IS practical

https://docs.google.com/document/d/1cB_tDX_IKPec8mbn5IdR4ZQ5JwVQ0JyRm5rWIkXlJm8/edit?usp=sharing

Practical 1 : Configure Cisco Routers for OSPF MD5 Authentication, Syslog and NTP
Commands:

Part 1: Configure OSPF MD5 Authentication

ROUTER 1: Type the following command in the CLI mode

```
Router>enable
Router#configure terminal
Router(config)#router ospf 1
Router(config-router)#network 192.168.1.0 0.255.255.255 area 1
Router(config-router)#network 10.1.1.0 0.255.255.255 area 1
Router(config-router)#exit
Router(config)#exit
Router#
```

ROUTER 2: Type the following command in the CLI mode

```
Router>enable
Router#configure terminal
Router(config)#router ospf 1
Router(config-router)#network 10.1.1.0 0.255.255.255 area 1
Router(config-router)#network 10.2.2.0 0.255.255.255 area 1
Router(config-router)#exit
Router(config)#exit
Router#
```

ROUTER 3: Type the following command in the CLI mode

```
Router>enable
```

Router#configure
terminal
Router(config)#router ospf 1
Router(config-router)#network 192.168.3.0 0.255.255.255 area 1
Router(config-router)#network 100.2.2.0
0.255.255.255 area 1 Router(config-router)#exit
Router(config)#exit
Router#

Now we verify the connectivity by using the following
In pc1:
Ping 192.168.1.3

MD5 Authentication

ROUTER 1: Type the following command in the CLI mode
Router>enable
Router#
Router#configure terminal
Router(config)#interface Serial0/0/0
Router(config-if)#ip ospf authentication message-digest
Router(config-if)#ip ospf message-digest-key 1 md5 smile
Router(config-if)#exit
Router(config)#exit

ROUTER 2: Type the following command in the CLI mode
Router>enable
Router#
Router#configure terminal
Router(config)#interface Serial0/0/0
Router(config-if)#ip ospf authentication message-digest
Router(config-if)#ip ospf message-digest-key 1 md5 smile
Router(config-if)#exit
Router(config)#exit

Verify the MD5 Authentication using the following command in the CLI mode of Router1
Router#show ip ospf interface gigabitEthernet 0/1

We get the following output:

GigabitEthernet0/1 is up, line protocol
is up Internet address is
192.168.2.1/24, Area 1
Process ID 1, Router ID 192.168.2.1, Network Type BROADCAST,
Cost: 1 Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 192.168.3.1, Interface address 192.168.2.2
Backup Designated Router (ID) 192.168.2.1, Interface address
192.168.2.1 Timer intervals configured, Hello 10, Dead 40, Wait 40,
Retransmit 5
Hello due in 00:00:06
Index 2/2, flood queue
length 0 Next
0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 192.168.3.1 (Designated
Router) Suppress hello for 0 neighbor(s)
Message digest authentication enabled
Youngest key id is 1
MD5 Authentication has been verified

PART - 2 NTP:

Now Go to CLI Mode of Router1 and type the following commands on both the Routers

Enable
Configure terminal
Ntp server 192.168.1.5
Ntp update calendar
exit

Show clock

PART - 3 SYSLOG server :
Now Go to CLI Mode of any Router and type the following commands in all the
Enable
configure terminal
logging 192.168.1.6
Exit

Check in syslog server

Practical 2 : Configure AAA Authentication on Cisco Routers

commands:

Type the following commands in the CLI mode of the Router0

```
Router>enable
Router#configure terminal
Router(config)#aaa new-model
Router(config)#tacacs-server host 192.168.2.3 key cisco
Router(config)#radius-server host 192.168.2.2 key cisco
Router(config)#aaa authentication login ismail group tacacs+ group radius local
Router(config)#line vty 0 4
Router(config-line)#login authentication ismail
Router(config-line)#exit
Router(config)#
```

To get check the output:
The Authentication can be done by typing the command telnet 192.168.2.1 (the Router IP) in any of the PCs
We get a prompt to type the username and password, the username and password set in TACACS are entered
Username: smile
Password: smile

PRACTICAL NO 3: Configuring Extended ACLs

Commands:

Type the following commands in Router1

```
Router#configure terminal
Router(config)#access-list 100 permit tcp host 192.168.3.2 host 192.168.1.3 eq ftp
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip access-group 100 out
Router(config-if)#exit
Router(config)#
```

Now verify the ftp (ftp 192.168.1.3) command from both the PCs, one would be successful (PC0) and other (PC1) would fail

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

We use the same topology for this case

Type the following command in the CLI mode of Router1

```
Router>enable
Router#configure terminal
Router(config)#ip access-list extended SMILE
Router(config-ext-nacl)#permit tcp host 192.168.3.3 host 192.168.1.3 eq www
Router(config-ext-nacl)#exit
Router(config)#
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip access-group SMILE out
Router(config-if)#exit
Router(config)#
```

Now verify the www (192.168.1.3) command from both the PCs browser, one would be successful (PC1) and other (PC0) would fail

PRACTICAL NO 4: Configure IP ACLs to Mitigate Attacks and Configuring IPv6 ACLs

ANS:

Part 2 – Secure Access to Routers

We configure ACL 10 to block all remote access to the Routers and allow remote access only from PC. We type the following commands in all the Routers (Router0, Router1, and Router2). This part is divided in 2 subparts

Set up the SSH protocol

Enter the following commands in CLI mode of Router0

```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router0
Router0(config)#
Router0(config)#crypto key generate rsa
Router0(config)#line vty 0 4
Router0(config-line)#transport input ssh
Router0(config-line)#login local
Router0(config-line)#exit
Router0(config)#username SSHadmin privilege 15 password ismail
Router0(config)#exit
Router0#
```

Enter the following commands in CLI mode of Router1

```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router1
Router1(config)#
Router1(config)#crypto key generate rsa
Router1(config)#line vty 0 4
Router1(config-line)#transport input ssh
Router1(config-line)#login local
Router1(config-line)#exit
Router1(config)#username SSHadmin privilege 15 password ismail
Router1(config)#exit
Router1#
```

Enter the following commands in CLI mode of Router2

```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router2
Router2(config)#
Router2(config)#crypto key generate rsa
Router2(config)#line vty 0 4
```

Router2(config-line)#transport input ssh
Router2(config-line)#login local
Router2(config-line)#exit
Router2(config)#username SSHadmin privilege 15 password ismail
Router2(config)#exit
Router2#

Create an ACL 10 to permit remote access to PC only
Enter the following commands in CLI mode of all Routers
Router>enable
Router#configure terminal
Router(config)#access-list 10 permit host 192.168.4.2
Router(config)#line vty 0 4
Router(config-line)#access-class 10 in

Now we verify the remote access from PC using the following and find it
to be successful

Part 3 - Create a Numbered IP ACL 120 on R1
We need to perform the following in this part
1) Create an IP ACL numbered 120 on R1 using the following rules
2) Permit any outside host to access DNS, SMTP, and FTP services on server
3) Deny any outside host access to HTTPS services on server
4) Permit PC to access Router1 via SSH. (Done in previous part)

Enter the following commands in the CLI mode of Router1
Router1>enable
Router1#
Router1#configure terminal
Router1(config)#access-list 120 permit udp any host 192.168.1.2 eq domain
Router1(config)#access-list 120 permit tcp any host 192.168.1.2 eq smtp
Router1(config)#access-list 120 permit tcp any host 192.168.1.2 eq ftp
Router1(config)#access-list 120 deny tcp any host 192.168.1.2 eq 443
Router1(config)#exit
Router1#configure terminal
Router1(config)#interface Serial0/1/1
Router1(config-if)#ip access-group 120 in

Practical 4 b :Configuring IPv6 ACLs

For setting the ipv6 addresses we need to use the CLI mode for each Router as follows

Configuring Router0

```
Router>
Router>enable
Router#
Router#configure terminal
Router(config)#ipv6 unicast-routing

Router(config)#interface GigabitEthernet0/0
Router(config-if)#ipv6 address 2002::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#

Router(config)#interface GigabitEthernet0/1
Router(config-if)#ipv6 address 2001::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#

Router(config)#interface Serial0/1/0
Router(config-if)#ipv6 address 2003::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Configuring Router1

```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing

Router(config)#
Router(config)#interface Serial0/1/0
Router(config-if)#ipv6 address 2003::1/64
Router(config-if)#ipv6 rip a enable Router(config-if)#no shutdown
```

```
Router(config-if)#
Router(config-if)#exit
Router(config)#

Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 address 2004::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Configuring Router2

```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing

Router(config)#
Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 address 2004::2/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit

Router(config)#interface GigabitEthernet0/0
Router(config-if)#ipv6 address 2005::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

We configure the ACL and apply it to the Router1 with the following conditions

1) No HTTP or HTTPS allowed on server by any host
2) No www service accessible on the server by any host
3) Only ipv6 packets allowed towards the server

We enter the following commands in the CLI mode of the Router1 and Router2, apply it at the proper interface

```
Router>
Router>enable
Router#configure terminal
Router(config)#ipv6 access-list smile
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq www
```

Router(config-ipv6-acl)#deny tcp any host 2005::2 eq 443
Router(config-ipv6-acl)#permit ipv6 any any
Router(config-ipv6-acl)# Router(config-ipv6-acl)#exit
Router(config)# Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 traffic-filter smile in
Router(config-if)#exit
Router(config)#
We verify the configuration by first accessing the www service from the
browser of both PCs and get failure

PRACTICAL NO 5: Configuring a Zone-Based Policy Firewall (ZPF) Commands:

Part 2: Configuring SSH on Router 2
Type the following commands in the CLI mode of Router2
Router>enable
Router#configure terminal
Router(config)#ip domain-name .com
Router(config)#hostname Router2
Router2(config)#crypto key generate rsa
Router2 (config)#line vty 0 4
Router2 (config-line)#transport input ssh
Router2 (config-line)#login local
Router2 (config-line)#exit
Router2 (config)#username ismail privilege 15 password cisco
Now verify ssh from PC0 by typing the following command
ssh -l ismail 192.168.3.1

Next we access the web services of the Server using the web browser of
PC using the following

Part 3: Create the Firewall Zones on Router1
Type the following commands in the CLI mode of Router1
Router>enable
Router#configure terminal
Router(config)#show version
Router#configure terminal
Router (config)#license boot module c1900 technology-package securityk9
ACCEPT? [yes/no]: y
Router(config)#exit
Router>enable

```
Router#reload
Router>enable
Router#show version

Router# Router#configure
terminal
Router(config)#zone security in-zone
Router(config-sec-zone)#exit
Router(config)#zone security out-zone
Router(config-sec-zone)#exit
Router(config)#access-list 101 permit ip

192.168.4.0 0.0.0.255 any Router(config)#class-
map type inspect match-all in-map Router(config-
cmap)#match access-group 101

Router(config-cmap)#exit
Router(config)#policy-map type

inspect in-out Router(config-pmap)#class type inspect in-
map Router(config-pmap-
c)#inspect

Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)#
Router(config)#zone-pair security in-out-zone source in-zone destination out-zone
Router(config-sec-zone-pair)#service-policy type inspect in-out
Router(config-sec-zone-pair)#exit
Router(config)#
Router(config)#interface GigabitEthernet0/0
Router(config-if)#zone-member security in-zone
Router(config-if)#exit
Router(config)#
Router(config)#interface Serial0/1/1
Router(config-if)#zone-member security out-zone
Router(config-if)#exit
Router(config)#exit
Router#copy running-config startup-config
```

PRACTICAL NO 6: Configure IOS Intrusion Prevention System (IPS) Using the CLI Commands:

type the following command in Router1

Router#configure terminal
Router(config)#license boot module c1900 technology-package
securityk9 ACCEPT? [yes/no]: y
Press enter key
Router#
Router#reload
System configuration has been modified. Save? [yes/no]:y
Proceed with reload? [confirm] Press Enter key
Press RETURN to get started! Press Enter key

Router>enable
Router# Router#show version

We will get a message informing whether the security package is enabled or
Now type the following commands in the CLI mode of Router1

Router#
Router#clock set 10:30:45 march 3 2022
Router#mkdir smile
Create directory filename [smile]? Press enter key
Created dir flash:smile
Router#
Router#configure terminal
Router(config)#ip ips config location flash:smile
Router(config)#ip ips name iosips
Router(config)#ip ips notify log
Router(config)#ip ips signature-category
Router(config-ips-category)#category all
Router(config-ips-category-action)#retired true
Router(config-ips-category-action)#exit

Router(config-ips-category)#category ios_ips basic
Router(config-ips-category-action)#retired false
Router(config-ips-category-action)#exit
Router(config-ips-category)#exit
Do you want to accept these changes? [confirm]y
Router(config)#interface Serial0/1/0
Router(config-if)#ip ips iosips out
Router(config-if)#
Press enter key
Router(config-if)#exit
Router(config)#

Part 2: Modify the Signature

Type the following commands in the CLI mode of Router1
Router(config)#
Router(config)#ip ips signature-definition
Router(config-sigdef)#signature 2004 0
Router(config-sigdef-sig)#status
Router(config-sigdef-sig-status)#retired false
Router(config-sigdef-sig-status)#enabled true
Router(config-sigdef-sig-status)#exit
Router(config-sigdef-sig)#engine
Router(config-sigdef-sig-engine)#event-action produce-alert
Router(config-sigdef-sig-engine)#event-action deny-packet-inline
Router(config-sigdef-sig-engine)#exit
Router(config-sigdef-sig)#exit
Router(config-sigdef)#exit
Do you want to accept these changes? [confirm]y
Router(config)#

Now we need to verify the above IPS configuration, we do it first
by pinging PC1 to SERVER and then from SERVER to PC1
PC1 to SERVER – The ping fails
Server to PC1 – The Ping is successful

We check the Syslog service on the server to check the logging activity,
by typing the following commands in Router0
Router>enable
Router#configure terminal
Router(config)#logging 192.168.1.2
Router(config)#
Router(config)#
Router(config)#exit

Router#
Router#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/3 ms
Router#

Practical 7: Packet Tracer - Layer 2 Security Topology

The output shows that the bridge connected to GigabitEthernet 0/2 is the
Root Bridge, i.e., Switch 2 is the Root Bridge in the above topology.

Now we need to make Multilayer Switch0 as the Root Bridge. Type the
following commands in the CLI mode of Multilayer Switch0.
Switch#

```
Switch#configure terminal
Switch(config)#spanning-tree vlan 1 root primary
Switch(config)#do show spann
```

Now, we have made the Multilayer Switch0 as the Root Bridge.
But we also need to remove the Switch2 from Root Bridge. For that open the CLI mode of Switch2 and type the following code.

