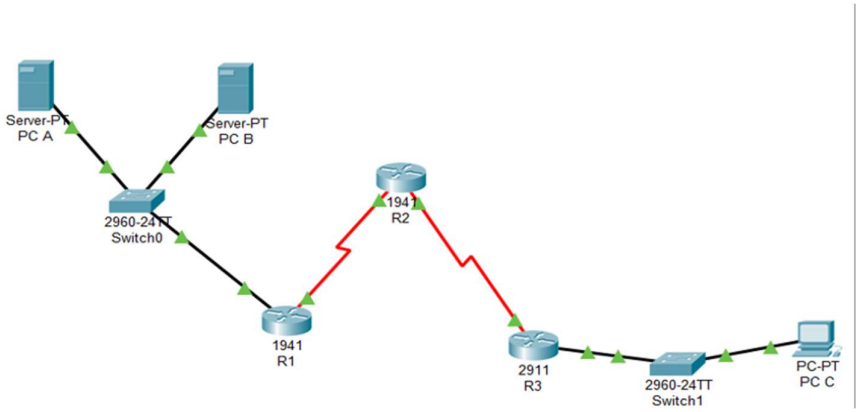
Roll No: A002 **Name :Vidya Bhandari**

	Practical no 1: Configure Routers	Date:26/01/2025

a OSPF MD5 authentication.



Addressing Table

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

Step 1: Test connectivity. All devices should be able to ping all other IP addresses.
Step 2: Configure OSPF MD5 authentication for all the routers in area 0. Configure OSPF MD5 authentication for all the routers in area 0.
R1(config)# router ospf 1
R1(config-router)# area 0 authentication message-digest

```
r1>enable
r1#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
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

```
r2>enable
r2#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
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

```
r3#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
r3(config)#router ospf 1
r3(config-router)#
r3(config-router)#area 0 authentication message-digest
```

Step 3: Configure the MD5 key for all the routers in area 0. Configure an MD5 key on the serial
interfaces on R1, R2 and R3. Use the password MD5pa55 for key 1.
R1(config)# interface s0/0/0
R1(config-if)# ip ospf message-digest-key 1 md5 MD5pa55

```
r1(config)#interface s0/0/0
r1(config-if)#ip ospf message-digest-key 1 md5 MD5pa55
```

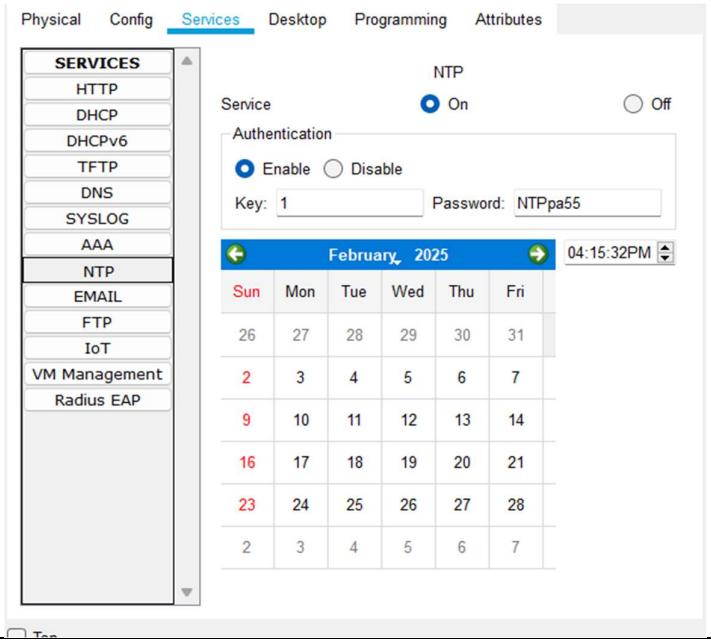
R2(config)# interface s0/0/0
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55
R2(config-if)# interface s0/0/1
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55

```
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.

```
r3(config)#interface s0/0/1
r3(config-if)#ip ospf message-dgest-key 1 md5 MD5pa55
                              ^
% Invalid input detected at '^' marker.

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.



b

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

```
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

```
r1(config)#ntp update-calendar
r2(config)#ntp update-calendar
r3(config)#ntp update-calendar
```

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

Step 4: Configure NTP authentication on the routers. Configure NTP authentication on R1, R2, and R3 using key 1 and password NTPpa55.
R1(config)# ntp authenticate
R1(config)# ntp trusted-key 1
R1(config)# ntp authentication-key 1 md5 NTPpa55

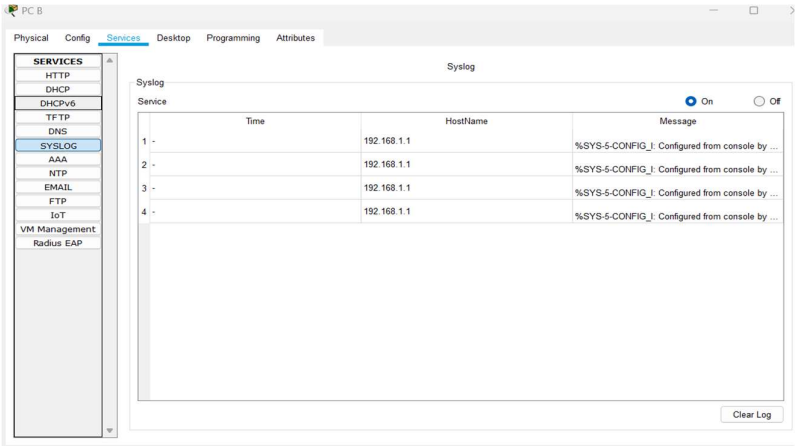
```
r1(config)#ntp authenticate
r1(config)#ntp trusted-key 1
r1(config)#ntp authentication-key 1 md5 NTPpa55
```

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

```
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

```
r3(config)#ntp authenticate
r3(config)#ntp trusted-key 1
r3(config)#ntp authentication-key 1 md5 NTPpa55
```

	Step 5: Configure routers to timestamp log messages. Configure timestamp service for logging on the routers. R1(config)# service timestamps log datetime msec R2(config)# service timestamps log datetime msec R3(config)# service timestamps log datetime msec.
c	Configure Routers to Log Messages to the Syslog Server to log messages to the syslog server. 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. Use the command show logging. 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. 
d	to support SSH connections. Part 4: Configure R3 to Support SSH Connections. Step 1: Configure a domain name. Configure a domain name of ccnasecurity.com on R3. R3(config)# ip domain-name ccnasecurity.com Step 2: Configure users for login to the SSH server on R3. Create a user ID of SSHadmin with the highest possible privilege level and a secret password of ciscosshpa55. R3(config)# username SSHadmin privilege 15 secret ciscosshpa55 Step 3: Configure the incoming vty lines on R3. Use the local user accounts for mandatory login and validation. Accept only SSH connections. R3(config)# line vty 0 4 R3(config-line)# login local R3(config-line)# transport input ssh Step 4: Erase existing key pairs on R3. Any existing RSA key pairs should be erased on the router.

R3(config)# crypto key zeroize rsa

```
R3(config)#ip domain-name ccnasecurity.com
R3(config)# username SSHadmin privilege 15 secret ciscosshpa55
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)# transport input ssh
R3(config-line)#exit
R3(config)# crypto key zeroize rsa
% No Signature RSA Keys found in configuration.

R3(config)#
```

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

R3(config)# crypto key generate rsa

```
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]
```

Step 6: Verify the SSH configuration.

Use the show ip ssh command to see the current settings.

```
R3#show ip ssh
SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
R3#
```

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

To confirm that the values have been changed, use command show ip ssh.

```
R3#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 90 secs; Authentication retries: 2
R3#
```

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.

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

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

PC> ssh –l SSHadmin 192.168.3.1

```
C:\>ssh -l SSHadmin 192.168.3.1

Password:


R3#
```

Step 10: Connect to R3 using SSH on R2.

R2# ssh –v 2 –l SSHadmin 10.2.2.1

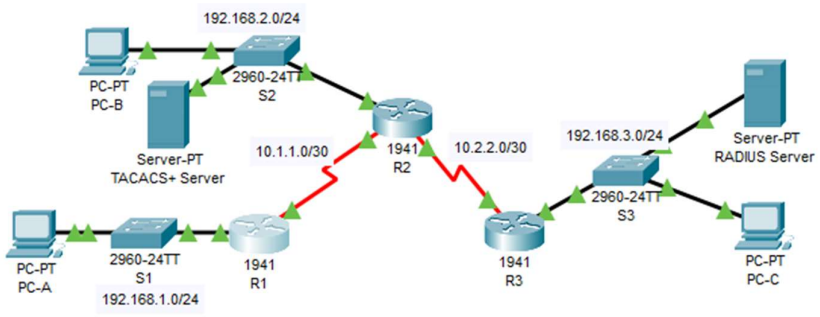
```
R2(config)#exit
R2#
*Feb 02, 22:29:20.2929: SYS-5-CONFIG_I: Configured from console by console
R2# ssh -v 2 -l SSHadmin 10.2.2.1

Password:



R3#
```

Step 11: Check results.

