Fa0/24	1
Gig0/2	1

SW2#

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/24 (15), with SWB FastEthernet0/24 (1).

```
SWA#show int trunk
Port      Mode      Encapsulation  Status      Native vlan
Fa0/2     desirable  n-802.1q       trunking    1
Fa0/3     desirable  n-802.1q       trunking    1
Fa0/4     desirable  n-802.1q       trunking    1
Fa0/24    desirable  n-802.1q       trunking    1

Port      Vlans allowed on trunk
Fa0/2     1-1005
Fa0/3     1-1005
Fa0/4     1-1005
Fa0/24    1-1005

Port      Vlans allowed and active in management domain
Fa0/2     1
Fa0/3     1
Fa0/4     1
Fa0/24    1

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/2     1
Fa0/3     1
Fa0/4     1
Fa0/24    1

SWA#
SWA#
SWA#
SWA#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/24 (1)
with SW1 FastEthernet0/24 (15).
```

```
SWB#show int trunk
Port      Mode      Encapsulation  Status      Native vlan
Fa0/1     desirable  n-802.1q       trunking    1
Fa0/2     desirable  n-802.1q       trunking    1
Fa0/3     desirable  n-802.1q       trunking    1
Fa0/4     desirable  n-802.1q       trunking    1
Fa0/24    desirable  n-802.1q       trunking    1

Port      Vlans allowed on trunk
Fa0/1     1-1005
Fa0/2     1-1005
Fa0/3     1-1005
Fa0/4     1-1005
Fa0/24    1-1005

Port      Vlans allowed and active in management domain
Fa0/1     1
Fa0/2     1
Fa0/3     1
Fa0/4     1
Fa0/24    1

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/1     1
Fa0/2     1
Fa0/3     1
Fa0/4     1
Fa0/24    1
```

Step 8: Create sub-interfaces on router to support VLAN

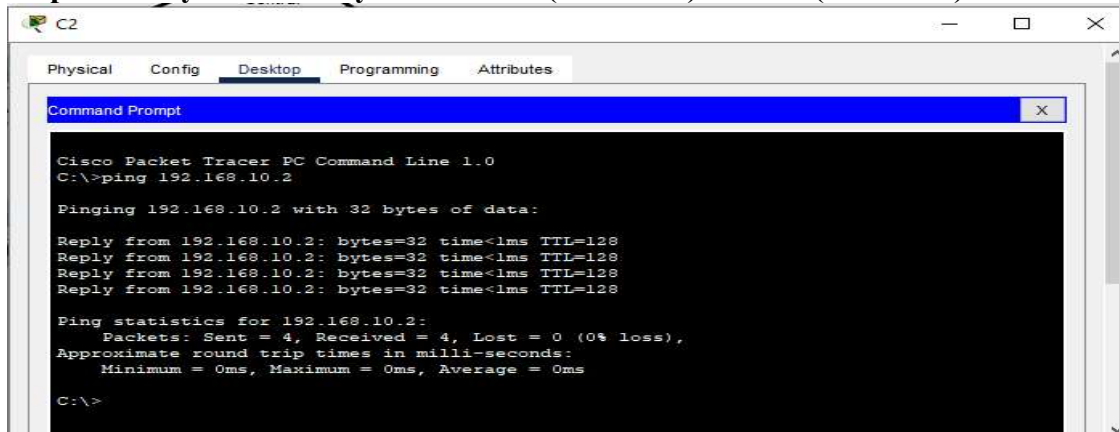
```

R1(config)#int gig0/0.1
R1(config-subif)#encapsulation dot1q 5
R1(config-subif)#ip address 192.168.5.100 255.255.255.0
R1(config-subif)#exit
R1(config)#int gig0/0.2
R1(config-subif)#encapsulation dot1q 10
R1(config-subif)#ip address 192.168.10.100 255.255.255.0
R1(config-subif)#exit
R1(config)#int gig0/0.15
R1(config-subif)#encapsulation dot1q 15
R1(config-subif)#ip address 192.168.15.100 255.255.255.0
R1(config-subif)#exit
R1(config)#

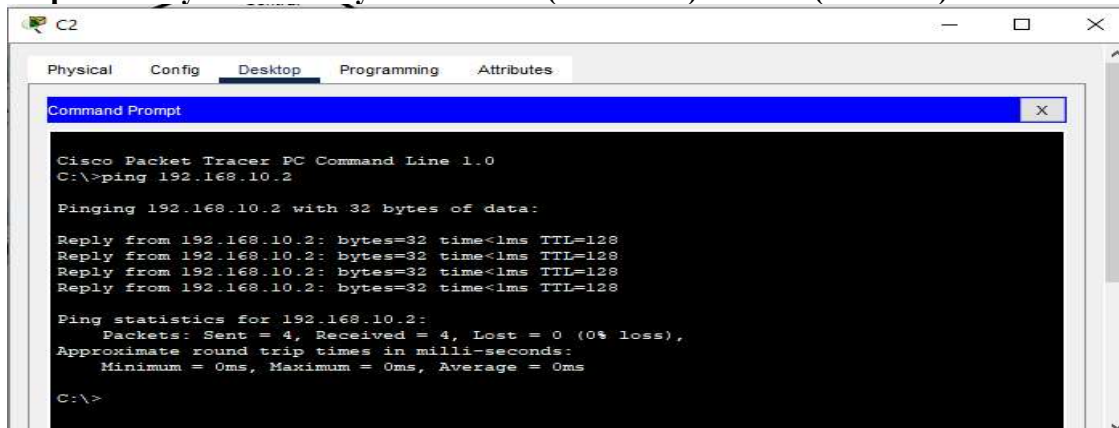
```