```
Switch2#configure terminal
Switch2(config)#spanning-tree vlan 1 root secondary
Switch2(config)#do show span
```

Part 2: Protect Against STP Attacks
Open CLI mode of Switch a and type the following command

```
Switcha>enable
Switcha#configure termial
Switcha(config)#interface range fastEthernet 0/1-2
Switcha(config-if-range)#switchport mode access
Switcha(config-if-range)#spanning-tree portfast
Switcha(config-if-range)#spanning-tree bpduguard enable
```

Now minimize the Switch a window and open the Switch b CLI mode and type the same command

```
Switchb>enable
Switchb#configure termial
Switchb(config)#interface range fastEthernet 0/1-2
Switchb(config-if-range)#switchport mode access
Switchb(config-if-range)#spanning-tree portfast
Switchb(config-if-range)#spanning-tree bpduguard enable
```

Now minimize the Switch b window and open the Switch 1 CLI mode and type the following command

```
Switch1>enable
Switch1#configure terminal
Switch1(config)#interface range fastEthernet 0/23-24
Switch1(config-if-range)#spanning-tree guard root
```

Now minimize the Switch 1 window and open the Switch 2 CLI mode and type the same command

```
Switch2>enable
Switch2#configure terminal
Switch2(config)#interface range fastEthernet 0/23-24
Switch2(config-if-range)#spanning-tree guard root
```

Thus, we have Protected all the switch against STP Attacks.

Part 3: Configure Port Security and Disable unsed ports
Open CLI mode of Switch a and type the following command

Switcha(config-if-range)#switchport port-security
Switcha(config-if-range)#switchport port-security maximum 2
Switcha(config-if-range)#switchport port-security mac-address sticky
Switcha(config-if-range)#switchport port-security violation shutdown

Now minimize the Switch a window and open the Switch b CLI mode and type the same command
Switchb(config-if-range)#switchport port-security
Switchb(config-if-range)#switchport port-security maximum 2
Switchb(config-if-range)#switchport port-security mac-address sticky
Switchb(config-if-range)#switchport port-security violation shutdown

Now let us check if the security is enabled or not. Open CLI mode of Switch a and type the following
Switcha(config-if-range)# CTRL Z
Switcha#show port-security interface f0/1

Let us now disable all the unused ports in switch a and switch b.
Open the CLI mode of Switch a and type the following command
Switcha#enable
Switcha#configure terminal
Switcha(config)#interface range fastEthernet 0/3-22
Switcha(config-if-range)#shutdown
Open the CLI mode of Switch b and type the following command
Switchb#enable
Switchb#configure terminal
Switchb(config)#interface range fastEthernet 0/3-22
Switchb(config-if-range)#shutdown

Thus, Port Security is enabled and all the unsed ports are disabled.

PRACTICAL - 8 Packet Tracer - Layer 2 VLAN Security

Objectives
● Connect a new redundant link between SW-1 and SW-2.
● Enable trunking and configure security on the new trunk link between SW-1 and SW-2.
● Create a new management VLAN (VLAN 20) and attach a management PC to that VLAN.
● Implement an ACL to prevent outside users from accessing the management VLAN.
Background / Scenario
A company's network is currently set up using two separate VLANs: VLAN 5 and VLAN 10. In addition, all trunk ports are configured with native VLAN 15. A network administrator wants to add a redundant link between switch SW-1 and SW-2. The link must have trunking enabled and all security requirements should be in place.
In addition, the network administrator wants to connect a management PC to switch SW-A. The administrator
would like to enable the management PC to connect to all switches and the router, but does not want any

other devices to connect to the management PC or the switches. The administrator would like to create a new
VLAN 20 for management purposes.
All devices have been preconfigured with:
o Enable secret password: ciscoenpa55
o Console password: ciscoconpa55

o SSH username and password: SSHadmin / ciscosshpa55
Part 1: Verify Connectivity
Step 1: Verify connectivity between C2 (VLAN 10) and C3 (VLAN 10).
Step 2: Verify connectivity between C2 (VLAN 10) and D1 (VLAN 5).
Note: If using the simple PDU GUI packet, be sure to ping twice to allow for ARP.
Part 2: Create a Redundant Link Between SW-1 and SW-2
Step 1: Connect SW-1 and SW-2.
Using a crossover cable, connect port F0/23 on SW-1 to port F0/23 on SW-2.
Step 2: Enable trunking, including all trunk security mechanisms on the link between SW-1 and SW-2.
Trunking has already been configured on all pre-existing trunk interfaces. The new link must be configured for
trunking, including all trunk security mechanisms. On both SW-1 and SW-2, set the port to trunk, assign native VLAN 15 to the trunk port, and disable auto-negotiation.
SW-1(config)# interface f0/23
SW-1(config-if)# switchport mode trunk
SW-1(config-if)# switchport trunk native vlan 15
SW-1(config-if)# switchport nonegotiate
SW-1(config-if)# no shutdown
SW-2(config)# interface f0/23
SW-2(config-if)# switchport mode trunk
SW-2(config-if)# switchport trunk native vlan 15
SW-2(config-if)# switchport nonegotiate
SW-2(config-if)# no shutdown
Part 3: Enable VLAN 20 as a Management VLAN
The network administrator wants to access all switch and routing devices using a management PC. For security purposes, the administrator wants to ensure that all managed devices are on a separate VLAN.
Step 1: Enable a management VLAN (VLAN 20) on SW-A.
a. Enable VLAN 20 on SW-A.
SW-A(config)# vlan 20
SW-A(config-vlan)# exit
b. Create an interface VLAN 20 and assign an IP address within the 192.168.20.0/24 network.
SW-A(config)# interface vlan 20
SW-A(config-if)# ip address 192.168.20.1 255.255.255.0
Step 2: Enable the same management VLAN on all other switches.
a. Create the management VLAN on all switches: SW-B, SW-1, SW-2, and Central.

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

SW-2(config)# vlan 20
SW-2(config-vlan)# exit
Central(config)# vlan 20
Central(config-vlan)# exit
b. Create an interface VLAN 20 on all switches and assign an IP address within the 192.168.20.0/24 network.
SW-B(config)# interface vlan 20
SW-B(config-if)# ip address 192.168.20.2 255.255.255.0
SW-1(config)# interface vlan 20
SW-1(config-if)# ip address 192.168.20.3 255.255.255.0
SW-2(config)# interface vlan 20
SW-2(config-if)# ip address 192.168.20.4 255.255.255.0
Central(config)# interface vlan 20
Central(config-if)# ip address 192.168.20.5 255.255.255.0

Step 3: Connect and configure the management PC.
Connect the management PC to SW-A port F0/1 and ensure that it is assigned an available IP address within
the 192.168.20.0/24 network.
Step 4: On SW-A, ensure the management PC is part of VLAN 20.
Interface F0/1 must be part of VLAN 20.
SW-A(config)# interface f0/1
SW-A(config-if)# switchport access vlan 20
SW-A(config-if)# no shutdown
Step 5: Verify connectivity of the management PC to all switches.
The management PC should be able to ping SW-A, SW-B, SW-1, SW-2, and Central.
Part 4: Enable the Management PC to Access Router R1
Step 1: Enable a new subinterface on router R1.
a. Create subinterface g0/0.3 and set encapsulation to dot1q 20 to account for VLAN 20.
R1(config)# interface g0/0.3
R1(config-subif)# encapsulation dot1q 20

b. Assign an IP address within the 192.168.20.0/24 network.
R1(config)# interface g0/0.3
R1(config-subif)# ip address 192.168.20.100 255.255.255.0
Step 2: Verify connectivity between the management PC and R1.
Be sure to configure the default gateway on the management PC to allow for connectivity.
Step 3: Enable security.
While the management PC must be able to access the router, no other PC should be able to access the management VLAN.
a. Create an ACL that allows only the Management PC to access the router.
Example: (may vary from student configuration)
R1(config)# access-list 101 deny ip any 192.168.20.0 0.0.0.255
R1(config)# access-list 101 permit ip any any
R1(config)# access-list 102 permit ip host 192.168.20.50 any
b. Apply the ACL to the proper interface(s).
Example: (may vary from student configuration)

R1(config)# interface g0/0.1
R1(config-subif)# ip access-group 101 in
R1(config-subif)# interface g0/0.2
R1(config-subif)# ip access-group 101 in
R1(config-subif)# line vty 0 4
R1(config-line)# access-class 102 in
Note: Access list 102 is used to only allow the Management PC (192.168.20.50 in this example) to access the
router. This prevents an IP address change to bypass the ACL.
Note: There are multiple ways in which an ACL can be created to accomplish the necessary security. For this
reason, grading on this portion of the activity is based on the correct connectivity requirements. The management PC must be able to connect to all switches and the router. All other PCs should not be able to connect to any devices within the management VLAN.
Step 4: Verify security.
a. Verify only the Management PC can access the router. Use SSH to access R1 with username SSHadmin and password ciscosshpa55.
PC> ssh -l SSHadmin 192.168.20.100
b. From the management PC, ping SW-A, SW-B, and R1. Were the pings successful? Explain.

_ _ _
_ _ _
_ _ _

The pings should have been successful because all devices within the 192.168.20.0 network should be able to ping one another. Devices within VLAN20 are not required to route through the router.
c. From D1, ping the management PC. Were the pings successful? Explain.

The ping should have failed because for a device within a different VLAN to successfully ping a device within VLAN20, it must be routed. The router has an ACL that prevents all packets from accessing the 192.168.20.0 network.
Step 5: Check results.
Your completion percentage should be 100%. Click Check Results to view feedback and verification of which
required components have been completed.
If all components appear to be correct and the activity still shows incomplete, it could be due to the connectivity tests that verify the ACL operation.
!!! Script for SW-1
conf t
interface f0/23
switchport mode trunk
switchport trunk native vlan 15
switchport nonegotiate
no shutdown
vlan 20
exit
interface vlan 20
ip address 192.168.20.3 255.255.255.0

```
!!! Script for SW-2
conf t
interface f0/23
switchport mode trunk
switchport trunk native vlan 15
switchport nonegotiate
no shutdown
vlan 20
exit
interface vlan 20
ip address 192.168.20.4 255.255.255.0
!!! Script for SW-A
conf t
vlan 20
exit
interface vlan 20
ip address 192.168.20.1 255.255.255.0
interface f0/1
switchport access vlan 20
no shutdown