Roll No: A002 **Name :Vidya Bhandari**

	Practical 2 configure AAA Authentication	Date:30/01/2025

a Configure a local user account on Router and configure authenticate on the console and vty lines using local AAA



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/1
R1	S0/0/0 (DCE)	10.1.1.2	255.255.255.252	N/A	N/A
R2	G0/0	192.168.2.1	255.255.255.0	N/A	S2 F0/2
R2	S0/0/0	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/1 (DCE)	10.2.2.1	255.255.255.252	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
R3	S0/0/1	10.2.2.2	255.255.255.252	N/A	N/A
TACACS+ Server	NIC	192.168.2.2	255.255.255.0	192.168.2.1	S2 F0/6
RADIUS Server	NIC	192.168.3.2	255.255.255.0	192.168.3.1	S3 F0/7
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/2
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1	S2 F0/1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Part 1: Configure Local AAA Authentication for Console Access on R1
Step 1: Test connectivity.
• Ping from PC-A to PC-B.
• Ping from PC-A to PC-C.
• Ping from PC-B to PC-C.Step1: Test connectivity. All devices should be able to ping all other IP addresses.

Step 2: Configure a local username on R1.
Configure a username of Admin1 with a secret password of admin1pa55.

```
R1>enable
Password:
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#username Admin1 secret admin1pa55
```

Step 3: Configure local AAA authentication for console access on R1.
Enable AAA on R1 and configure AAA authentication for the console login to use the local database.

```
R1(config)#aaa new-model
R1(config)#aaa authentication login default local
```

Step 4: Configure the line console to use the defined AAA authentication method.
Enable AAA on R1 and configure AAA authentication for the console login to use the default method list.

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

Step 5: Verify the AAA authentication method.
Verify the user EXEC login using the local database.

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

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


User Access Verification

Username: Admin1
Password:
R1>
```

Part 2: Configure Local AAA Authentication for vty lines on R1
Step 1: Configure domain name and crypto key for use with SSH.
a.Use ccnasecurity.com as the domain name on R1.

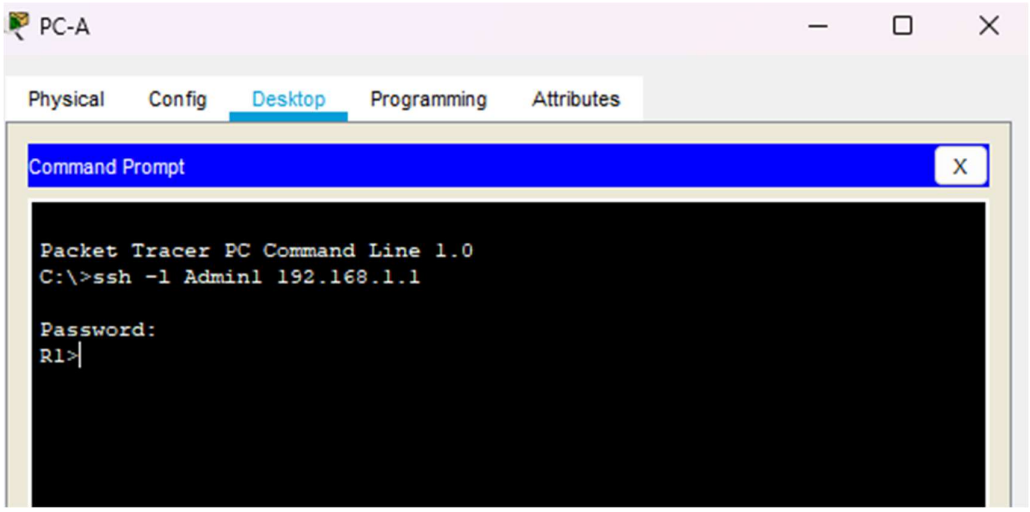
```
R1>en
Password:
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip domain-name ccnasecurity.com
```

b. Create an RSA crypto key using 1024 bits.

```
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 2048 for
your
  General Purpose Keys. Choosing a key modulus greater than 512 may
take
  a few minutes.

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

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

```
R1(config)#aaa authentication login SSH-LOGIN local
*Mar 1 2:16:29.111: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#line vty 0 4
```

	Step 3: Configure the vty lines to use the defined AAA authentication method.
	`R1(config-line)#login authentication SSH-LOGIN`
	`R1(config-line)#transport input ssh`
	`R1(config-line)#end`
	`R1#`
	`%SYS-5-CONFIG_I: Configured from console by console`
	
	Step 4: Verify the AAA authentication method.
	Verify the SSH configuration SSH to R1 from the command prompt of PC-A..
	
	 

b	Verify local AAA authentication from the Router console and the PC-A client Step 1: Configure a backup local database entry called Admin. For backup purposes, configure a local username of Admin2 and a secret password of admin2pa55. R2(config)# username Admin2 secret admin2pa55 Step 2: Verify the TACACS+ Server configuration. Click the TACACS+ Server. On the Services tab, click AAA. Notice that there is a Network configuration entry for R2 and a User Setup entry for Admin2. Step 3: Configure the TACACS+ server specifics on R2. Configure the AAA TACACS server IP address and secret key on R2. Note: The commands tacacs-server host and tacacs-server key are deprecated. Currently, Packet Tracer does not support the new command tacacs server. R2(config)# tacacs-server host 192.168.2.2 R2(config)# tacacs-server key tacacspa55 Step 4: Configure AAA login authentication for console access on R2. Enable AAA on R2 and configure all logins to authenticate using the AAA TACACS+ server. If it is not available, then use the local database. R2(config)# aaa new-model R2(config)# aaa authentication login default group tacacs+ local Step 5: Configure the line console to use the defined AAA authentication method. Configure AAA authentication for console login to use the default AAA authentication method. R2(config)# line console 0 R2(config-line)# login authentication default

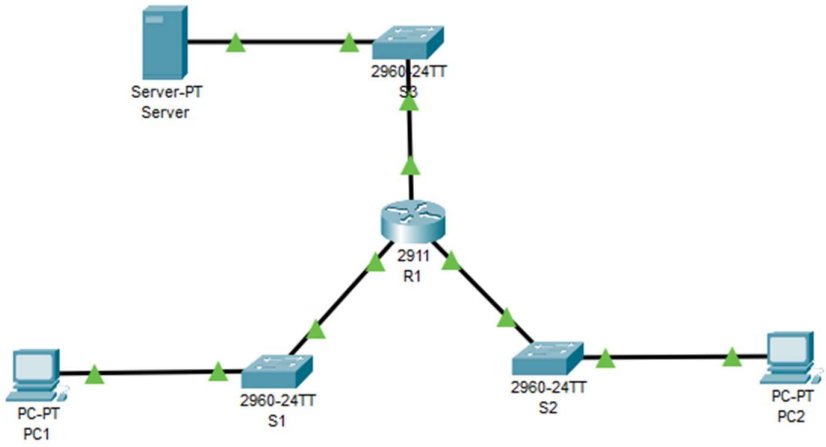
Step 6: Verify the AAA authentication method. Verify the user EXEC login using the AAA TACACS+ server.
R2(config-line)# end

%SYS-5-CONFIG_I: Configured from console by console
R2# exit
User Access Verification
Username: Admin2
Password: admin2pa55
R2>

Part 4: Configure Server-Based AAA Authentication Using RADIUS on R3
Step 1: Configure a backup local database entry called Admin. For backup purposes, configure a local username of Admin3 and a secret password of admin3pa55.
R3(config)# username Admin3 secret admin3pa55
Step 2: Verify the RADIUS Server configuration. Click the RADIUS Server. On the Services tab, click AAA. Notice that there is a Network configuration entry for R3 and a User Setup entry for Admin3.
Step 3: Configure the RADIUS server specifics on R3. Configure the AAA RADIUS server IP address and secret key on R3. Note: The commands radius-server host and radius-server key are deprecated. Currently Packet Tracer does not support the new command radius server.
R3(config)# radius-server host 192.168.3.2
R3(config)# radius-server key radiuspa55
Step 4: Configure AAA login authentication for console access on R3. Enable AAA on R3 and configure all logins to authenticate using the AAA RADIUS server. If it is not available, then use the local database.
R3(config)# aaa new-model
R3(config)# aaa authentication login default group radius local
Step 5: Configure the line console to use the defined AAA authentication method. Configure AAA authentication for console login to use the default AAA authentication method.
R3(config)# line console 0 Page 5 of 7 Configure AAA Authentication on Cisco Routers R3(config-line)# login authentication default
Step 6: Verify the AAA authentication method. Verify the user EXEC login using the AAA RADIUS server.
R3(config-line)# end %SYS-5-CONFIG_I: Configured from console by console
R3# exit R3 con0 is now available Press RETURN to get started. ********** AUTHORIZED ACCESS ONLY ********** UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.
User Access Verification Username: Admin3 Password: admin3pa55
R3>
Step 7: Check results. Your completion percentage should be 100%. Click Check Results to see feedback and verification of which required components have been completed.