Part 3: 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).



Part 4: Create a Redundant Link between SW-1 and SW-2

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

Using a crossover cable, connect port Fa0/23 on SW-1 to port Fa0/23 on SW-2.

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

```

SW1(config)#int fa0/23
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk native vlan 15
SW1(config-if)#switchport nonegotiate

```



```
SW2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)#int fa0/23
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk native vlan 15
SW2(config-if)#switchport nonegotiate
SW2(config-if)#
```

Part 5: Enable VLAN 20 as a Management VLAN

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

```
SWA(config)#vlan 20
SWA(config-vlan)#exit
SWA(config)#int vlan 20
SWA(config-if)#
%LINK-5-CHANGED: Interface Vlan20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up

SWA(config-if)#ip address 192.168.20.1 255.255.255.0
SWA(config-if)#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/24 (1),
with SW1 FastEthernet0/24 (15).

SWA(config-if)#
SWA#
%SYS-5-CONFIG_I: Configured from console by console

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/24 (1),
with SW1 FastEthernet0/24 (15).
```

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

```
SWB(config)#int vlan 20
SWB(config-if)#ip address 192.168.20.2 255.255.255.0
SW1(config-if)#int vlan 20
SW1(config-if)#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/24 (15),
with SWA FastEthernet0/24 (1).

SW1(config-if)#ip address 192.168.20.3 255.255.255.0
SW1(config-if)#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on GigabitEthernet0/1 (1),
with Central GigabitEthernet0/1 (15).

SW2(config-if)#int vlan 20
SW2(config-if)#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on GigabitEthernet0/2 (1),
with Central GigabitEthernet0/2 (15).

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/24 (15),
with SWB FastEthernet0/24 (1).

SW2(config-if)#ip address 192.168.20.4 255.255.255.0
SW2(config-if)#
Central(config-if)#int vlan 20
Central(config-if)#ip address 192.168.20.5 255.255.255.0
Central(config-if)#
```

Step 3: Connect and configure the management PC.

Connect the management PC using copper straight-through to SW-A portFa0/1 and ensure that it is assigned an available IP address 192.168.20.50

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

```
SWA(config-if)#int fa0/1
SWA(config-if)#switchport mode access
SWA(config-if)#switchport access vlan 20
SWA(config-if)#
```

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

```
C1> ping 192.168.20.1
```

C1> ping 192.168.20.2

C1> ping 192.168.20.3

C1> ping 192.168.20.4

C1> ping 192.168.20.5