!!! Script for SW-B
conf t
vlan 20
exit
interface vlan 20
ip address 192.168.20.2 255.255.255.0
!!! Script for Central
conf t
vlan 20
exit
interface vlan 20
ip address 192.168.20.5 255.255.255.0
!!! Script for R1
conf t
interface GigabitEthernet0/0.1
ip access-group 101 in
interface GigabitEthernet0/0.2
ip access-group 101 in
interface g0/0.3
encapsulation dot1q 20
ip address 192.168.20.100 255.255.255.0
access-list 101 deny ip any 192.168.20.0 0.0.0.255
access-list 101 permit ip any any
access-list 102 permit ip host 192.168.20.50 any
line vty 0 4
access-class 102 in
```

PRACTICAL - 1

Configure Cisco Routers for OSPF MD5 Authentication, Syslog and NTP

OSPF, MD5 Authentication

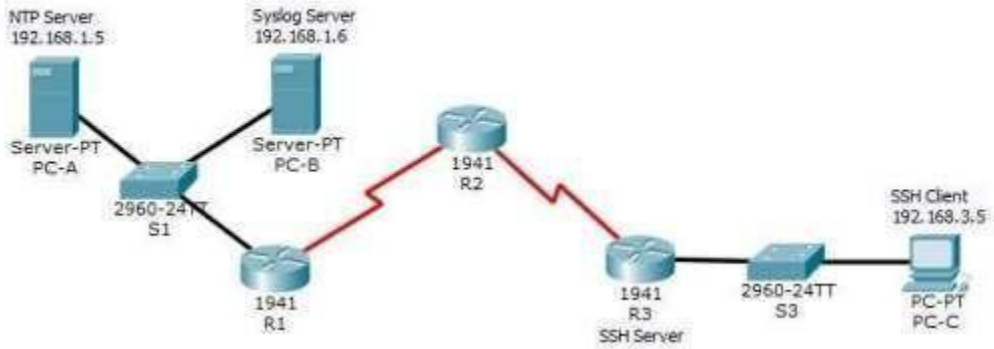
- OSPF is a routing protocol. Two routers speaking OSPF to each other exchange information about the routes they know about and the cost for them to get there.
- When many OSPF routers are part of the same network, information about all of the routes in a network are learned by all of the OSPF routers within that network— technically called an **area**. (We'll talk more about area as we go on).
- Each OSPF router passes along information about the routes and costs they've heard about to all of their adjacent OSPF routers, called **neighbors**.
- OSPF routers rely on **cost** to compute the shortest path through the network between themselves and a remote router or network destination.
- The shortest path computation is done using Djikstra's algorithm. This algorithm isn't unique to OSPF. Rather, it's a mathematical algorithm that happens to have an obvious application to networking.

MD5 Authentication

- MD5 authentication provides higher security than plain text authentication.
- This method uses the MD5 algorithm to compute a hash value from the contents of the OSPF packet and a password (or key).
- This hash value is transmitted in the packet, along with a key ID and a non-decreasing sequence number.
- The receiver, which knows the same password, calculates its own hash value.

- If nothing in the message changes, the hash value of the receiver should match the hash value of the sender which is transmitted with the message.
- The key ID allows the routers to reference multiple passwords.
- This makes password migration easier and more secure.
- For example, to migrate from one password to another, configure a password under a different key ID and remove the first key.
- The sequence number prevents replay attacks, in which OSPF packets are captured, modified, and retransmitted to a router.
- As with plain text authentication, MD5 authentication passwords do not have to be the same throughout an area. However, they do need to be the same between neighbors.

Topology

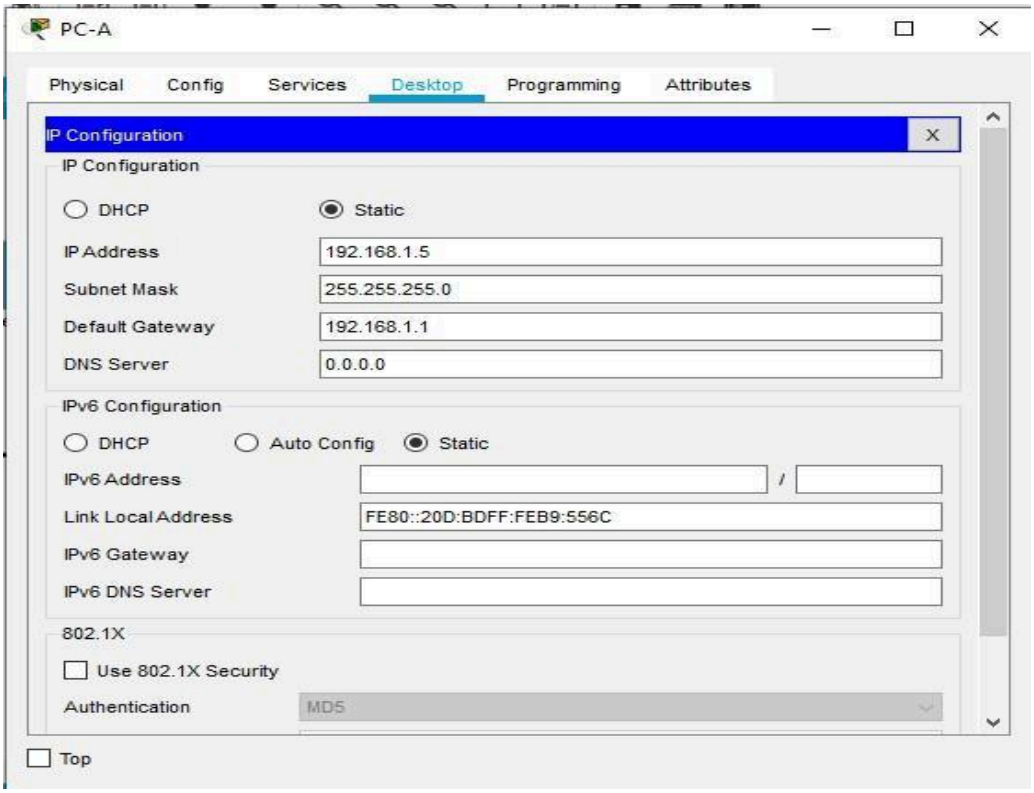


Addressing Table

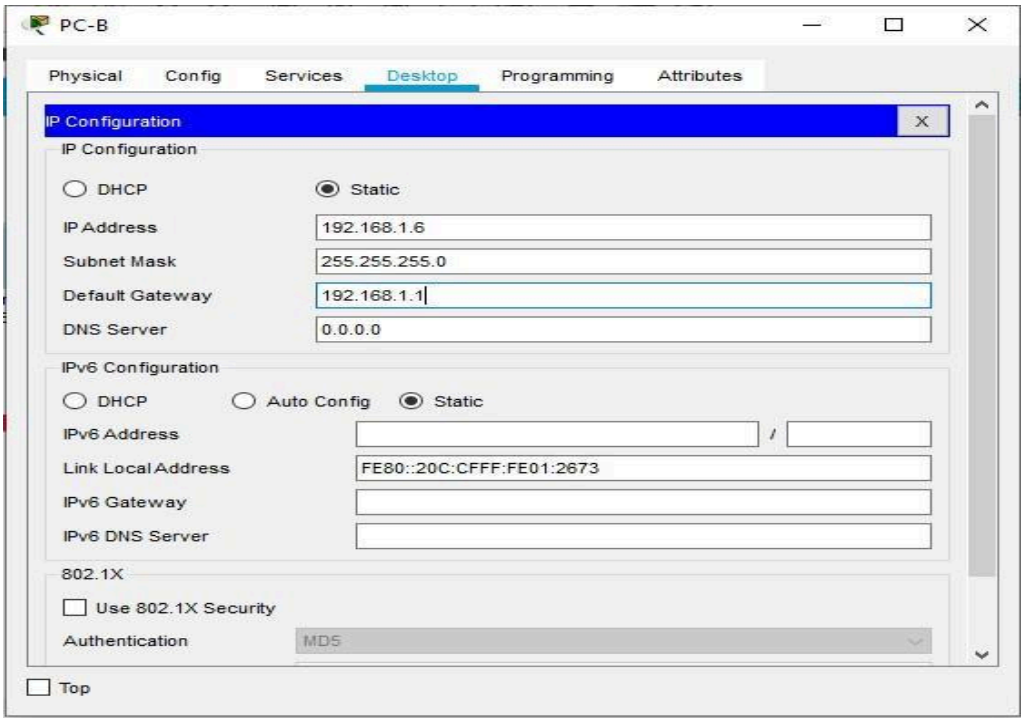
Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1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

Configuration

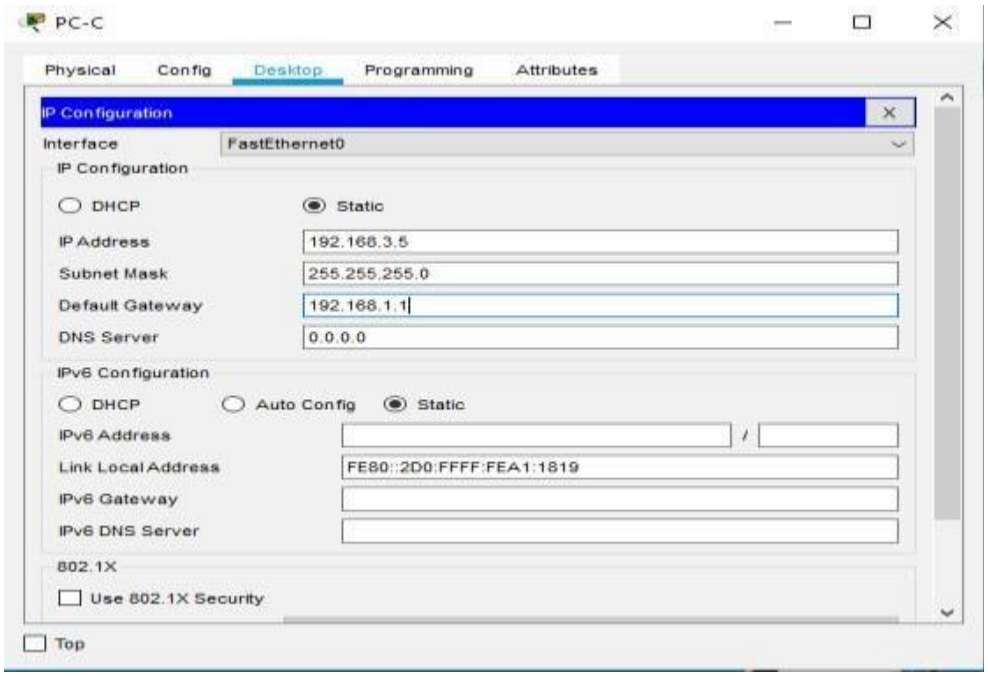
PC - A



PC - B

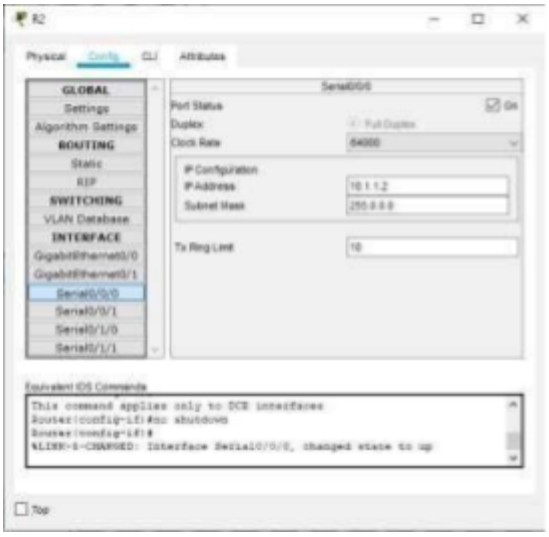


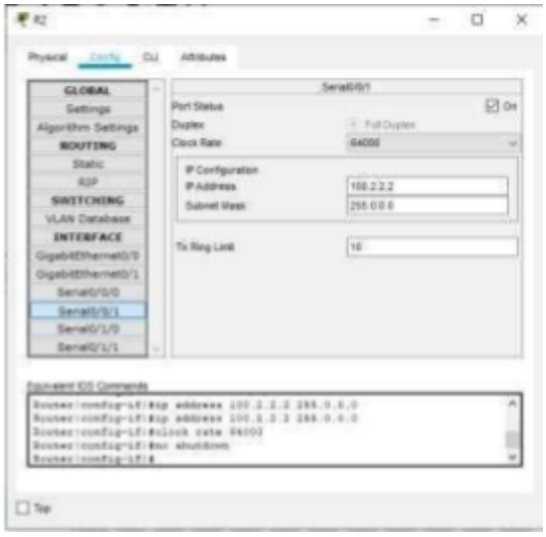
PC - C



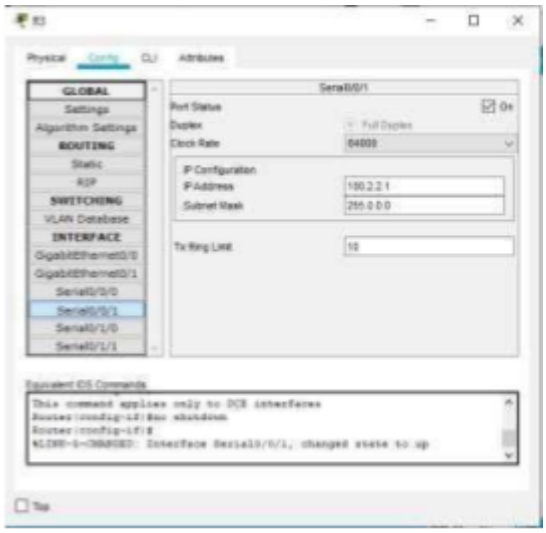
ROUNTER - 1

ROUNTER - 2





ROUNTER - 3



Part 1: Configure OSPF MD5 Authentication

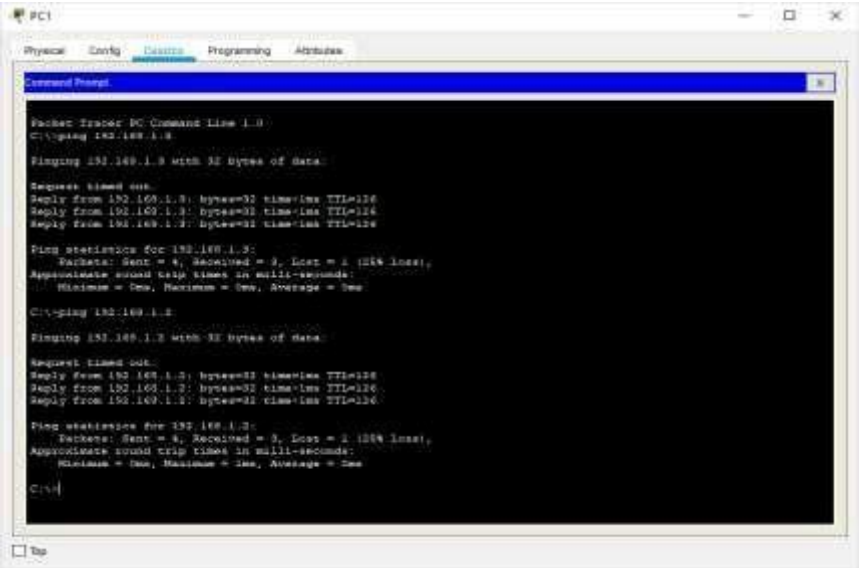
ROUTER 1: Type the following command in the CLI mode
Router>enable
Router#configure
terminal
Router(config)#router
ospf 1
Router(config-router)#network 192.168.1.0 0.255.255.255 area 1
Router(config-router)#network 10.1.1.0
0.255.255.255 area 1 Router(config-router)#exit

Router(config)#exit
Router#

ROUTER 2: Type the following command in the CLI mode
Router>enable
Router#configure
terminal
Router(config)#router
ospf 1
Router(config-router)#network 10.1.1.0 0.255.255.255 area 1
Router(config-router)#network 100.2.2.0
0.255.255.255 area 1 Router(config-router)#exit
Router(config)#exit
Router#

ROUTER 3: Type the following command in the CLI mode
Router>enable
Router#configure
terminal
Router(config)#router
ospf 1
Router(config-router)#network 192.168.3.0 0.255.255.255 area 1
Router(config-router)#network 100.2.2.0
0.255.255.255 area 1 Router(config-router)#exit
Router(config)#exit
Router#

Now we verify the connectivity by using the following



Hence OSPF has been verified

MD5 Authentication

ROUTER 1: Type the following command in the CLI mode

```
Router>e
nable
Router#
Router#configure terminal
Router(config)#interface
Serial0/0/0
Router(config-if)#ip ospf authentication
message-digest Router(config-if)#ip ospf
message-digest-key 1 md5 smile
Router(config-if)#exit
Router(config)#exit
```

ROUTER 2: Type the following command in the CLI mode

```
Router>e
nable
Router#
Router#configure terminal
Router(config)#interface
Serial0/0/0
Router(config-if)#ip ospf authentication
message-digest Router(config-if)#ip ospf
message-digest-key 1 md5 smile
Router(config-if)#exit
Router(config)#exit
```

Verify the MD5 Authentication using the following command in the CLI mode of Router1

Router#show ip ospf interface gigabitEthernet 0/1

We get the following output:

```
GigabitEthernet0/1 is up, line protocol
is up Internet address is
192.168.2.1/24, Area 1
Process ID 1, Router ID 192.168.2.1, Network Type BROADCAST,
Cost: 1 Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 192.168.3.1, Interface address 192.168.2.2
```

Backup Designated Router (ID) 192.168.2.1, Interface address
192.168.2.1 Timer intervals configured, Hello 10, Dead 40, Wait 40,
Retransmit 5
Hello due in 00:00:06
Index 2/2, flood queue
length 0 Next
0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 192.168.3.1 (Designated
Router) Suppress hello for 0 neighbor(s)
Message digest authentication enabled

Youngest key id is 1

MD5 Authentication has been verified

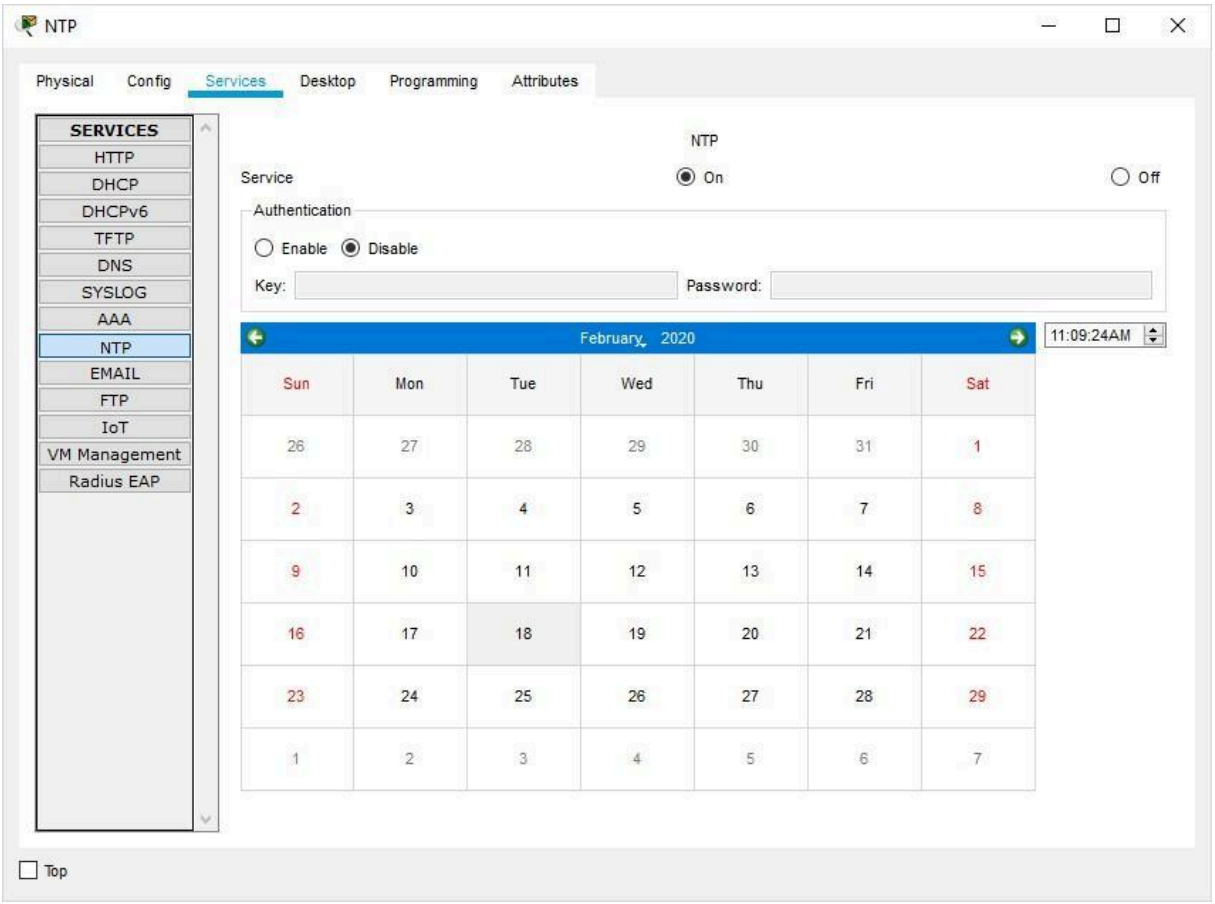
PART - 2 NTP

Network Time Protocol (NTP) is a TCP/IP protocol used to synchronize computer clocks across data networks.

NTP was developed in the 1980s by D.L. Mills at the University of Delaware to achieve highly accurate time synchronization and to sustain the effects of variable latency over packet-switched data networks through a jitter buffer.

We use the same topology to study the given protocol

Configure NTP Server and enable the NTP service



Now Go to CLI Mode of Router1 and type the following commands on both the Routers