Roll No: A002 **Name :Vidya Bhandari**

	Practical 3 configure extended ACLs	Date:07/02/2025
a	Configure, Apply and Verify an Extended Numbered ACL(Scenario-1)	



Configure Router:

Step 1: Configure password for vty lines

```
R1(config)#line vty 0 4
R1(config-line)#password vtypa55
R1(config-line)#login
R1(config-line)#exit
```

Step 2: Configure secret on router

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

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

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

a. We are only permitting FTP traffic; therefore, enter the eq keyword, followed by a question mark to display the available options. Then, enter ftp and press Enter.

R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ftp

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

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

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

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

```
R1(config)#int gig0/0
R1(config-if)#ip access-group 100 in
```

Step 3: Verify the ACL implementation.

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

b. FTP from PC1 to Server. The username and password are both cisco. PC> ftp 172.22.34.62

c. Exit the FTP service of the Server.

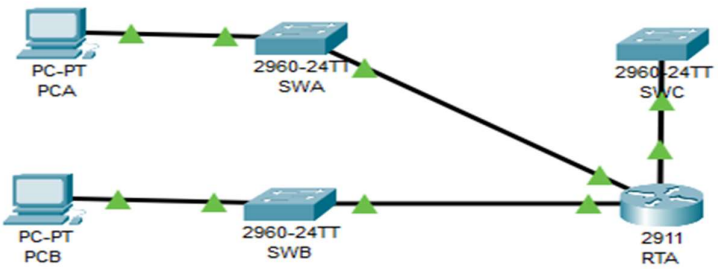
ftp> quit

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

```
C:\>ftp 172.22.34.62
Trying to connect...172.22.34.62
Connected to 172.22.34.62
220- Welcome to PT Ftp server
Username:cisco
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>quit

221- Service closing control connection.
C:\>
```

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. The ping should be successful, if the ping is unsuccessful, verify the IP addresses before continuing.
b. FTP from PC2 to Server. The connection should fail.
c. Open the web browser on PC2 and enter the IP address of Server as the URL. The connection should be successful.

Configuring Extended ACLs - Scenario 2



Step 1:Configure the ip address on switch.

```
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)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up

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

Step 2: configure the secret on router and switch. And also configure console password on router and switch.

```
SWA(config)#
SWA(config)#enable secret enpa55
SWA(config)#line console 0
SWA(config-line)#password tyit
SWA(config-line)#login
SWA(config-line)#
```

Step 3:test connectivity.

Ping from PCA to PCB.

```
C:\>ping 10.101.117.35

Pinging 10.101.117.35 with 32 bytes of data:

Reply from 10.101.117.35: bytes=32 time<1ms TTL=127
Reply from 10.101.117.35: bytes=32 time=1ms TTL=127
Reply from 10.101.117.35: bytes=32 time=2ms TTL=127
Reply from 10.101.117.35: bytes=32 time<1ms TTL=127

Ping statistics for 10.101.117.35:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

Ping from PCA to SWC.

```
C:\>ping 10.101.117.2

Pinging 10.101.117.2 with 32 bytes of data:

Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.101.117.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping from PCB to SWC.

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.101.117.2

Pinging 10.101.117.2 with 32 bytes of data:

Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time=1ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time=1ms TTL=254

Ping statistics for 10.101.117.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Part 1: Configure switch and router to support SSH connection

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

```
Enter configuration commands, one per line.  E
SWA(config)#ip domain-name ccnasecurity.com
SWA(config)#
```

Step 2:Configure users to login to SSH

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

Step 3:Configure incoming vty lines.

```
SWA(config)#line vty 04
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]
```

Step 4:Verify the SSH connection

ssh -1 Admin1 10.101.117.34

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
RTA(config-if)#
```

Step 3: : Verify the extended ACL implementation.

a. Ping from PCB to all of the other IP addresses in the network.

```
C:\>ping 10.101.117.51

Pinging 10.101.117.51 with 32 bytes of data:

Reply from 10.101.117.51: bytes=32 time=1ms TTL=127
Reply from 10.101.117.51: bytes=32 time<1ms TTL=127
Reply from 10.101.117.51: bytes=32 time<1ms TTL=127
Reply from 10.101.117.51: bytes=32 time<1ms TTL=127

Ping statistics for 10.101.117.51:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
C:\>ping 10.101.117.2

Pinging 10.101.117.2 with 32 bytes of data:

Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time=1ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.101.117.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

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

```
C:\> ssh -l Admin 10.101.117.2

Password:

Password:


SWC>
```

c. Exit the SSH session to SWC.

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

```
C:\>ping 10.101.117.35

Pinging 10.101.117.35 with 32 bytes of data:

Reply from 10.101.117.35: bytes=32 time<1ms TTL=127
Reply from 10.101.117.35: bytes=32 time=1ms TTL=127
Reply from 10.101.117.35: bytes=32 time=1ms TTL=127
Reply from 10.101.117.35: bytes=32 time=2ms TTL=127

Ping statistics for 10.101.117.35:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 1ms

C:\>ping 10.101.117.2

Pinging 10.101.117.2 with 32 bytes of data:

Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time=2ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254
Reply from 10.101.117.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.101.117.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

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

f. SSH from PCA to SWB. The username is Admin, and the password is Adminpa55.

```
C:\>ssh -l Admin 10.101.117.34

Password:

Password:


SWB>
```

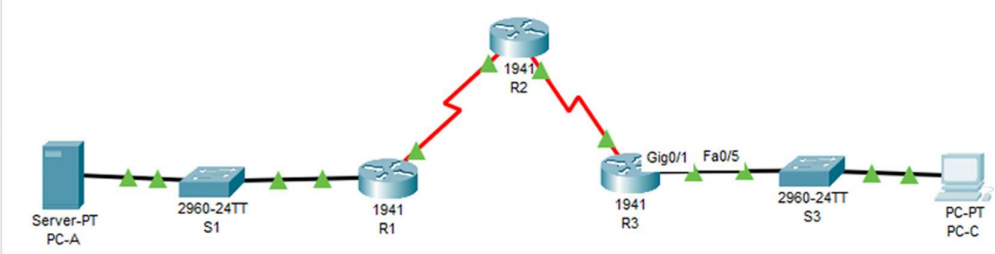
g. After logging into SWB, do not log out. SSH to SWC in privileged EXEC mode.

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


SWC>
```

Roll No: A002 **Name :Vidya Bhandari**

	Practical 4 configure IP ACLs mitigate attacks	Date:14/02/2025

a Configure IP ACLs to Mitigate Attacks



Addressing Table

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

Configure enable secret password on all routers:
enable secret enpa55

Configure console password on all routers:

```
R1(config)#enable secret enpa55
R1(config)#line console 0
R1(config-line)#password conpa55
R1(config-line)#login
R1(config-line)#exit
```

Configure domain name, username and line vty on all routers:

```
R2(config)#ip domain-name ccnasecurity.co
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)#exit
```

Crypto key
R2(config)#crypto key generate rsa

Loopback address on R2

```
R2(config)#interface Loopback 0

R2(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state t

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopb

R2(config-if)#ip address 192.168.2.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)# show ip interface brief
          ^
% Invalid input detected at '^' marker.

R2(config)# do show ip interface brief
```

Configure static routing on Routers

```
R1(config)#ip route 192.168.3.0 255.255.255.0 10.1.1.2
*Mar 1 0:59:30.542: %SSH-5-ENABLED: SSH 1.99 has been enabled
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)#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)#
```

```
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 10.1.1.0 255.255.255.252 10.2.2.2
R3(config)#
```

Part 1: Verify Basic Network Connectivity

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

```
C:\>ping 192.168.3.3

Pinging 192.168.3.3 with 32 bytes of data:

Reply from 192.168.3.3: bytes=32 time=16ms TTL=125
Reply from 192.168.3.3: bytes=32 time=3ms TTL=125
Reply from 192.168.3.3: bytes=32 time=10ms TTL=125
Reply from 192.168.3.3: bytes=32 time=10ms TTL=125

Ping statistics for 192.168.3.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 16ms, Average = 9ms

C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=2ms TTL=254

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\>ssh -l admin 192.168.2.1

Password:



R2>
R2>
R2>
R2>exit

[Connection to 192.168.2.1 closed by foreign host]
C:\>
```

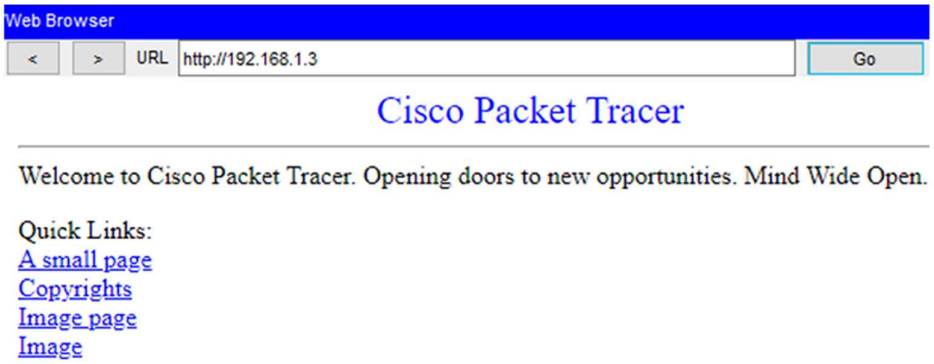
Password: adminpa55

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

```
C:\>ping  192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=2ms TTL=125
Reply from 192.168.1.3: bytes=32 time=10ms TTL=125
Reply from 192.168.1.3: bytes=32 time=10ms TTL=125
Reply from 192.168.1.3: bytes=32 time=3ms TTL=125

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 10ms, Average = 6ms

C:\>ping  192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=14ms TTL=254
Reply from 192.168.2.1: bytes=32 time=2ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 14ms, Average = 4ms

C:\>ssh -l admin 192.168.2.1

Password:



R2>
R2>exit

[Connection to 192.168.2.1 closed by foreign host]
```

Password :adminpa55

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



Part 2: Secure Access to Routers

Step 1: Configure ACL and line vty on R1, R2, and R3.

```
R1(config)#access-list 10 permit host 192.168.3.3
R1(config)#line vty 0 4
R1(config-line)#access-class 10 in
R1(config-line)#exit
R1(config)#
```

Step 2:Verify access from PC-C

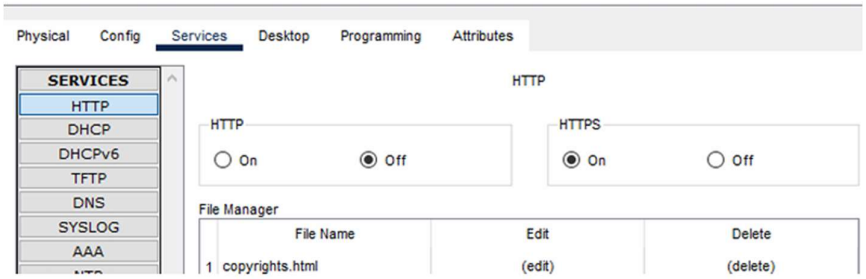
```
C:\>ssh -l admin 192.168.2.1

Password:



R2>
R2>exit

[Connection to 192.168.2.1 closed by foreign host]
C:\>
```

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

```
C:\>ssh -l admin 192.168.2.1

% Connection refused by remote host
C:\>
```

Part 3: 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.



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 any host 192.168.3.3 host 10.1.1.1 eq 22
                                                             ^
% Invalid input detected at '^' marker.

R1(config)#access-list 120 permit tcp host 192.168.3.3 host 10.1.1.1 eq 22
```

Step 3: Apply the ACL to interface S0/0/0.

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

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

Part 4: Modify an Existing ACL on R1

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.

```
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.

```
C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=2ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254
Reply from 192.168.2.1: bytes=32 time=1ms TTL=254

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

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.
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.

```
R3(config)#int gig0/1
R3(config-if)#ip access-group 110 in
R3(config-if)#exit
```

Part 6: 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 192.168.3.1 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 0.255.255.255 any
R3(config)#access-list 100 deny ip 192.168.0.0 0.255.255.255 any
R3(config)#access-list 100 deny ip 127.0.0.0 0.255.255.255 any
R3(config)#access-list 100 deny ip 224.0.0.0 15.255.255.255 any
R3(config)#access-list 100 permit ip any any
```

Step 2: Apply the ACL to interface Serial 0/0/1

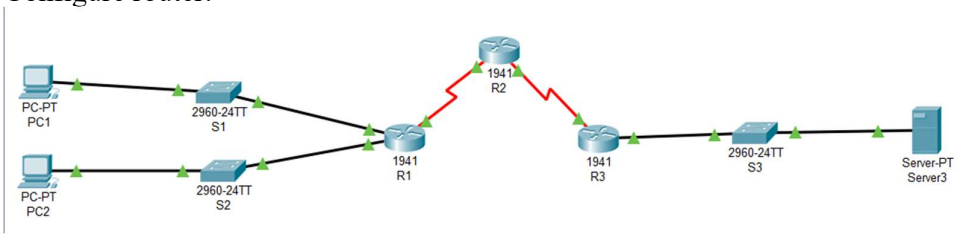
```
R3(config)#int se0/0/1
R3(config-if)#ip access-group 100 in
```

Step 3: Confirm that the specified traffic entering interface Serial 0/0/1 is handled correctly.
a. From the PC-C command prompt, ping the PC-A server. The ICMP echo replies are blocked by the ACL since they are sourced from the 192.168.0.0/16 address space.
b. Establish an SSH session to 192.168.2.1 from PC-C (should be successful).
Step 4: Check results.
Your completion percentage should be 100%. Click Check Results to see feedback and verification of which required components have been completed.

Roll No: A002 **Name :Vidya Bhandari**

Practical no 5: configure IPV6 ACLs	Date:02/02/2025

Configure router:



Configure, Apply, and Verify an IPv6 ACL
Router (config)#hostname R1
R1 (config)#ipv6 unicast-routing
R1 (config)#int g0/0
R1 (config-if)#ipv6 enable
R1 (config-if)#no shut

```
R1(config)#ipv6 unicast-routing
R1(config)#int g0/0
R1(config-if)#ipv6 enable
R1(config-if)#no shut

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

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

R1 (config)#int g0/1
R1 (config-if)#ipv6 enable
R1 (config-if)#no shut

```
R1(config)#int g0/1
R1(config-if)#ipv6 enable
R1(config-if)#no shut

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

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

R1 (config)#Int se 0/0/0
R1 (config-if)#ipv6 enable
R1 (config-if)#no shut

```
R1(config-if)#int se 0/0/0
R1(config-if)#ipv6 enable
R1(config-if)#no shut

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#
R1(config-if)#ipv6 address 2001:DB8:1:20::1/64
R1(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

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

```
R1(config)#ipv6 route 2001:DB8:1:25::/64 2001:DB8:1:20::2
R1(config)#ipv6 route 2001:DB8:1:30::/64 2001:DB8:1:20::2
R1(config)#
```

Router (config)#hostname R2
R2 (config-if)#ipv6 unicast-routing
R2 (config-if)#int s0/0/0
R2 (config-if)#no shut

```
R2(config)#ipv6 unicast-routing
R2(config)#int se 0/0/0
R2(config-if)#ipv6 enable
R2(config-if)#no shut

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

R2(config-if)#ipv6 address
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up

% Incomplete command.
R2(config-if)#ipv6 address 2001:DB8:1:20::2/64
```

R2 (config)#Int se 0/0/1
R2 (config-if)#ipv6 enable
R2 (config-if)#no shut

```
R2(config-if)#int se 0/0/1
R2(config-if)#ipv6 enable
R2(config-if)#no shut

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

R2(config-if)#i[
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/1, changed state to up

               ^
% Invalid input detected at '^' marker.

R2(config-if)#ipv6 address 2001:DB8:1:25::2/6
%Serial0/0/1: Error: 2000::/6 is overlapping with 2001:DB8:1:20::/64 on Serial0/0/0
R2(config-if)#ipv6 address 2001:DB8:1:25::2/64
```

```
R2(config)#ipv6 route 2001:DB8:1:10::/64 2001:DB8:1:20::1
R2(config)#ipv6 route 2001:DB8:1:11::/64 2001:DB8:1:20::1
R2(config)#ipv6 route 2001:DB8:1:30::/64 2001:DB8:1:25::1
R2(config)#
```

Router (config)#hostname R3
R3 (config)#ipv6 unicast-routing
R3 (config)#int g0/1
R3 (config)#no shut