```
Router#enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ntp server 192.168.1.5
Router(config)#ntp update-calendar
Router(config)#exit
Router#
```

To verify the Output we use the following command

Router#show clock

18:12:43.760 UTC Fri Jan 14 2022
Router#

PART - 3 SYSLOG server

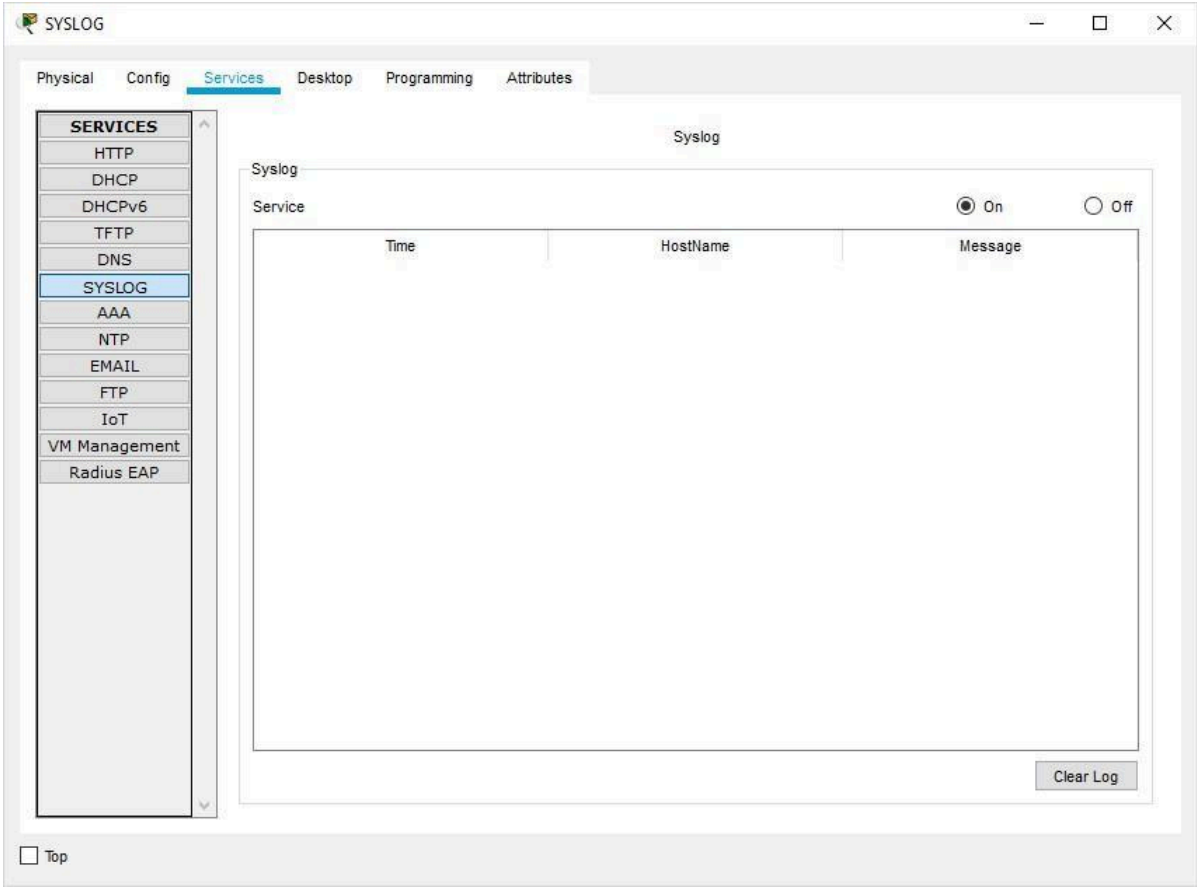
Configure SYSLOG Server and enable the service

Syslog is a way for network devices to send event messages to a logging server usually known as a Syslog server.

The Syslog **protocol** is supported by a wide range of devices and can be used to log different types of events.

For example, a router might send messages about users logging on to console sessions, while a web-server might log access-denied events.

Turn ON the SYSLOG service on the server

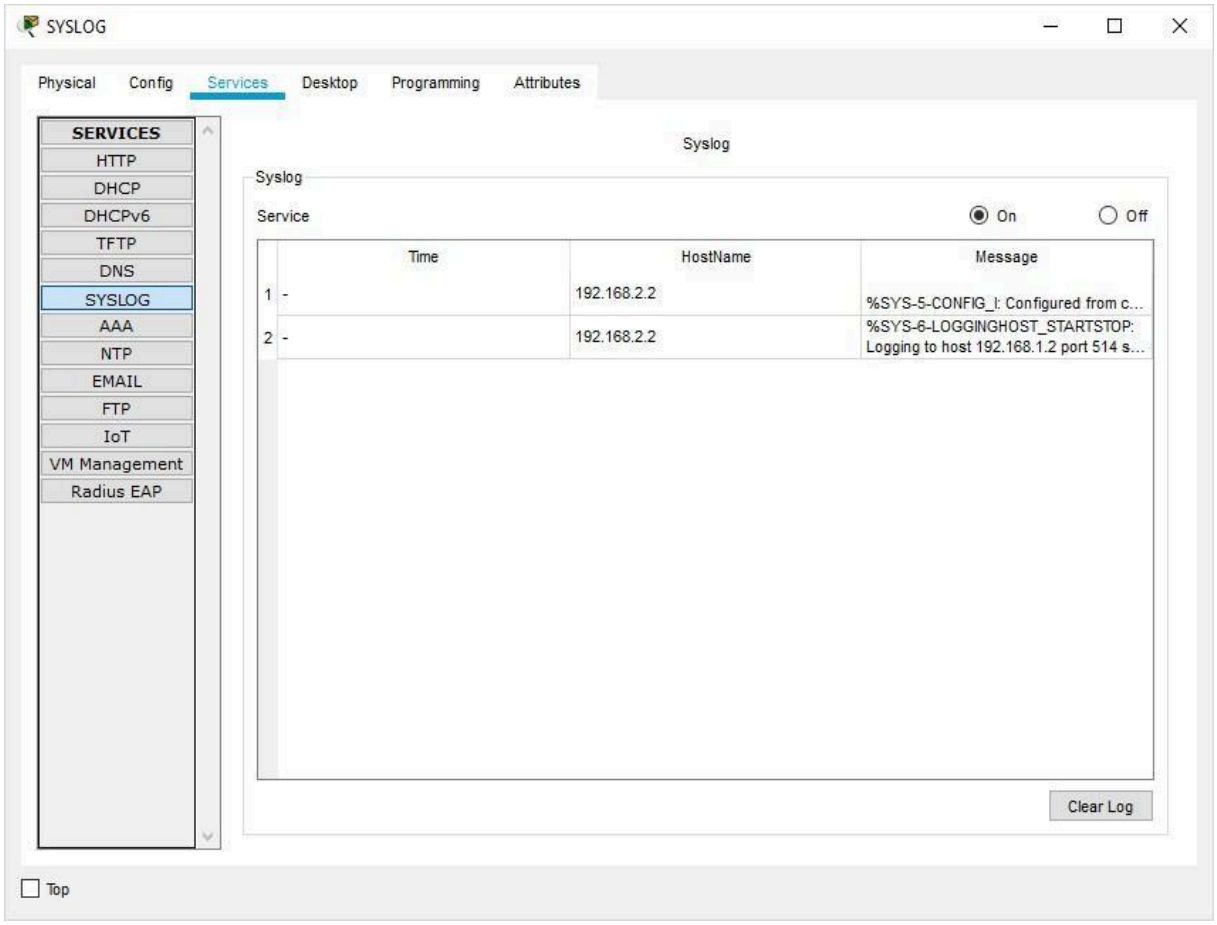


And Turn OFF on all other Servers

Now Go to CLI Mode of any Router and type the following commands in all the Routers.