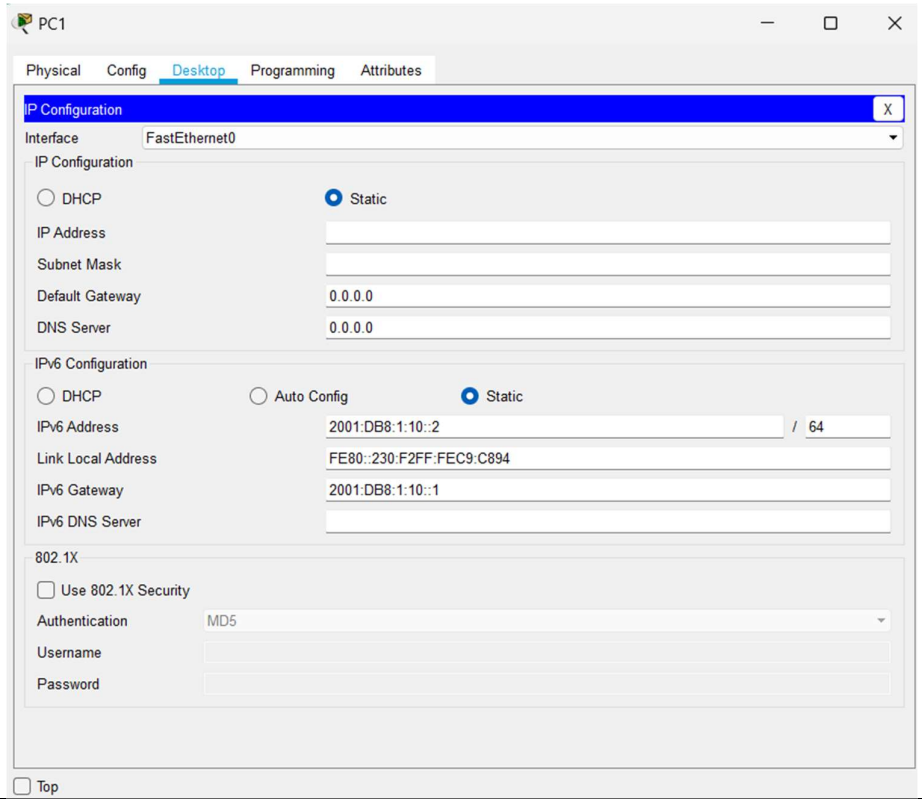
```
R3(config)#ipv6 unicast-routing
R3(config)#int g0/1
R3(config-if)#ipv6 enable
R3(config-if)#no shut

R3(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

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

R3 (config)#Int se 0/0/1
R3 (config-if)#ipv6 enable
R3 (config-if)#no shut

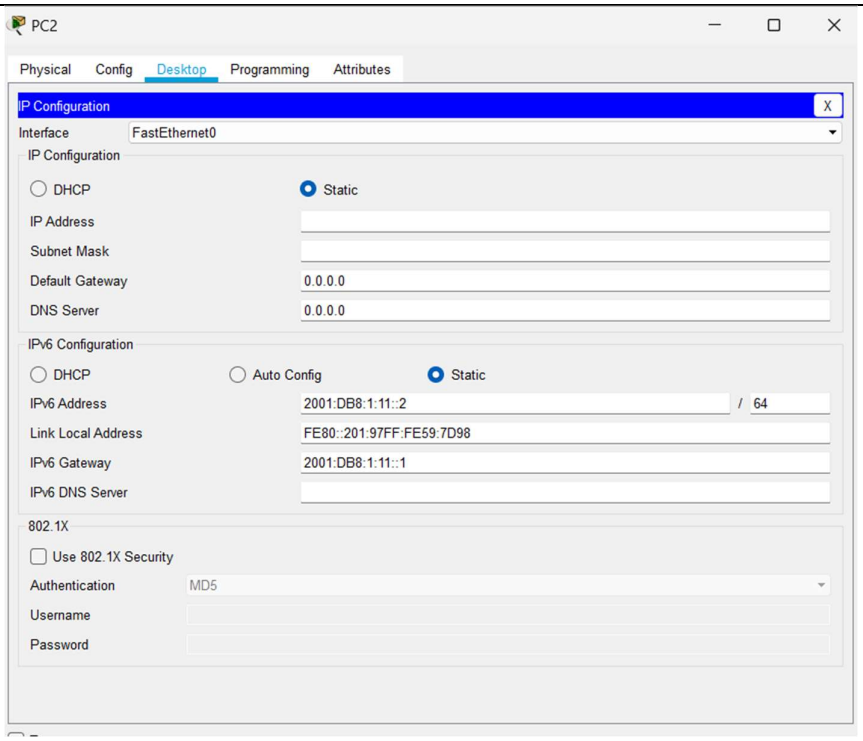
```
R3(config)#int se 0/0/1
R3(config-if)#ipv6 enable
R3(config-if)#no shut

%LINK-5-CHANGED: Interface Serial0/0/1, changed state to down
R3(config-if)#ipv6 address 2001:DB8:1:25::1/64
R3(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/1, changed state to up

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

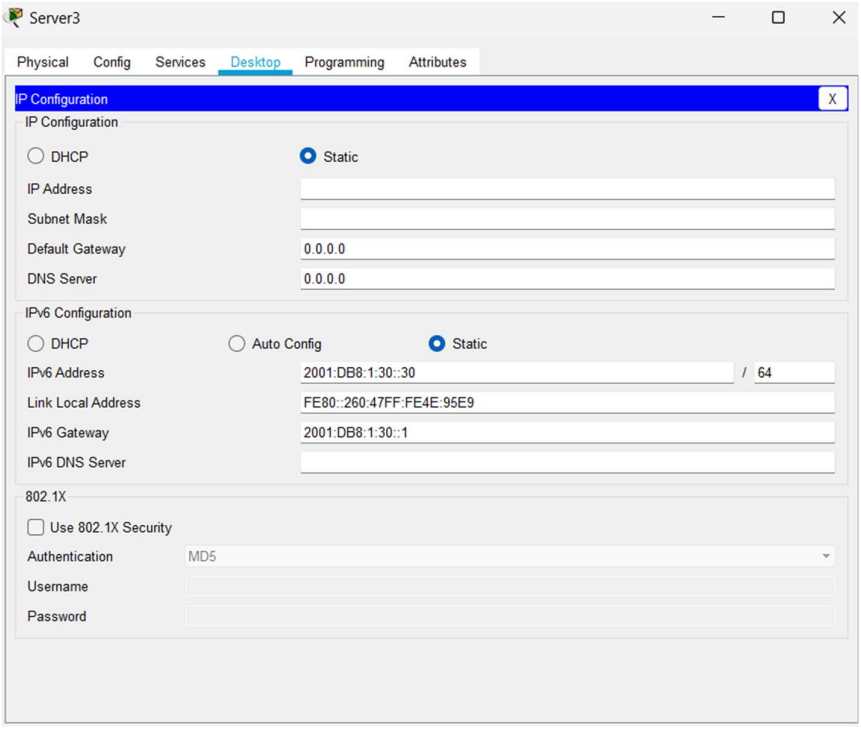
```
R3(config)#ipv6 route 2001:DB8:1:20::/64 2001:DB8:1:25::2
R3(config)#ipv6 route 2001:DB8:1:10::/64 2001:DB8:1:25::2
R3(config)#ipv6 route 2001:DB8:1:11::/64 2001:DB8:1:25::2
```

Then put IPV6 address in pc1 and pc2 and server3



Roll No: A002 **Name :Vidya Bhandari**





Add table for ip address of Pcs and server
Part 1: Configure, Apply, and Verify an IPv6 ACL
Step 1: Configure an ACL that will block HTTP and HTTPS access.
a. Block HTTP and HTTPS traffic from reaching Server3.
R1(config)# deny tcp any host 2001:DB8:1:30::30 eq www
R1(config)# deny tcp any host 2001:DB8:1:30::30 eq 443

b. Allow all other IPv6 traffic to pass.
R1(config)# permit ipv6 any any

Step 2: Apply the ACL to the correct interface. Apply the ACL on the
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ipv6 traffic-filter BLOCK_HTTP in

```
R1(config)#ipv6 ?
  access-list      Configure access lists
  cef              Cisco Express Forwarding
  dhcp             Configure Ipv6 DHCP
  general-prefix   Configure a general IPv6 prefix
  host             Configure static hostnames
  local            Specify local options
  nat              NAT-PT Configuration commands
  neighbor         Neighbor
  route            Configure static routes
  router           Enable an IPV6 routing process
  unicast-routing  Enable unicast routing
R1(config)#ipv6 access-list ?
  WORD  User selected string identifying this access list
R1(config)#ipv6 access-list BLOCK_HTTP
R1(config-ipv6-acl)#deny tcp?
tcp
R1(config-ipv6-acl)#deny tcp any
% Incomplete command.
R1(config-ipv6-acl)#deny tcp any host 2001:DB8:1:30::30 ?
  eq           Match only packets on a given port number
  established  Match established connections
  gt           Match only packets with a greater port number
  lt           Match only packets with a lower port number
  neq          Match only packets not on a given port number
  range        Match only packets in the range of port numbers
  <cr>
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
R1(config)#int g0/1
R1(config-if)#ipv6 ?
  address          Configure IPv6 address on interface
  authentication   authentication subcommands
  dhcp             IPv6 DHCP interface subcommands
  eigrp            Configure EIGRP IPv6 on interface
  enable           Enable IPv6 on interface
  flow             NetFlow Related commands
  hello-interval   Configures IP-EIGRP hello interval
  mtu              Set IPv6 Maximum Transmission Unit
  nat              Enable IPv6 NAT on interface
  nd               IPv6 interface Neighbor Discovery subcommands
  ospf             OSPF interface commands
  rip              Configure RIP routing protocol
  summary-address  Summary prefix
  traffic-filter   Access control list for packets
  unnumbered       Preferred interface for source address selection
R1(config-if)#ipv6 traffic-filter BLOCK_HTTP in
```

Configure, Apply, and Verify a Second IPv6 ACL
Step 1: Create an access list to block ICMP.
a. Block all ICMP traffic from any hosts to any destination.
R3(config)# deny icmp any any
b. Allow all other IPv6 traffic to pass.
R3(config)# permit ipv6 any any
Step 2: Apply the ACL to the correct interface.
R3(config)# interface GigabitEthernet0/0
R3(config-if)# ipv6 traffic-filter BLOCK_ICMP out

```
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
R3(config)#int g0/1
R3(config-if)#ipv6 traffic-filter BLOCK_ICMP out
```

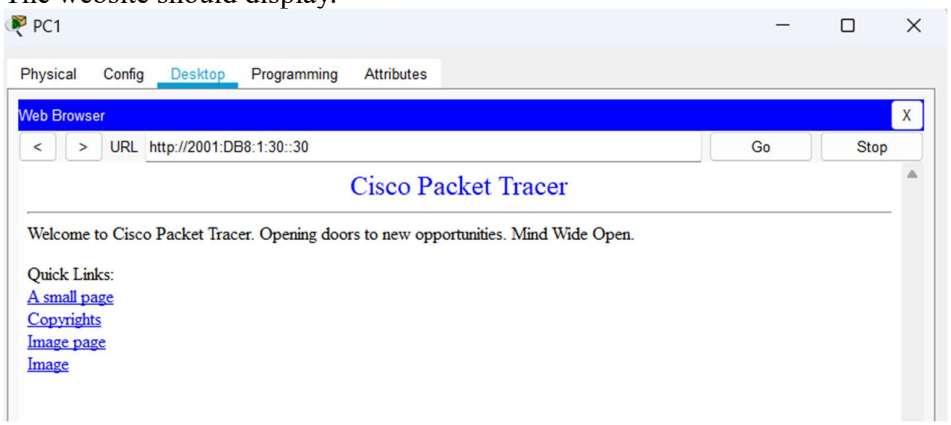
a. Ping from PC2 to 2001:DB8:1:30::30. The ping should fail.

```
Pinging 2001:DB8:1:30::30 with 32 bytes of data:

Reply from 2001:DB8:1:25::1: Destination host unreachable.
Reply from 2001:DB8:1:25::1: Destination host unreachable.
Reply from 2001:DB8:1:25::1: Destination host unreachable.
Reply from 2001:DB8:1:25::1: Destination host unreachable.

Ping statistics for 2001:DB8:1:30::30:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

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

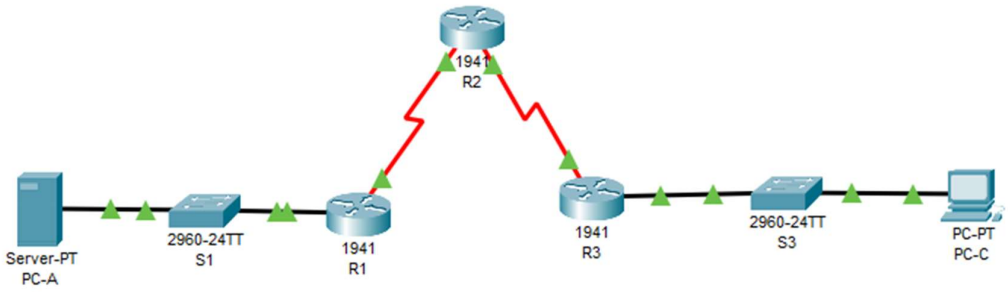
```
Pinging 2001:DB8:1:30::30 with 32 bytes of data:

Reply from 2001:DB8:1:25::1: Destination host unreachable.
Reply from 2001:DB8:1:25::1: Destination host unreachable.
Reply from 2001:DB8:1:25::1: Destination host unreachable.
Reply from 2001:DB8:1:25::1: Destination host unreachable.

Ping statistics for 2001:DB8:1:30::30:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

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



Roll No: A002 **Name :Vidya Bhandari**

Practical 6:Configure a Zone based policy firewall	Date:27/02/2025



Addressing Table

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

The routers have been pre-configured with the following:
Configure Console password:conpa55

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#line console 0
R1(config-line)#password conpa55
R1(config-line)#login
R1(config-line)#
```

Do this command in all routers
Configure line vty password:vtypa55

```
R1(config)#
R1(config)#line vty 0 4
R1(config-line)#password vtypa55
R1(config-line)#login
R1(config-line)#exit
R1(config)#
R1(config)#
```

Do this command in all routers
Configure Enable password:enpa55

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

Do this command in all routers
Configure SSH on routers

```
R1(config)#
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)#exit
R1(config)#
```

Do this command in all routers

```
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)#
```

Do this command in all routers

Configure Static Routing

```
R1(config)#ip route 10.2.2.0 255.255.255.252 10.1.1.2
R1(config)#
R1(config)#
R1(config)#
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
R2(config)#
```

```
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
R3(config)#
```

Part 1: Verify Basic Network Connectivity

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

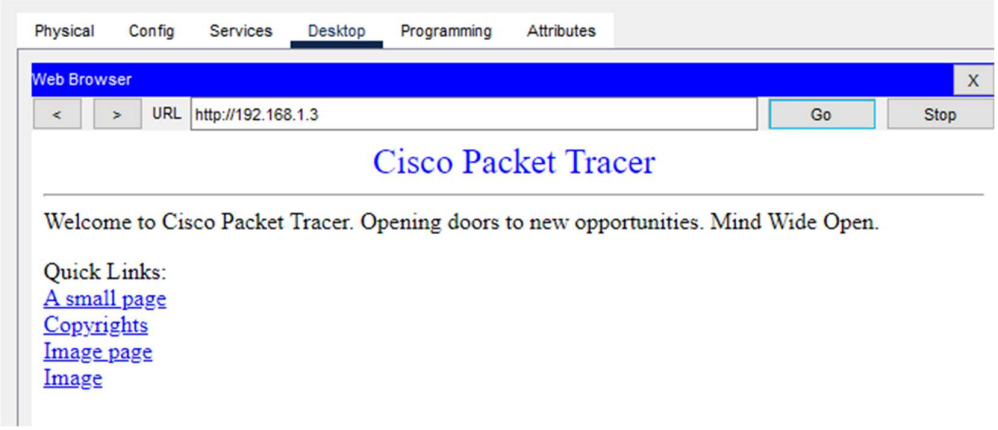
```
C:\>ping 192.168.3.3

Pinging 192.168.3.3 with 32 bytes of data:

Reply from 192.168.3.3: bytes=32 time=16ms TTL=125
Reply from 192.168.3.3: bytes=32 time=11ms TTL=125
Reply from 192.168.3.3: bytes=32 time=3ms TTL=125
Reply from 192.168.3.3: bytes=32 time=26ms TTL=125

Ping statistics for 192.168.3.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 26ms, Average = 14ms

C:\>
```

Step 2: Access R2 using SSH.

```
C:\>ssh -ladmin 10.2.2.2
Invalid Command.

C:\>ssh -l admin 10.2.2.2

Password:



R2>exit

[Connection to 10.2.2.2 closed by foreign host]
C:\>
```

Step 3: From PC-C, open a web browser to the PC-A server. a. Click the Desktop tab and then click the Web Browser application. Enter the PC-A IP address 192.168.1.3 as

the URL. The Packet Tracer welcome page from the web server should be displayed. b. Close the browser on PC-C.



Part 2: Create the Firewall Zones on R3

Step 1: Enable the Security Technology package. a. On R3, issue the show version command to view the Technology Package license information.

R3(config)#license boot module c1900 technology-package securityk9
ACCEPT? [yes/no]: yes
R3(config)#exit
R3#copy run start

R1#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
R1#reload
Proceed with reload? [confirm](press enter)
(Again)

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

R1>en
Password:
R1#show version

Step 2: Create an internal zone. Use the zone security command to create a zone named IN-ZONE.

R3(config)# zone security IN-ZONE
R3(config-sec-zone) exit

Step 3: Create an external zone. Use the zone security command to create a zone named OUT-ZONE.

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

Part 3: Identify Traffic Using a Class-Map

Step 1: Create an ACL that defines internal traffic. Use the access-list command to create extended ACL 101 to permit all IP protocols from the 192.168.3.0/24 source network to any destination.

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

Step 2: Create a class map referencing the internal traffic ACL. Use the class-map type inspect command with the match-all option to create a class map named IN NETCLASS-MAP. Use the match access-group command to match ACL 101.

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

Part 4: Specify Firewall Policies
Step 1: Create a policy map to determine what to do with matched traffic. Use the policy-map type inspect command and create a policy map named IN-2-OUT-PMAP.
R3(config)# policy-map type inspect IN-2-OUT-PMAP
Step 2: Specify a class type of inspect and reference class map IN-NET-CLASS-MAP.
R3(config-pmap)# class type inspect IN-NET-CLASS-MAP
Step 3: Specify the action of inspect for this policy map. The use of the inspect command invokes context-based access control (other options include pass and drop).
R3(config-pmap-c)# inspect
R3(config-pmap-c)#exit
R3(config-pmap)#exit

Part 5: Apply Firewall Policies
Step 1: Create a pair of zones. Using the zone-pair security command, create a zone pair named IN-2-OUT-ZPAIR. Specify the source and destination zones that were created in Task 1. R3(config)# zone-pair security IN-2-OUT-ZPAIR source IN-ZONE destination OUT-ZONE
Step 2: Specify the policy map for handling the traffic between the two zones. Attach a policy-map and its associated actions to the zone pair using the service-policy type inspect command and reference the policy map previously created, IN-2-OUT-PMAP.
R3(config-sec-zone-pair)# service-policy type inspect IN-2-OUT-PMAP
R3(config-sec-zone-pair)# exit
R3(config)# Step 3: Assign interfaces to the appropriate security zones.
Configuring a Zone-Based Policy Firewall (ZPF) 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
R3(config)#exit
R3#copy run start
R3#reload
Step 4: Copy the running configuration to the startup configuration.

Part 6: Test Firewall Functionality from IN-ZONE to OUT-ZONE Verify that internal hosts can still access external resources after configuring the ZPF.
Step 1: From internal PC-C, ping the external PC-A server. From the PC-C command prompt, ping PC-A at 192.168.1.3. The ping should succeed.
Step 2: From internal PC-C, SSH to the R2 S0/0/1 interface.

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

ssh –l admin 10.2.2.2

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

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

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

Step 4: From internal PC-C, open a web browser to the PC-A server web page. Enter the server IP address 192.168.1.3 in the browser URL field, and click Go. The HTTP session should succeed. While the HTTP session is active, issue the command show policy-map type inspect zone-pair sessions on R3 to view established sessions

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.

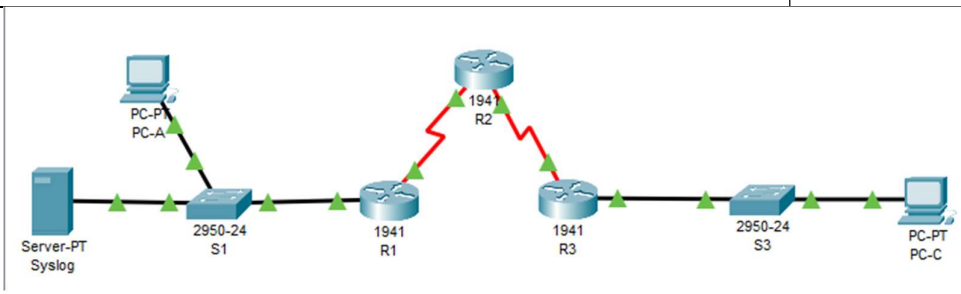
Step 2: From R2, ping PC-C. From R2, ping PC-C at 192.168.3.3. The ping should fail.

Step 3: Check results.

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

Roll No: A002 **Name :Vidya Bhandari**

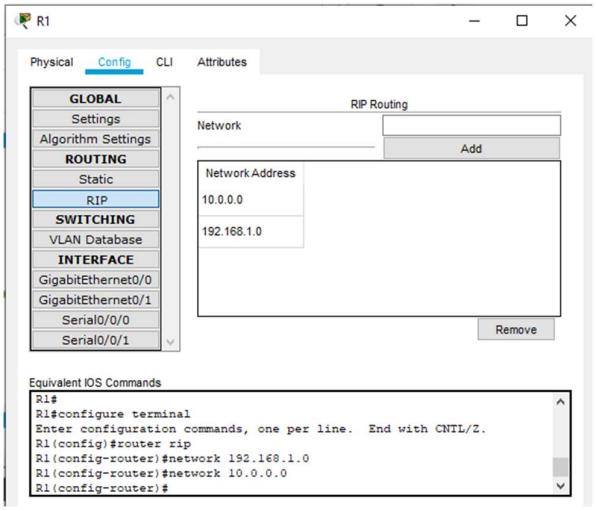
	Practical 7 Configuring IOS intrusion prevention system using the CLI	Date:01/03/2025



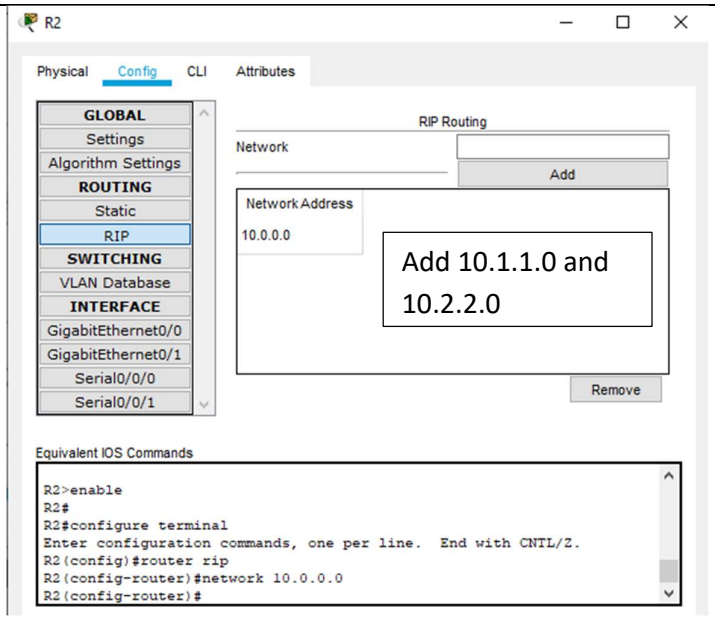
Addressing Table

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

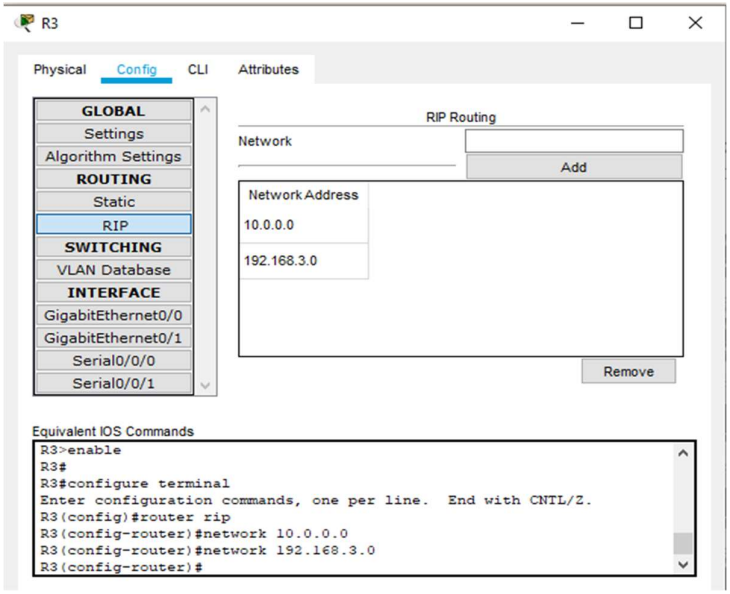
RIP:
Step 1: Go to R1 ,R2 or R3 and do RIP routing.
For R1



For R2

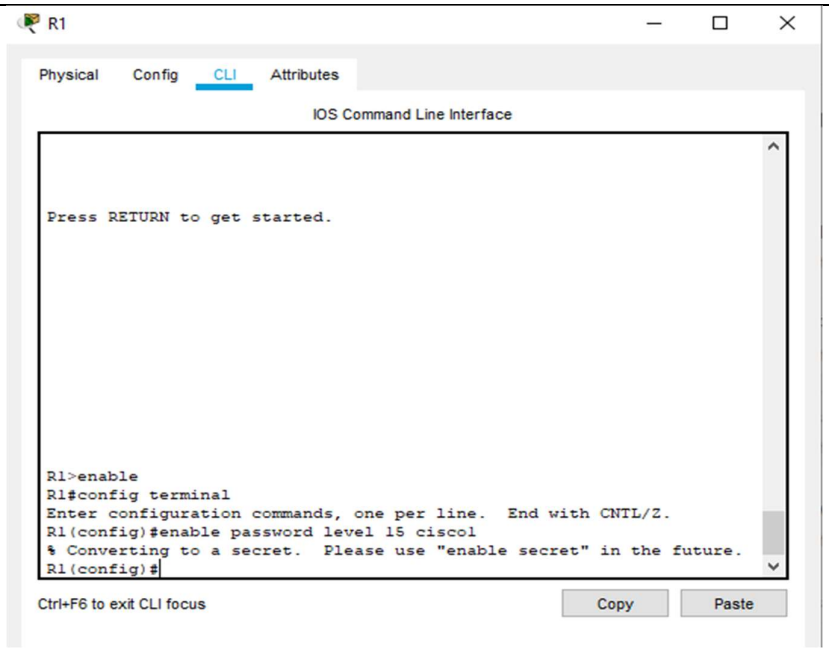


For R3