```
Router#
Router#configure terminal
Router(config)#logging 192.168.1.6
Router(config)#exit
Router#
```

Output



PRACTICAL NO 2: Configure AAA Authentication on Cisco Routers

To provide a centralized management system for the authentication, authorization and accounting (AAA framework), Access Control Server (ACS) is used. For the communication between the client and the ACS server, two protocols are used namely TACACS+ and RADIUS.

TACACS+

Terminal Access Controller Access Control System (TACACS+) is Cisco proprietary protocol which is used for the communication of the Cisco client and Cisco ACS server. It uses TCP port number 49 which makes it reliable.

RADIUS –

Remote Access Dial In User Service (RADIUS) is an open standard protocol used for the communication between any vendor AAA client and ACS server. If one of the client or servers is from any other vendor (other than Cisco) then we have to use RADIUS. It uses port number 1812 for authentication and authorization and 1813 for accounting.

TACACS+	RADIUS
Cisco proprietary protocol	open standard protocol
It uses TCP as transmission protocol	It uses UDP as transmission protocol
It uses TCP port number 49	It uses UDP port number 1812 for authentication and authorization and 1813 for accounting
Authentication, Authorization and Accounting is separated	Authentication, Authorization and Accounting is combined
All the AAA packets are encrypted	Only the passwords are encrypted while the other information such as username, accounting information are not encrypted
Preferably used for ACS	used when ISE is used
It provides more granular control i.e can specify the particular command for authorization	No external authorization of commands supported
offers multiprotocol support	No multiprotocol support
Used for device administration	used for network access

Similarities –

The process is start by Network Access Device (NAD – client of TACACS+ or RADIUS). NAD contact the TACACS+ or RADIUS server and transmit the request for authentication (username and password) to the server. First, NAD obtain username prompt and transmit the username to the server and then again, the server is contact by NAD to obtain password prompt and then the password is sent to the server.

The server replies with access-accept message if the credentials are valid otherwise send an access- reject message to the client. Further authorisation and accounting is different in both protocols as authentication and authorisation is combined in RADIUS.

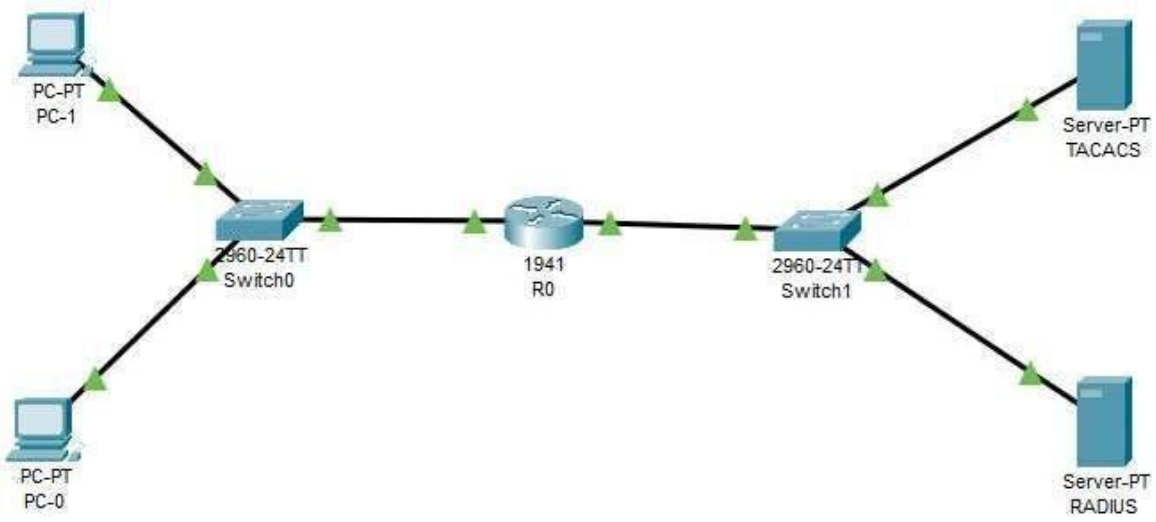
Advantages (TACACS+ over RADIUS) –

1. As TACACS+ uses TCP therefore more reliable than RADIUS.
2. TACACS+ provides more control over the authorization of commands while in RADIUS, no external authorization of commands is supported.
3. All the AAA packets are encrypted in TACACS+ while only the passwords are encrypted in RADIUS i.e more secure.

Advantage (RADIUS over TACACS+) –

1. As it is open standard therefore RADIUS can be used with other vendors device while because TACACS+ is Cisco proprietary, it can be used with Cisco devices only.
2. It has more extensive accounting support than TACACS+.

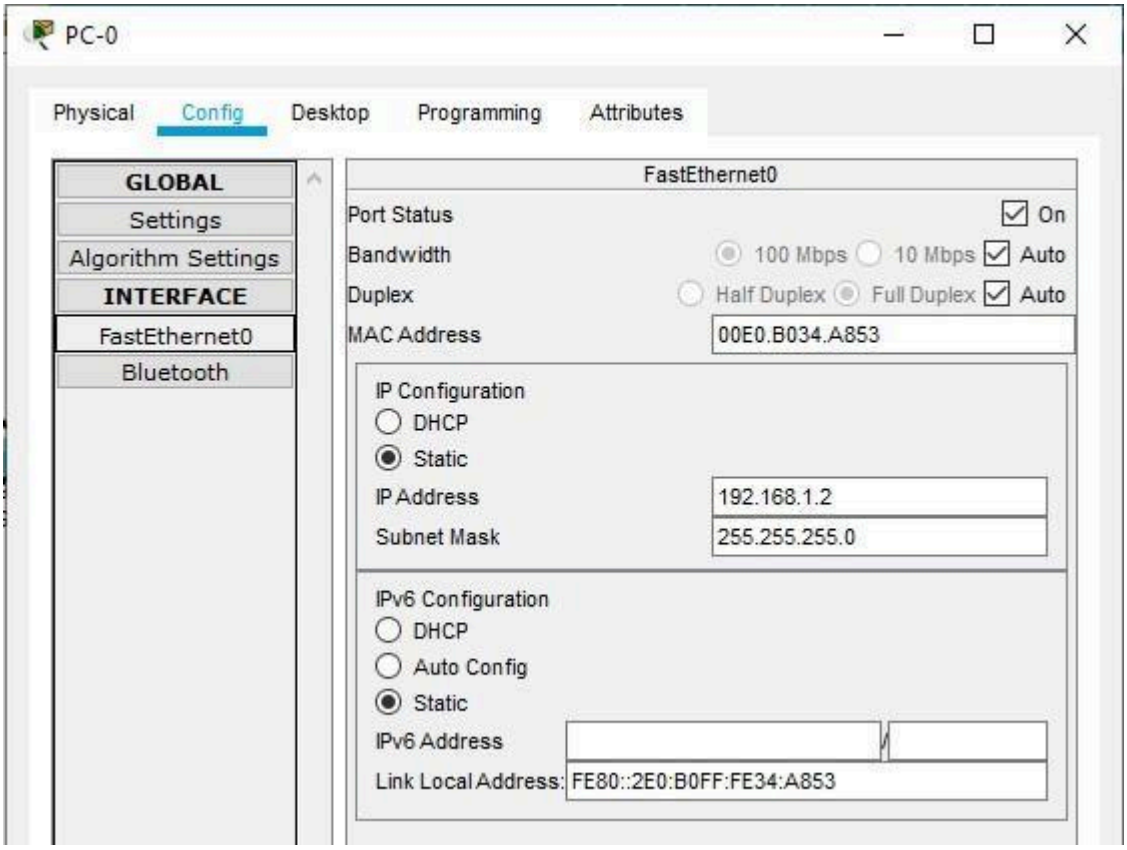
Let us consider the following Topology to understand the above AAA authentication.



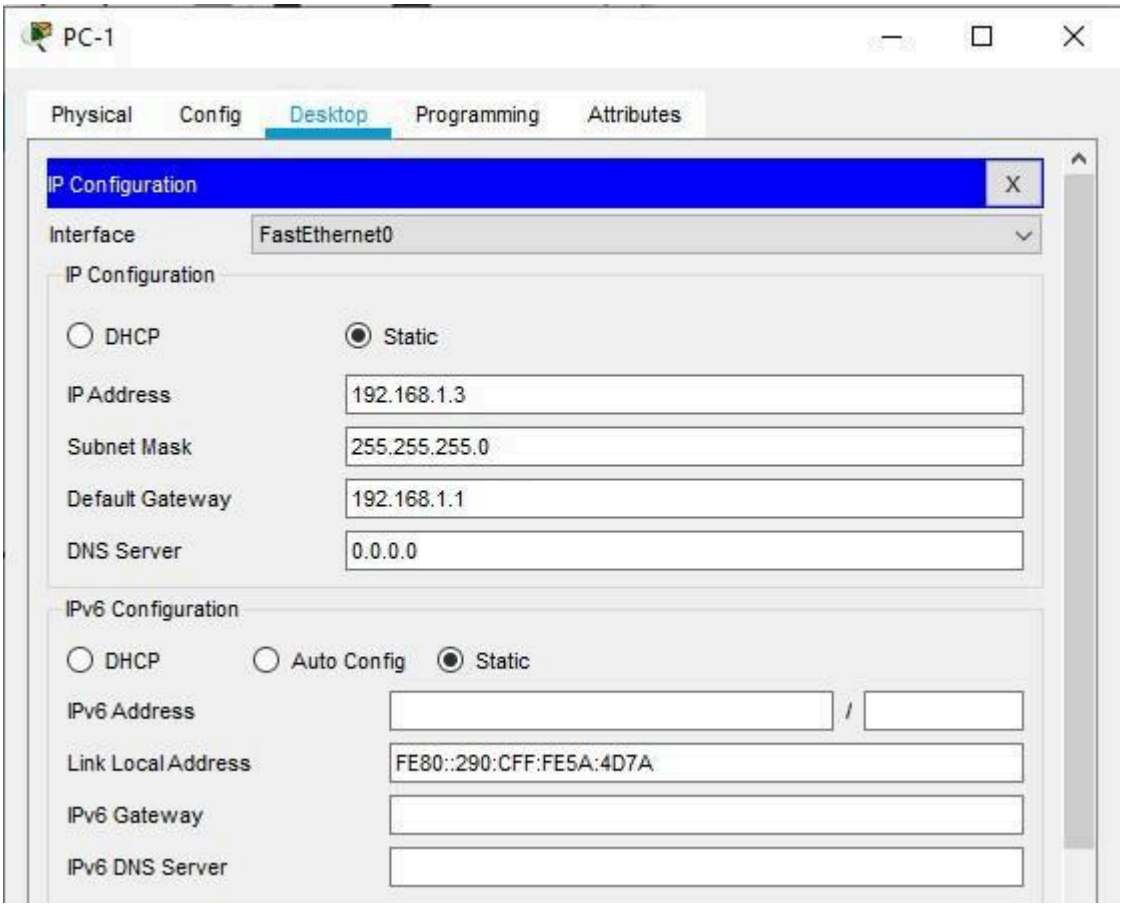
Let us consider the following Address table to configure the network devices:

Device	Interface	IP Address	Subnet Mask	Default gateway	Switch Port
TACACS	NIL	192.168.2.3	255.255.255.0	192.168.2.1	S1 F0/6
RADIUS	NIL	192.168.2.2	255.255.255.0	192.168.2.1	S1 F0/1
PC-0	NIL	192.168.1.2	255.255.255.0	192.168.1.1	S0 F0/6
PC-1	NIL	192.168.1.3	255.255.255.0	192.168.1.1	S0 F0/1
R0	GE0/0	192.168.1.1	255.255.255.0	NA	S0 F0/5
	GE0/1	192.168.2.1	255.255.255.0	NA	S1 F0/5

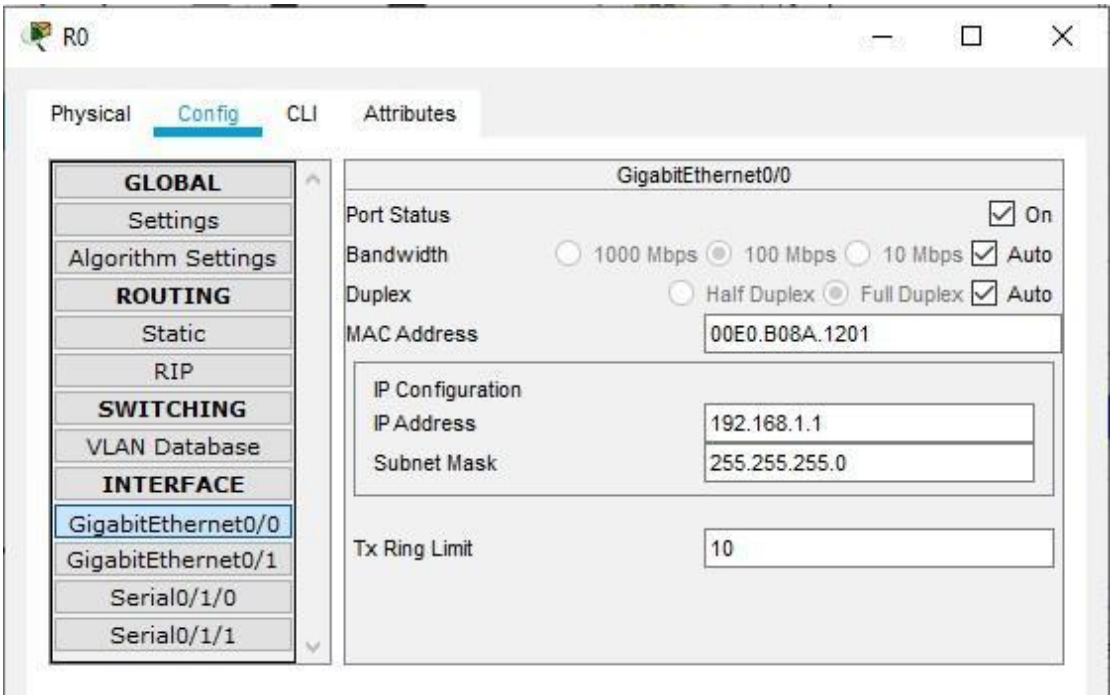
Configuring PC-0

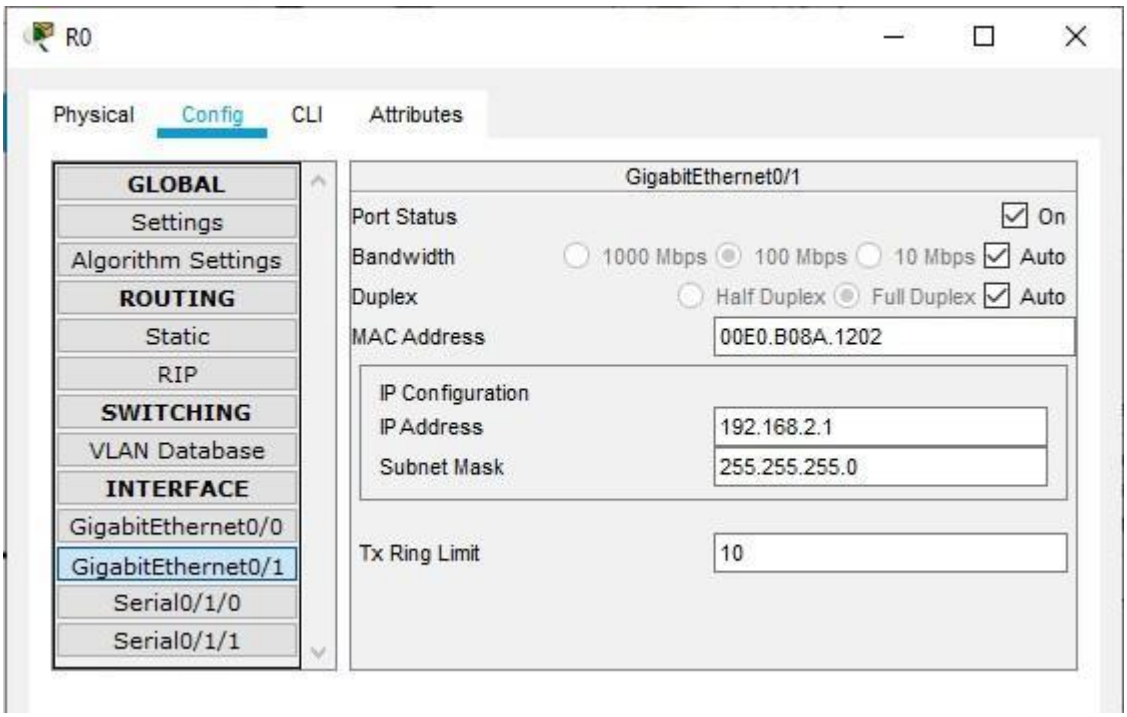


Configuring PC-1

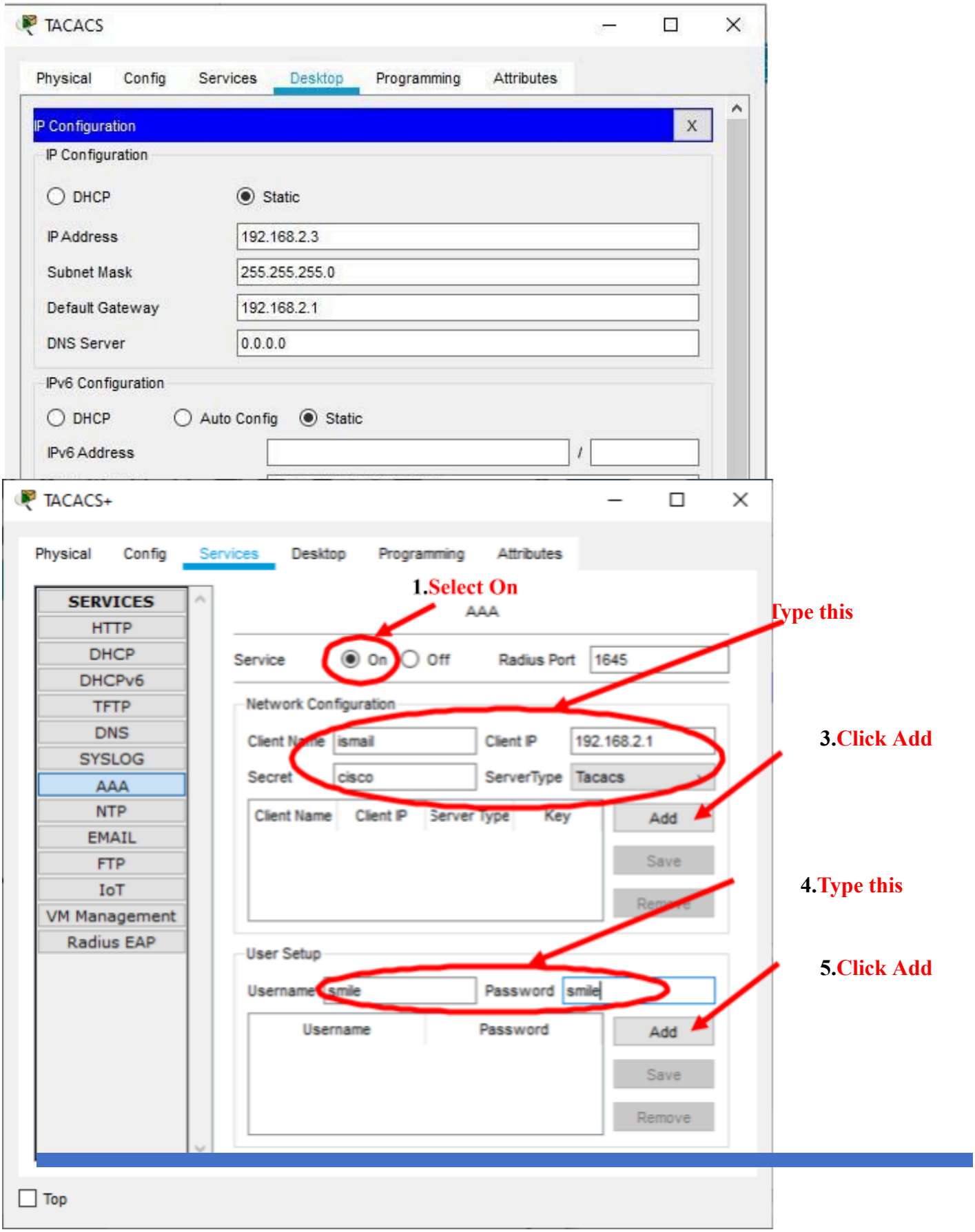


Configuring Router R0

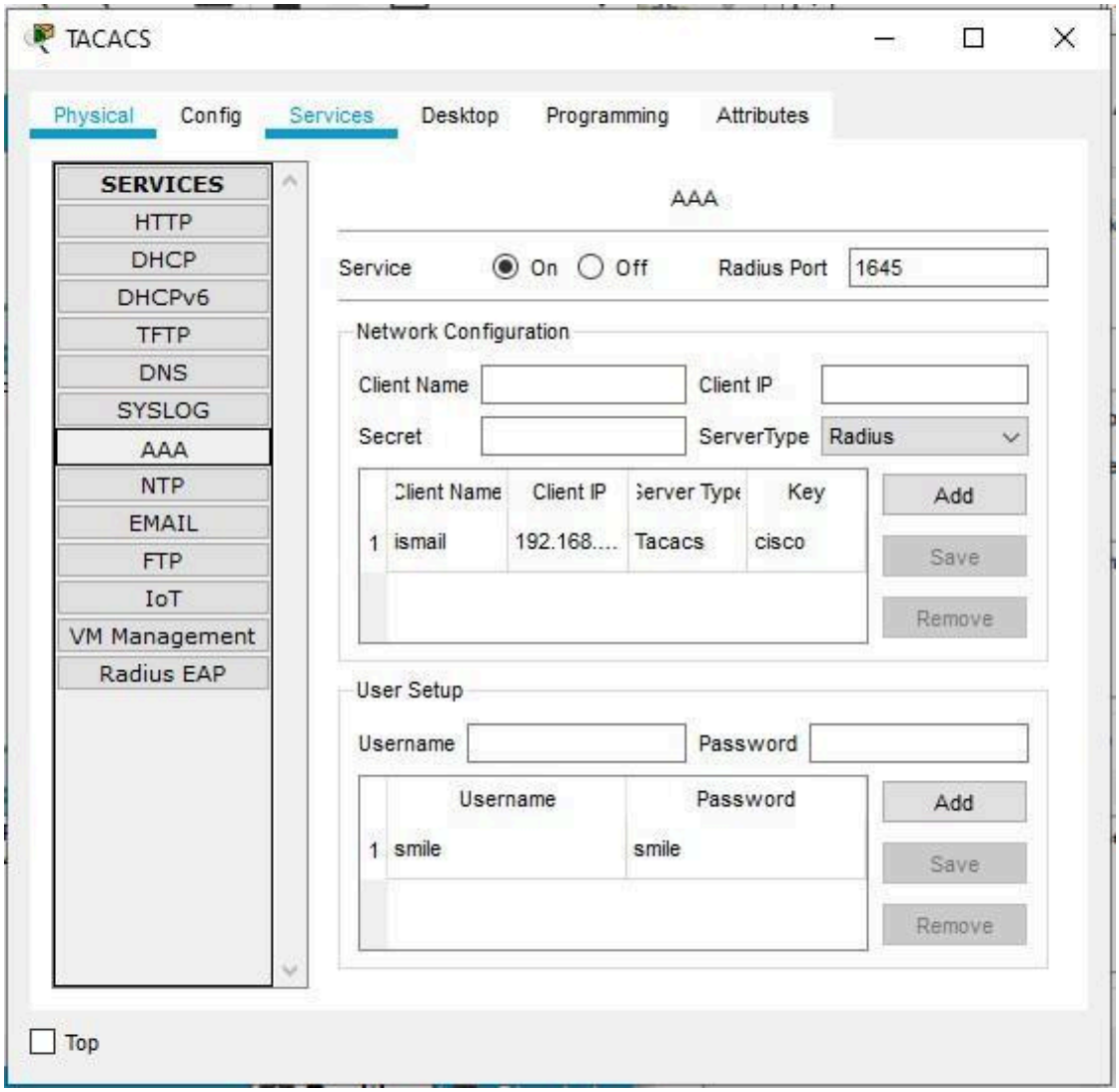