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

Step 2: Enable Password in all 3 routers.

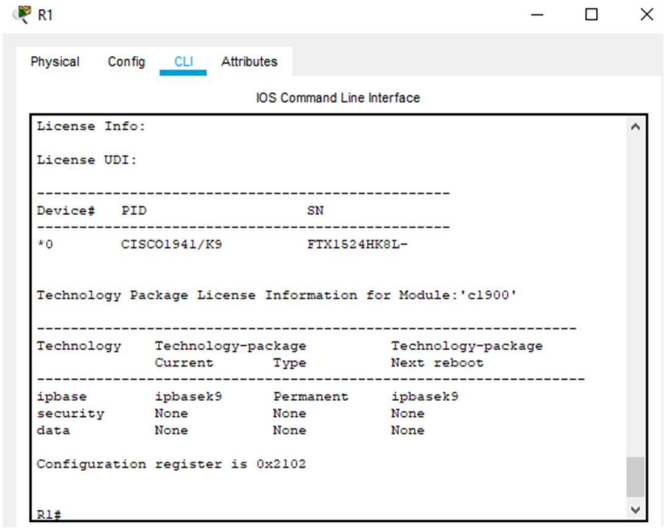


Do this commands for all routers.

Part 1: Enable IOS IPS

Step 1: Enable the Security Technology package.

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



b. If the Security Technology package has not been enabled, use the following command to enable the package.
R1(config)# license boot module c1900 technology-package securityk9

c. Accept the end user license agreement.
ACCEPT? [yes/no]: y

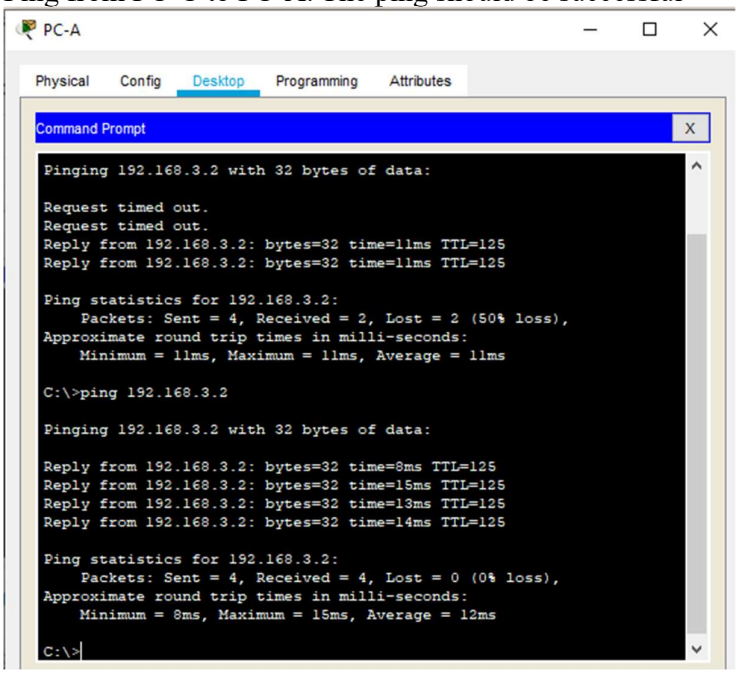
d. Save the running-config and reload the router to enable the security license.
R1(config)#: %IOS_LICENSE_IMAGE_APPLICATION-6-LICENSE_LEVEL: Module name = C1900 Next reboot level = securityk9 and License = securityk9
(For the command execute follow the following steps)
R1#copy run start
Destination filename [startup-config]?

Building configuration...
[OK]
R1#reload
Proceed with reload? [confirm](press enter)
(Again)

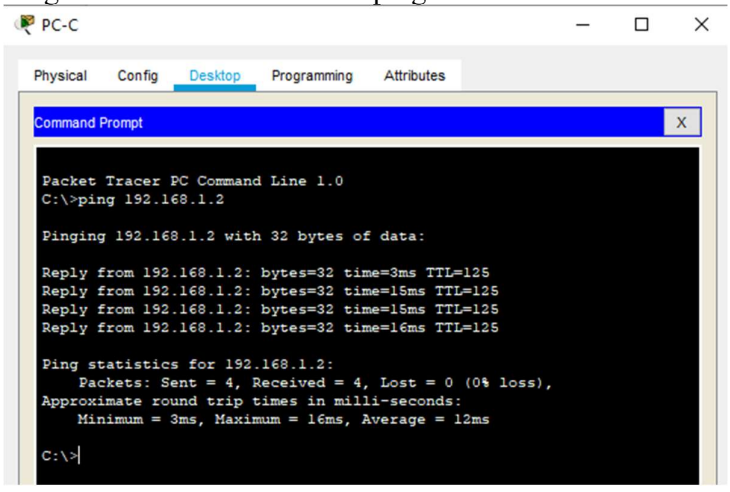
e. Verify that the Security Technology package has been enabled by using the show version command.
R1>enable
Password:
R1#show version
(now the packages are install and we are able to see them)

Step 2: Verify network connectivity.

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



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



Step 3:Create an IOS IPS configuration directory in flash. On R1, create a directory in flash using the mkdir command. Name the directory ipsdir.
R1# mkdir ipsdir
Create directory filename [ipsdir]? <Enter> Created dir flash:ipsdir
Step 4: Configure the IPS signature storage location. On R1, configure the IPS signature storage location to be the directory you just created.
R1(config)# ip ips config location flash:ipsdir

Step 5: Create an IPS rule.
On R1, create an IPS rule name using the ip ips name *name* command in global configuration mode. Name the IPS rule iosips.
R1(config)# ip ips name iosips

Step 6: Enable logging.
IOS IPS supports the use of syslog to send event notification. Syslog notification is enabled by default. If logging console is enabled, IPS syslog messages display.
a. Enable syslog if it is not enabled.
R1(config)# ip ips notify log

b. If necessary, use the clock set command from privileged EXEC mode to reset the clock. R1# clock set 10:20:00 10 january 2014 (We can skip this step)

c. Verify that the timestamp service for logging is enabled on the router using the show run command.
Enable the timestamp service if it is not enabled.
R1(config)# service timestamps log datetime msec

c. Send log messages to the syslog server at IP address 192.168.1.50. R1(config)# logging host 192.168.1.50

Step 7: Configure IOS IPS to use the signature categories.
Retire the all signature category with the retired true command (all signatures within the signature release).
Unretire the IOS_IPS Basic category with the retired false command.

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

Step 8: Apply the IPS rule to an interface.

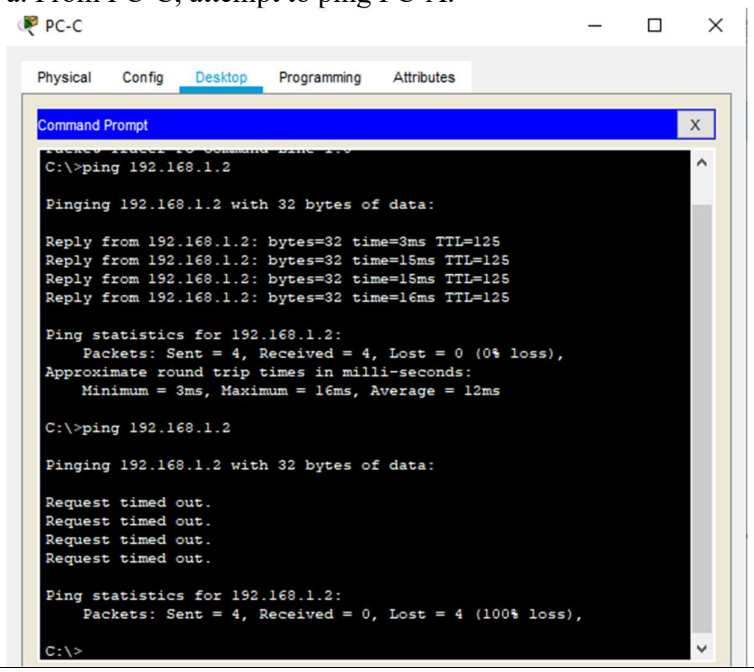
Apply the IPS rule to an interface with the ip ips name *direction* command in interface configuration mode. Apply the rule outbound on the G0/1 interface of R1. After you

enable IPS, some log messages will be sent to the console line indicating that the IPS engines are being initialized.
Note: The direction in means that IPS inspects only traffic going into the interface. Similarly, out means that IPS inspects only traffic going out of the interface.
R1(config)# interface g0/1
R1(config-if)# ip ips iosips out (Part 1)

Part 2: Modify the Signature
Step 1: Change the event-action of a signature.
Un-retire the echo request signature (signature 2004, subsig ID 0), enable it, and change the signature action to alert and drop.
R1(config)# ip ips signature-definition
R1(config-sigdef)# signature 2004 0
R1(config-sigdef-sig)# status
R1(config-sigdef-sig-status)# retired false
R1(config-sigdef-sig-status)# enabled true
R1(config-sigdef-sig-status)# exit
R1(config-sigdef-sig)# engine
R1(config-sigdef-sig-engine)# event-action produce-alert R1(config-sigdef-sig-engine)# event-action deny-packet-inline
R1(config-sigdef-sig-engine)# exit
R1(config-sigdef-sig)# exit
R1(config-sigdef)# exit
Do you want to accept these changes? [confirm] <Enter>

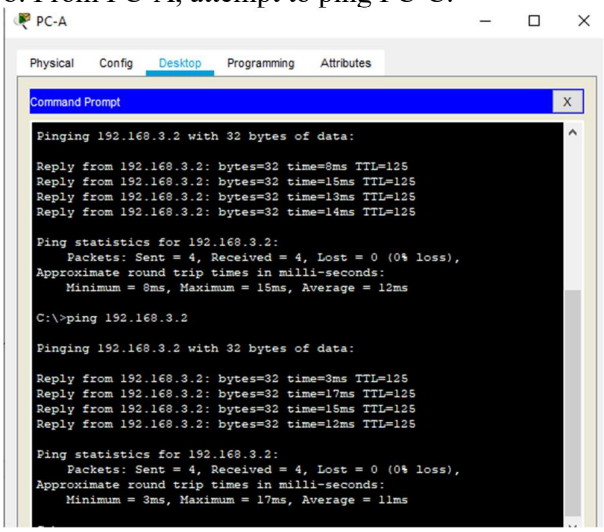
Step 2: Use show commands to verify IPS.
Use the show ip ips all command to view the IPS configuration status summary.
R1#show ip ips all

Step 3: Verify that IPS is working properly.
a. From PC-C, attempt to ping PC-A.



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

b. From PC-A, attempt to ping PC-C.



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

Step 4: View the syslog messages.
a. Click the Syslog server.
b. Select the Services tab.
c. In the left navigation menu, select SYSLOG to view the log file.