Configuring TACACS


TACACS
Physical
Config
Services
Desktop
Programming
Attributes
IP Configuration
IP Configuration
DHCP
Static
IP Address
192.168.2.3
Subnet Mask
255.255.255.0
Default Gateway
192.168.2.1
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
TACACS+
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
1.Select On
AAA
Service
On
Off
Radius Port
1645
Network Configuration
Client Name
ismail
Client IP
192.168.2.1
Secret
cisco
ServerType
Tacacs
Client Name
Client IP
Server Type
Key
Add
Save
Remove
Type this
3.Click Add
4.Type this
User Setup
Username
smile
Password
smile
Username
Password
Add
Save
Remove
5.Click Add
Top

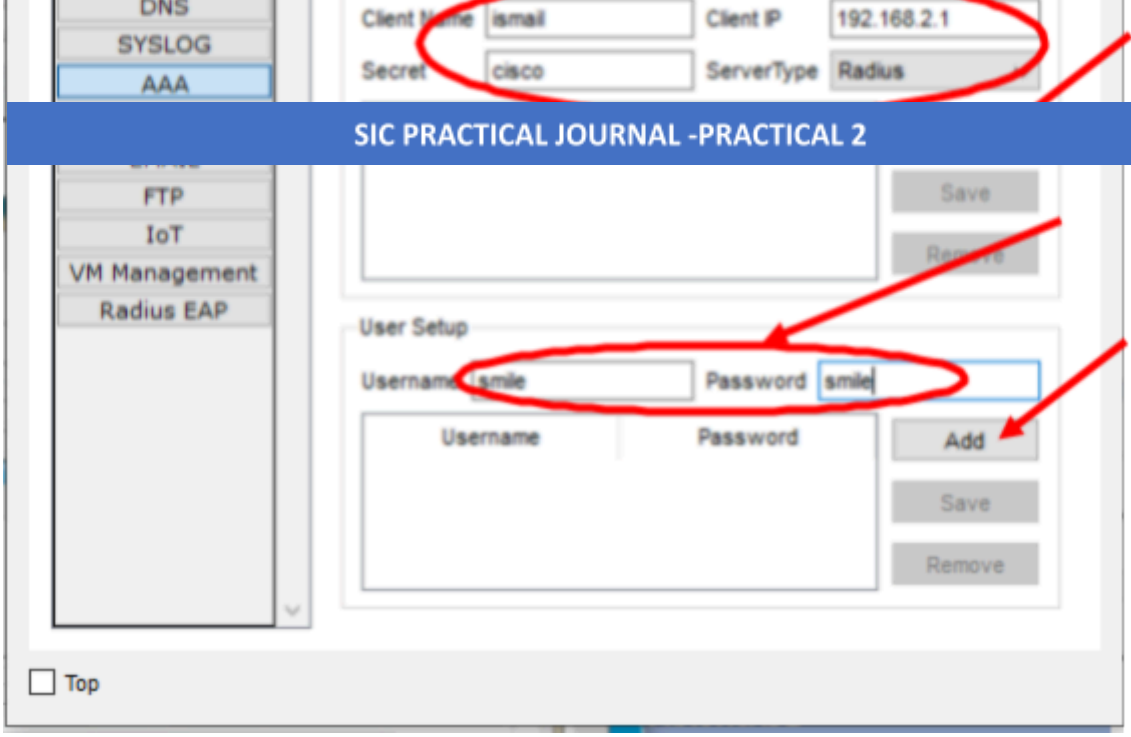
Your window should look like below image after you click Add button



Configuring RADIUS

2.Type this

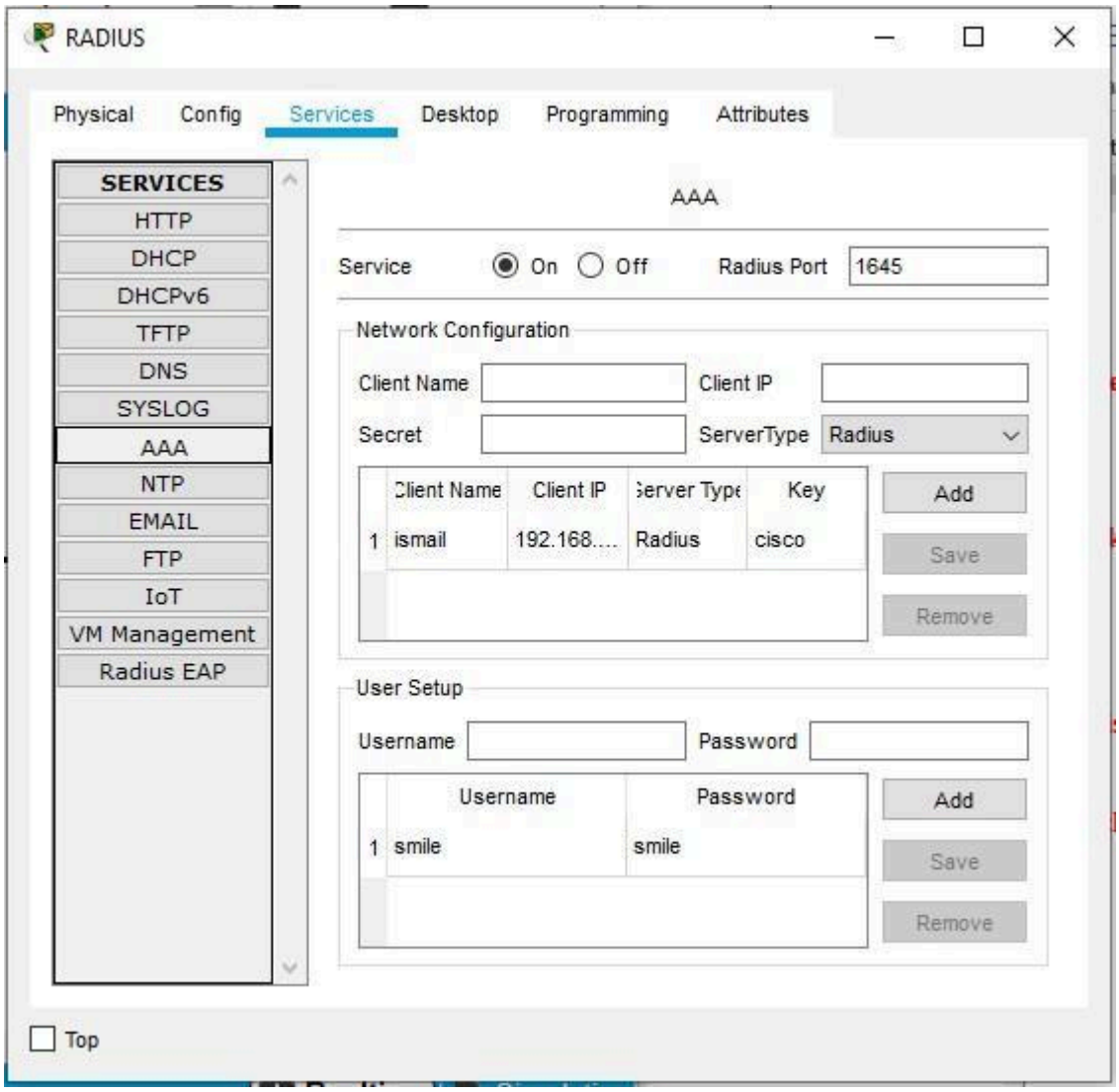
3.Click Add



4.Type this

5.Click Add

Your window should look like below image after you click Add button

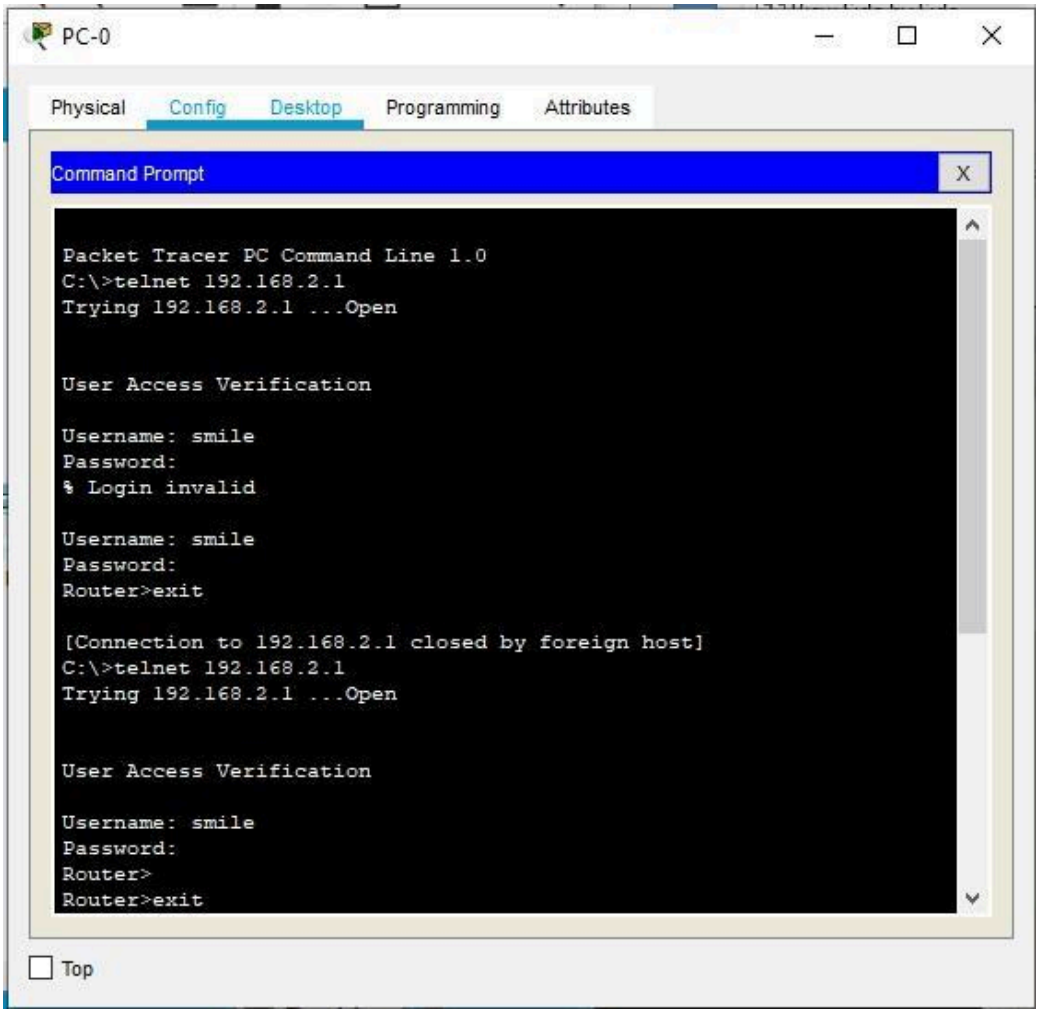


Type the following commands in the CLI mode of the Router0

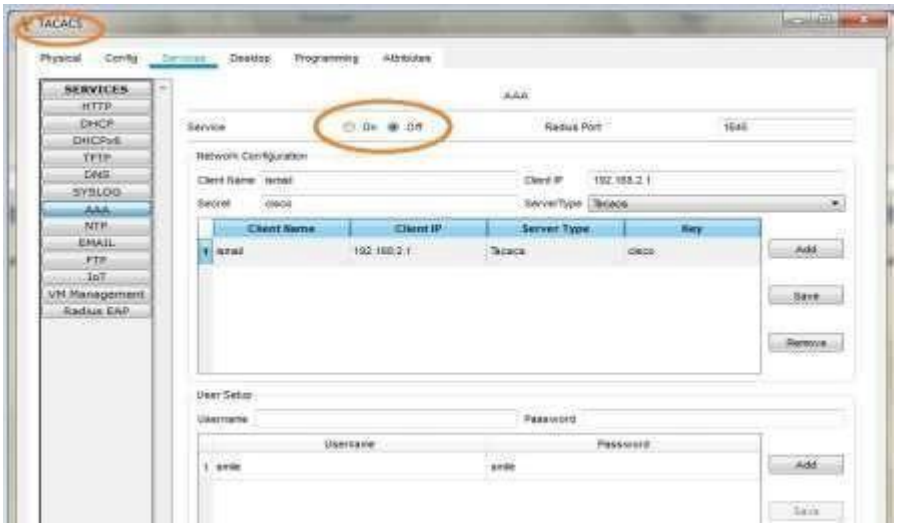
Router>enable
Router#configure terminal
Router(config)#aaa new-model
Router(config)#tacacs-server host 192.168.2.3 key cisco
Router(config)#radius-server host 192.168.2.2 key cisco
Router(config)#aaa authentication login ismail group tacacs+ group radius local
Router(config)#line vty 0 4
Router(config-line)#login authentication ismail
Router(config-line)#exit

Router(config)#
To get check the output:

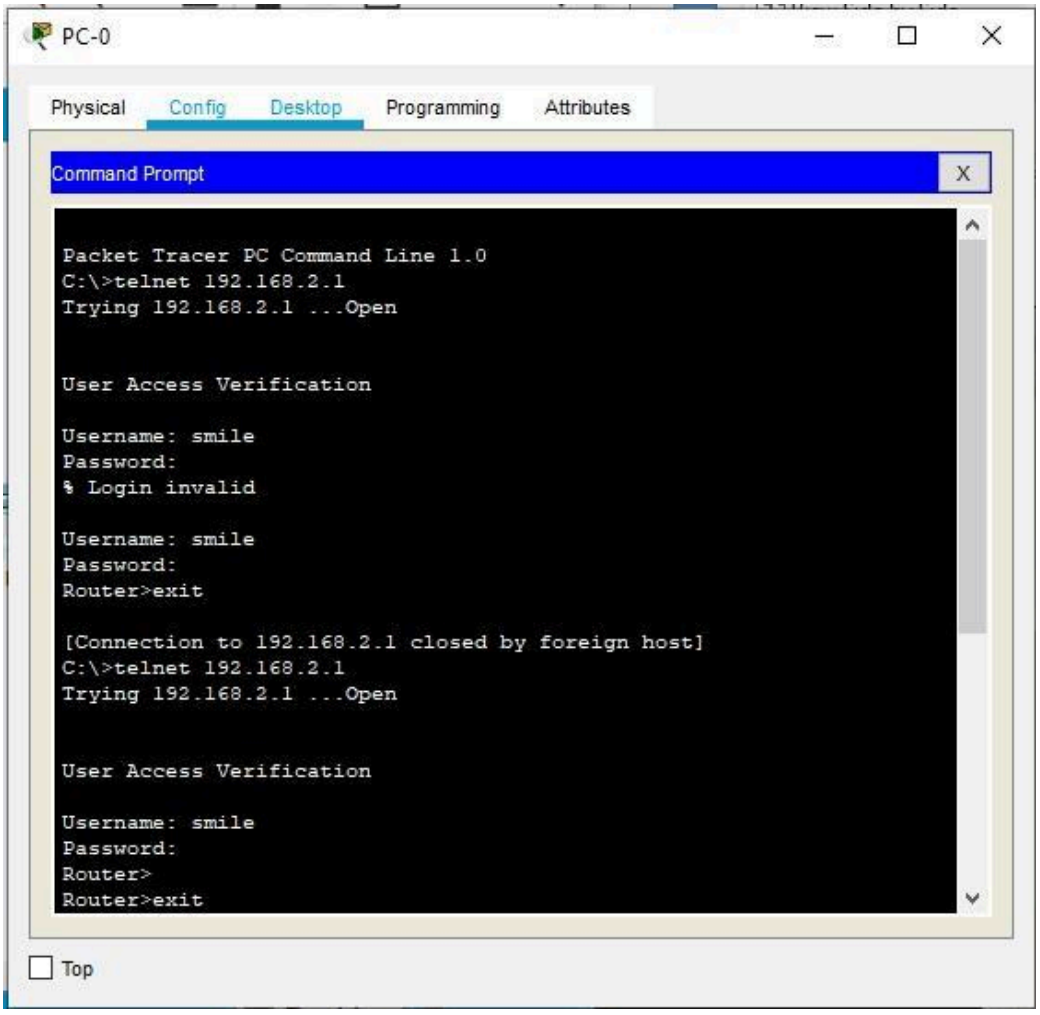
The Authentication can be done by typing the command **telnet 192.168.2.1 (**the Router IP**)** in any of the PCs
We get a prompt to type the username and password, the username and password set in TACACS are entered
Username: smile
Password: smile We get the following



In order to authenticate the RADIUS server we need to turn OFF the TACACS service



We again enter the command **telnet 192.168.2.1 (**the Router IP**)** and enter the username and password of the RADIUS server (Username: smile, Password: smile)
We get the following


PC-0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Packet Tracer PC Command Line 1.0
C:\>telnet 192.168.2.1
Trying 192.168.2.1 ...Open
User Access Verification
Username: smile
Password:
% Login invalid
Username: smile
Password:
Router>exit
[Connection to 192.168.2.1 closed by foreign host]
C:\>telnet 192.168.2.1
Trying 192.168.2.1 ...Open
User Access Verification
Username: smile
Password:
Router>
Router>exit
Top

PRACTICAL NO 3: Configuring Extended ACLs

The Cisco Access Control List (ACL) are used for filtering traffic based on a given filtering criteria on a router or switch interface. Based on the conditions supplied by the ACL, a packet is allowed or blocked from further movement.

Cisco ACLs are available for several types of routed protocols including IP, IPX, AppleTalk, XNS, DECnet, and others. However, we will be discussing ACLs pertaining to TCP/IP protocol only.

ACLs for TCP/IP traffic filtering are primarily divided into two types:

1. Standard Access Lists, and
2. Extended Access Lists

Standard Access Control Lists:
Standard IP ACLs range from 1 to 99. A Standard Access List allows you to permit or deny traffic FROM specific IP addresses. The destination of the packet and the ports involved can be anything. This is the command syntax format of a standard ACL.

access-list access-list-number {permit|deny}
{host|source source-wildcard|any} Standard ACL example: access-list
10 permit 192.168.2.0 0.0.0.255

This list allows traffic from all addresses in the range 192.168.2.0 to 192.168.2.255

Note that when configuring access lists on a router, you must identify each access list uniquely by assigning either a name or a number to the protocol's access list. There is an implicit deny added to every access list. If you entered the command:

show access-list 10

The output looks like: access-list 10 permit 192.168.2.0 0.0.0.255
access-list 10 deny any

Standard Access Control Lists:

Standard IP ACLs range from 1 to 99. A Standard Access List allows you to permit or deny traffic FROM specific IP addresses. The destination of the packet and the ports involved can be anything. This is the command syntax format of a standard ACL.

access-list access-list-number {permit|deny} {host|source source-wildcard|any} Standard ACL example: access-list 10 permit 192.168.2.0 0.0.0.255

This list allows traffic from all addresses in the range 192.168.2.0 to 192.168.2.255

Note that when configuring access lists on a router, you must identify each access list uniquely by assigning either a name or a number to the protocol's access list. There is an implicit deny added to every access list. If you entered the command:

show access-list 10

The output looks like: access-list 10 permit 192.168.2.0 0.0.0.255
access-list 10 deny any

Extended Access Control Lists:

Extended IP ACLs allow you to permit or deny traffic from specific IP addresses to a specific destination IP address and port. It also allows you to have granular control by specifying controls for different types of protocols such as ICMP, TCP, UDP, etc within the ACL statements. Extended IP ACLs range from 100 to 199. In Cisco IOS Software Release 12.0.1, extended ACLs began to use additional numbers (2000 to 2699). The syntax for IP Extended ACL is given below:

access-list access-list-number {deny | permit} protocol source source-wildcard destination destination-wildcard [precedence precedence]
Note that the above syntax is simplified, and given for general understanding only.

Extended ACL example:

access-list 110 - Applied to traffic leaving the office (outgoing) access-list
110 permit tcp 92.128.2.0 0.0.0.255 any eq 80
ACL 110 permits traffic originating from any address on the 92.128.2.0 network. The 'any' statement means that the traffic is allowed to have any destination address with the limitation of going to port 80. The value of
0.0.0.0/255.255.255.255 can be specified as 'any'.

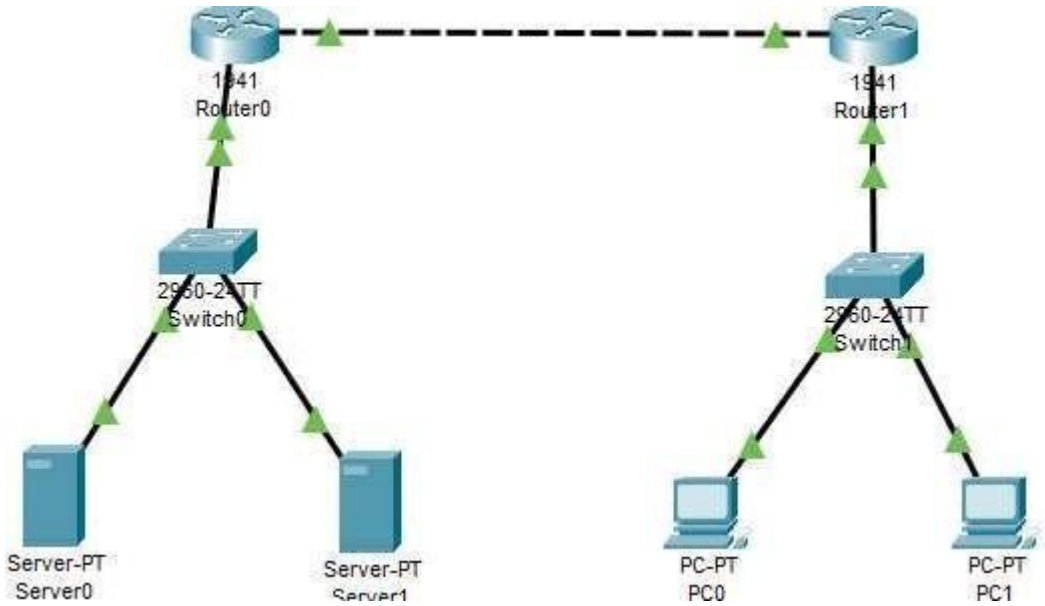
Applying an ACL to a router interface:
After the ACL is defined, it must be applied to the interface (inbound or outbound). The syntax for applying an ACL to a router interface is given below:

interface <interface>
ip access-group {number|name} {in|out}

An Access List may be specified by a name or a number. "in" applies the ACL to the inbound traffic, and "out" applies the ACL on the outbound traffic.

Example: To apply the standard ACL created in the previous example, use the following commands:
Rouer(config)#interface serial0
Rouer(config-if)#ip access-group 10 out

Consider the following topology



Let us consider the following Address table to configure the network devices:

Device	Interface	IP Address	Subnet Mask	Default gateway	Switch Port
Server 0	NA	192.168.1.3	255.255.255.0	192.168.1.1	Switch 0 F/06
Server 1	NA	192.168.1.2	255.255.255.0	192.168.1.1	Switch 0 F0/1
PC 0	NA	192.168.3.2	255.255.255.0	192.168.3.1	Switch 1 F/06
PC 1	NA	192.168.3.3	255.255.255.0	192.168.3.1	Switch 1 F0/1
Router 0	GE0/0	192.168.1.1	255.255.255.0	NA	Switch 0 F0/5
	GE0/1	192.168.2.2	255.255.255.0	NA	GE0/1
Router 1	GE0/0	192.168.3.1	255.255.255.0	NA	Switch 1 F0/5
	GE0/1	192.168.2.2	255.255.255.0	NA	GE 0/1

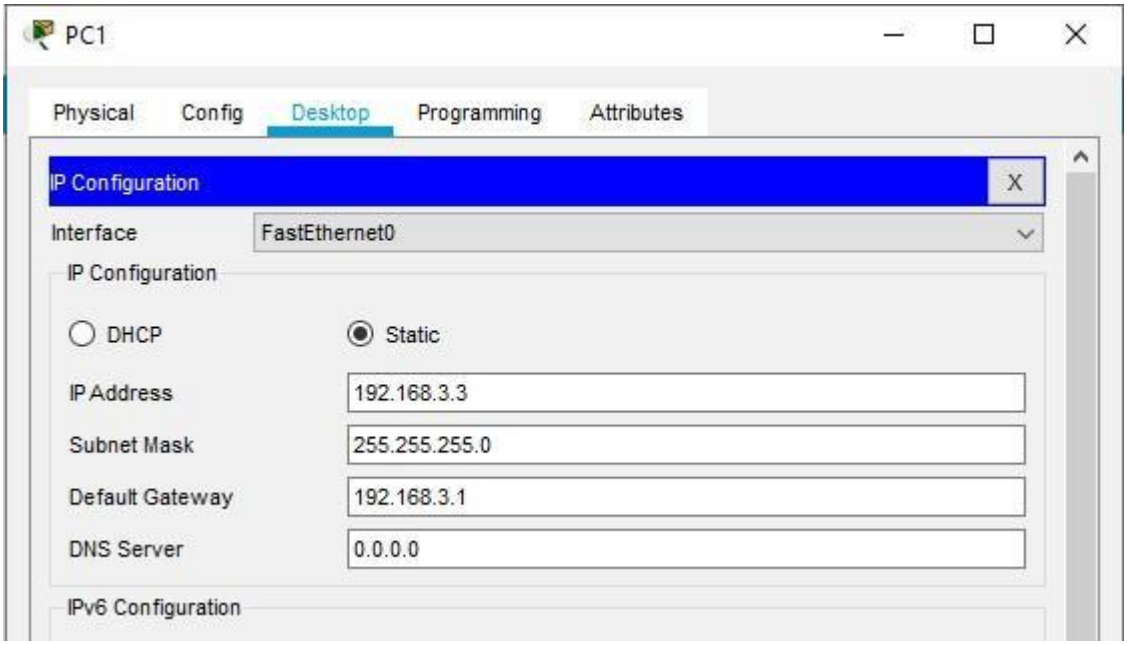
Part 1: Configure, Apply and Verify an Extended Numbered ACL

Configuring Server 0

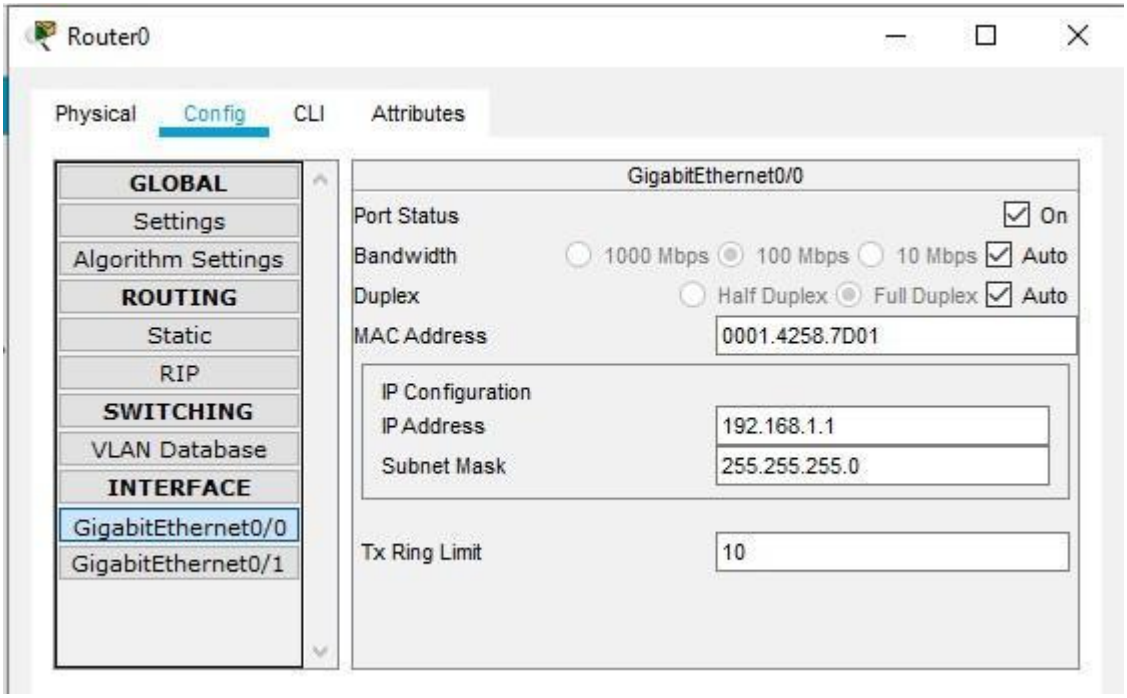
Configuring Server 1

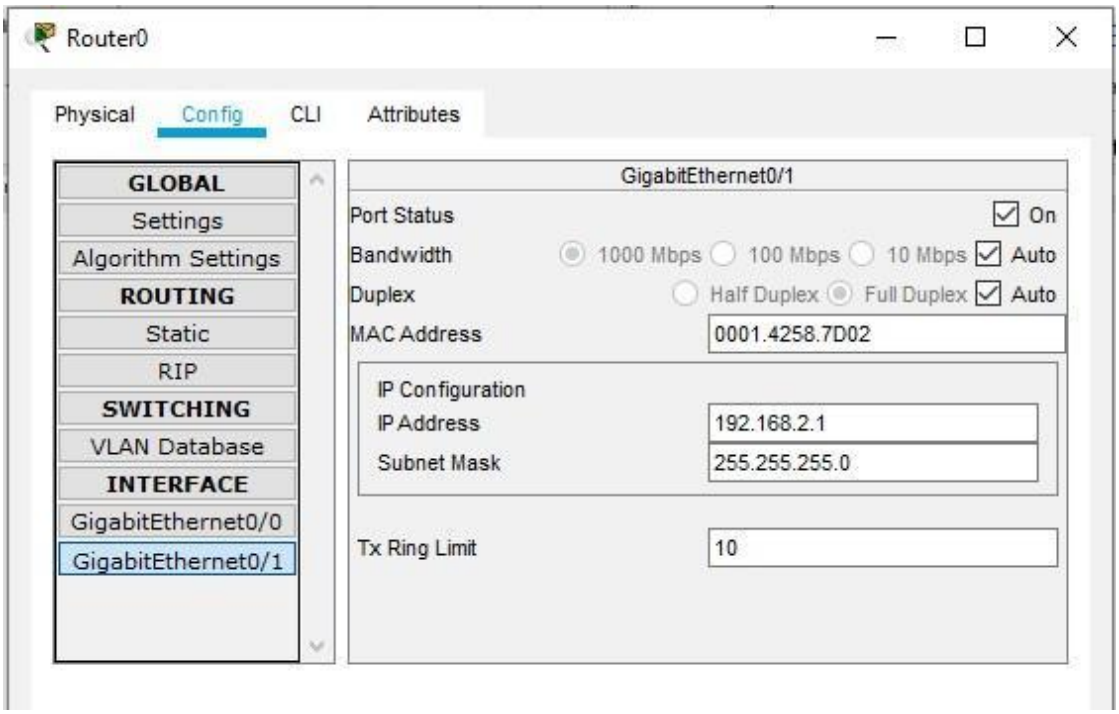
Configuring PC 0

Configuring PC 1

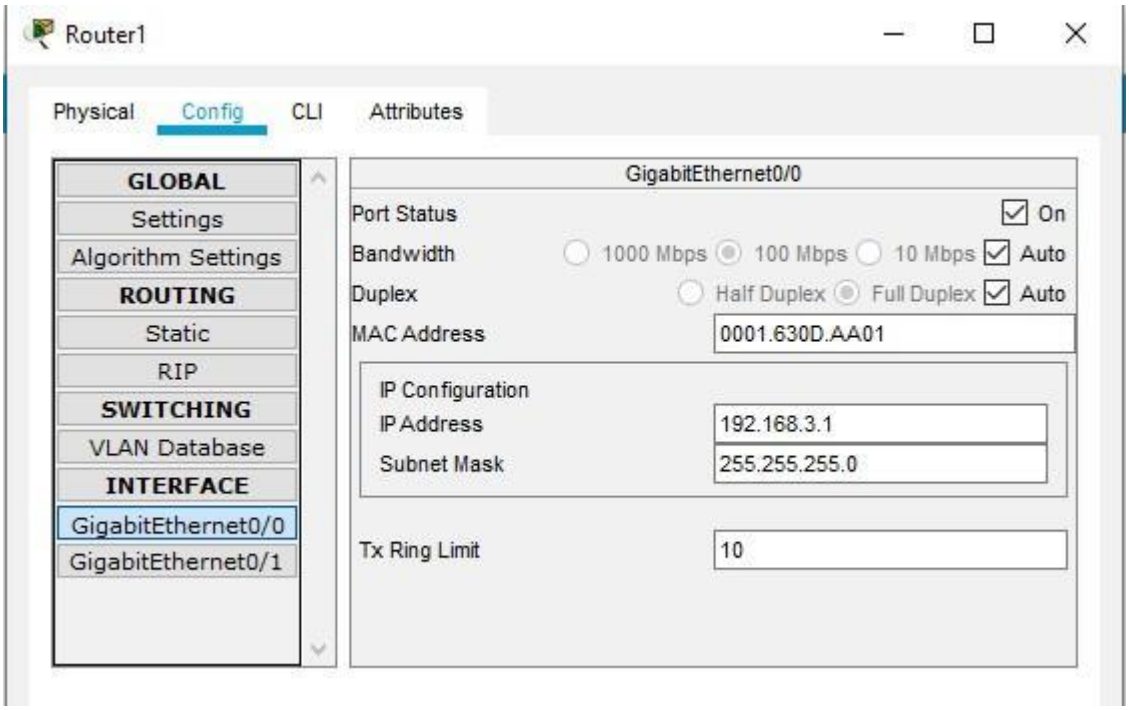


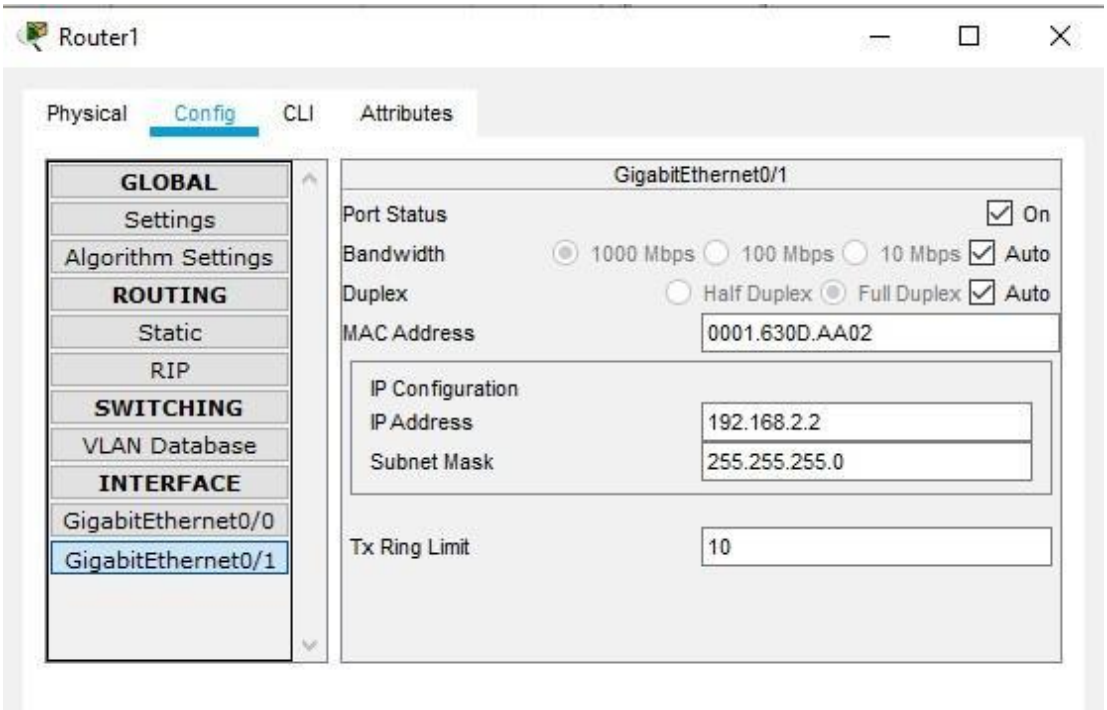
Configuring Router 0



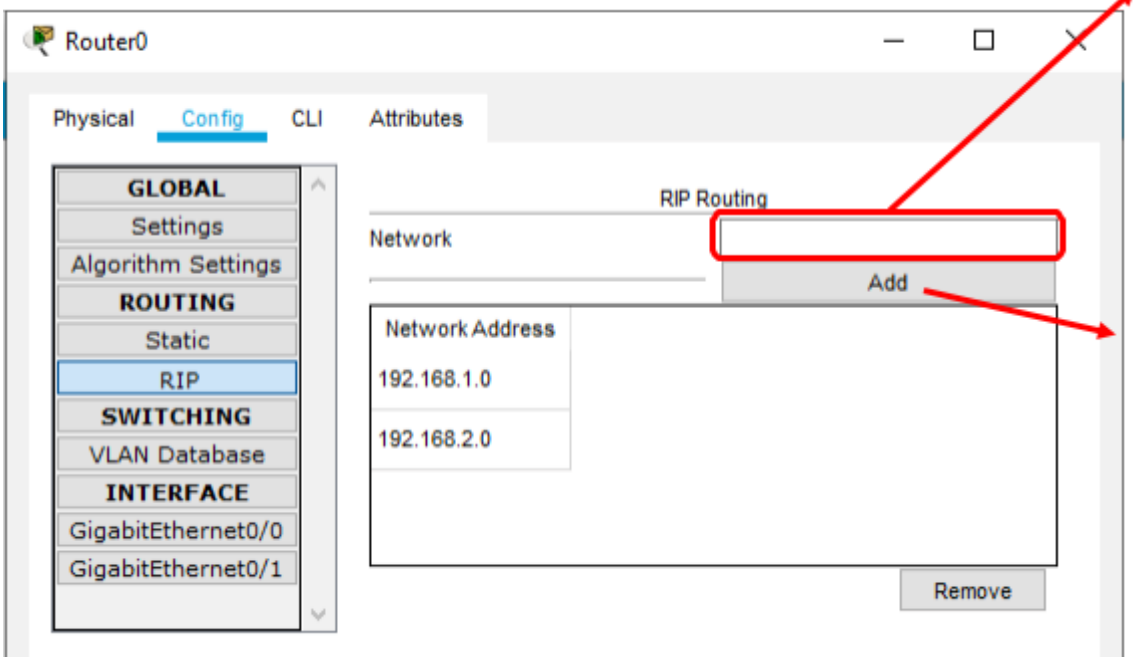


Configuring Router 1





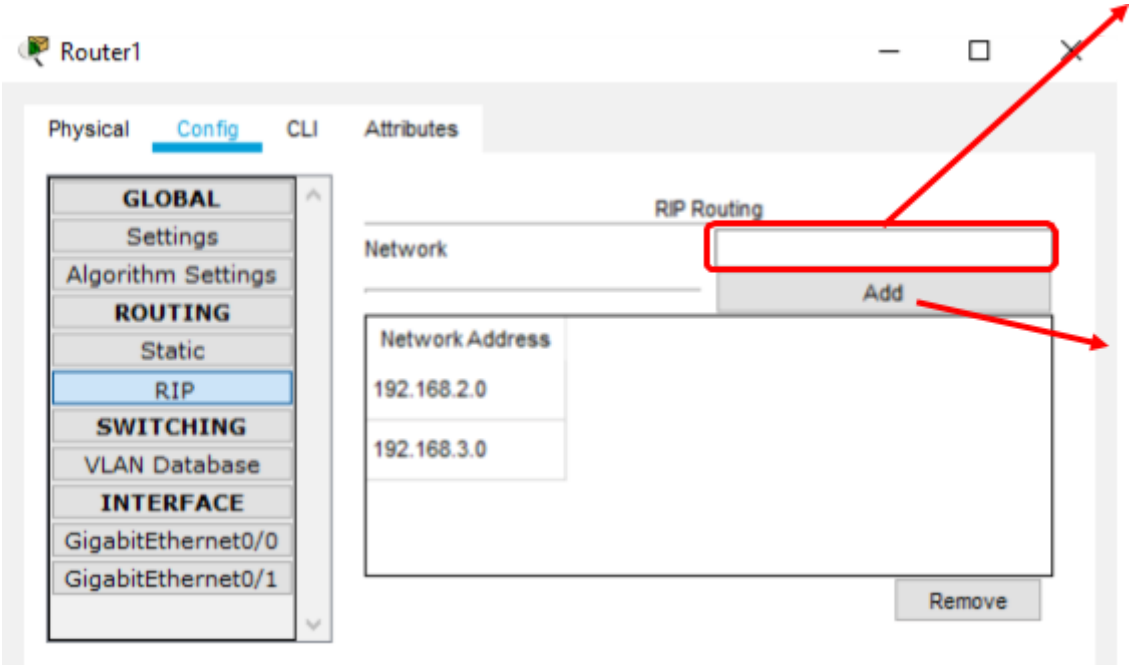
Set the RIP protocol on both the Type the IP address



Routers as follows

Click Add button

Type the IP address



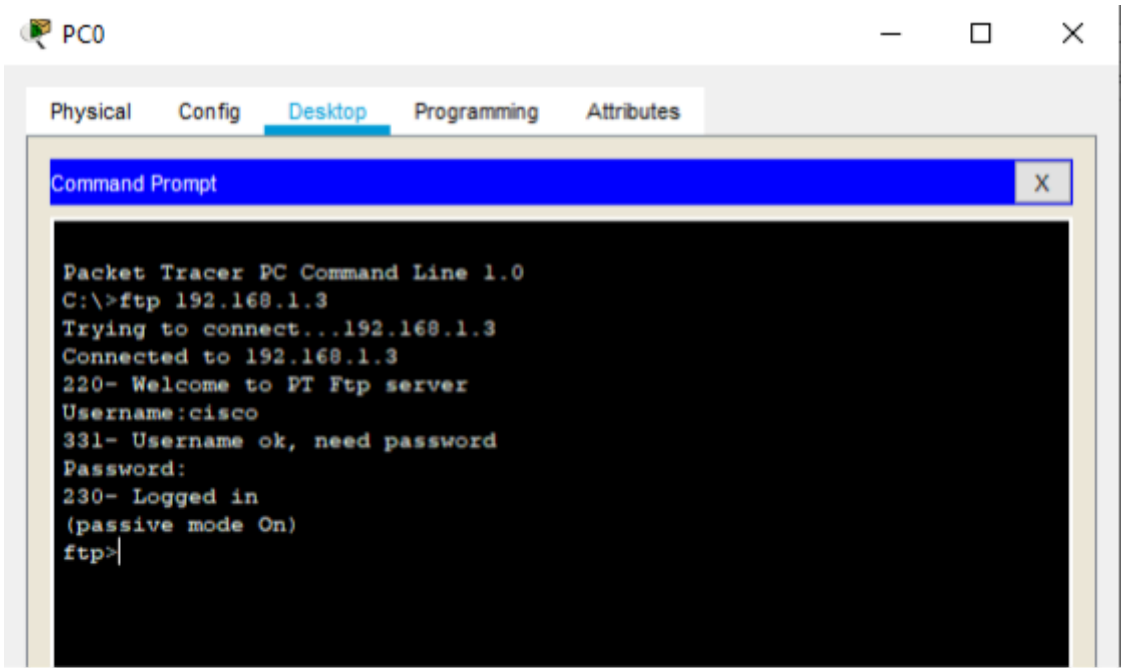
Cli

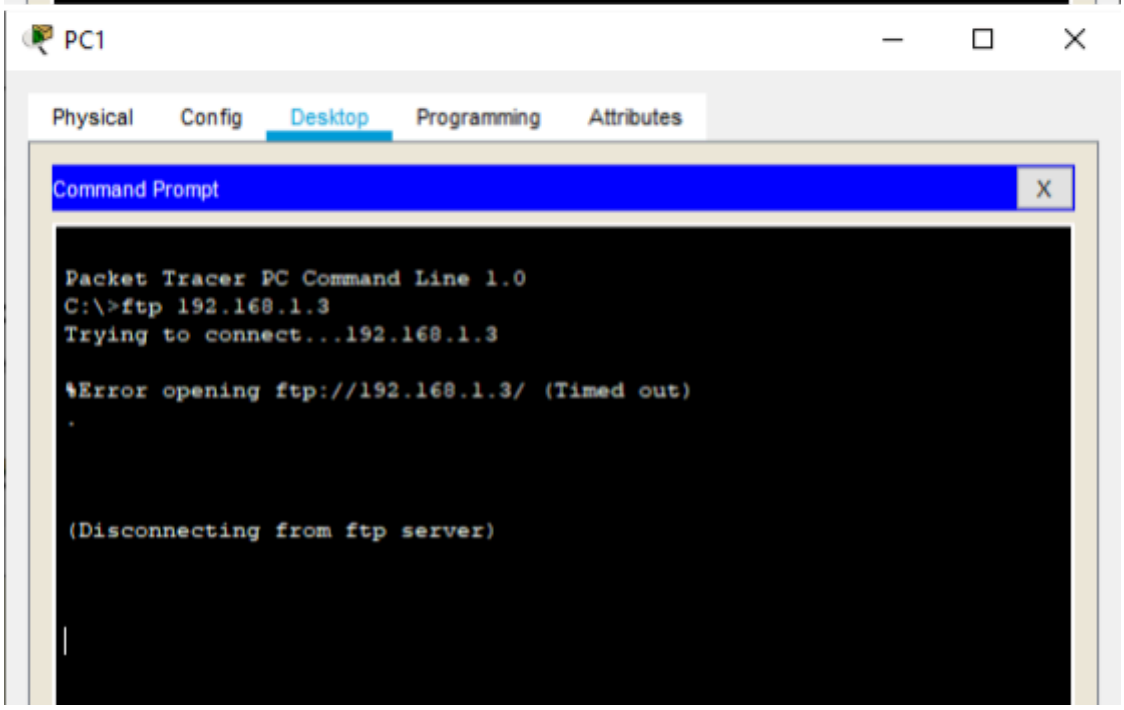
Check the connectivity between all the devices in the topology.

Type the following commands in Router1

```
Router#configure terminal
Router(config)#access-list 100 permit tcp host 192.168.3.2 host 192.168.1.3 eq ftp
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip access-group 100 out
Router(config-if)#exit Router(config)#
```

Now verify the ftp (ftp 192.168.1.3) command from both the PCs, one would be successful (PC0) and other (PC1) would fail



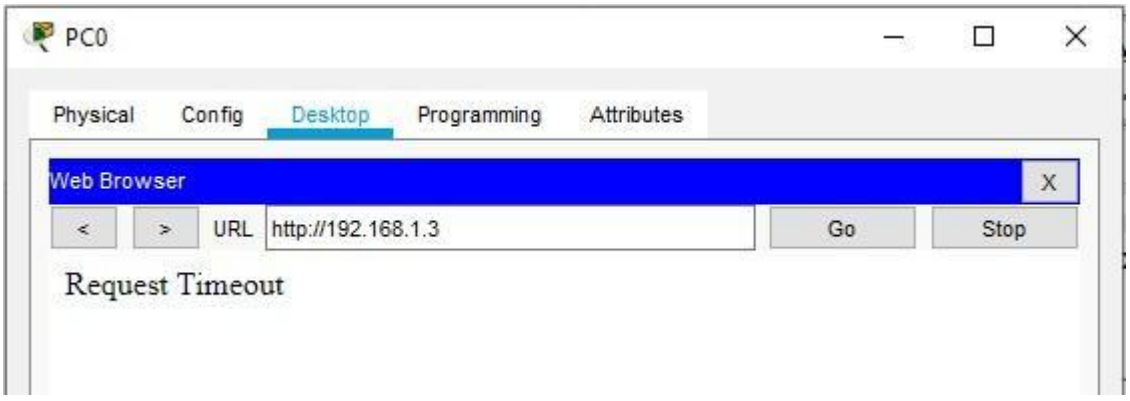


Part 2: Configure, Apply and Verify an Extended Named ACL
We use the same topology for this case
Type the following command in the CLI mode of Router1

Router> Router>enable router
Router#configure terminal
Router(config)#ip access-list extended SMILE

Router(config-ext-nacl)#permit tcp host 192.168.3.3 host 192.168.1.3 eq www
Router(config-ext-nacl)#exit
Router(config)#
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip access-group SMILE out
Router(config-if)#exit Router(config)#

Now verify the www (192.168.1.3) command from both the PCs browser, one would be successful (PC1) and other (PC0) would fail



Hence Extended Numbered ACLs as well as Extended Named ACLs have been verified.

PRACTICAL NO 4: Configure IP ACLs to Mitigate Attacks and Configuring IPv6 ACLs

Access Control Lists (ACLs)

Network administrators must figure out how to deny unwanted access to the network while allowing internal users appropriate access to necessary services. Although security tools, such as passwords, callback equipment, and physical security devices are helpful, they often lack the flexibility of basic traffic filtering and the specific controls most administrators prefer.

For example, a network administrator may want to allow users access to the Internet, but not permit external users telnet access into the LAN.

Routers provide basic traffic filtering capabilities, such as blocking Internet traffic, with access control lists (ACLs).

An ACL is a sequential list of permit or deny statements that apply to addresses or upper-layer protocols.

The router examines each packet to determine whether to forward or drop it, based on the conditions specified in the ACL. Some ACL decision points are:

1) IP source address
2) IP destination addresses
3) UDP or TCP protocols
4) Upper-layer (TCP/UDP) port numbers

ACLs must be defined on a:

1) Per-protocol (IP, IPX, AppleTalk)
2) Per direction (in or out)
3) Per port (interface) basis.
4) ACLs control traffic in one direction at a time on an interface.
5) A separate ACL would need to be created for each direction, one for inbound and one for outbound traffic.
6) Finally every interface can have multiple protocols and directions defined.

An ACL is a group of statements that define whether packets are accepted or rejected coming into an interface or leaving an interface.

1) ACL statements operate in sequential, logical order (top down).
2) If a condition match is true, the packet is permitted or denied and the rest of the ACL statements are not checked.
3) If all the ACL statements are unmatched, an implicit "deny any" statement is placed at the end of the list by default. (not visible) When first learning how to create ACLs, it is a good idea to add the implicit deny at the end of ACLs to reinforce the dynamic presence of the command line.

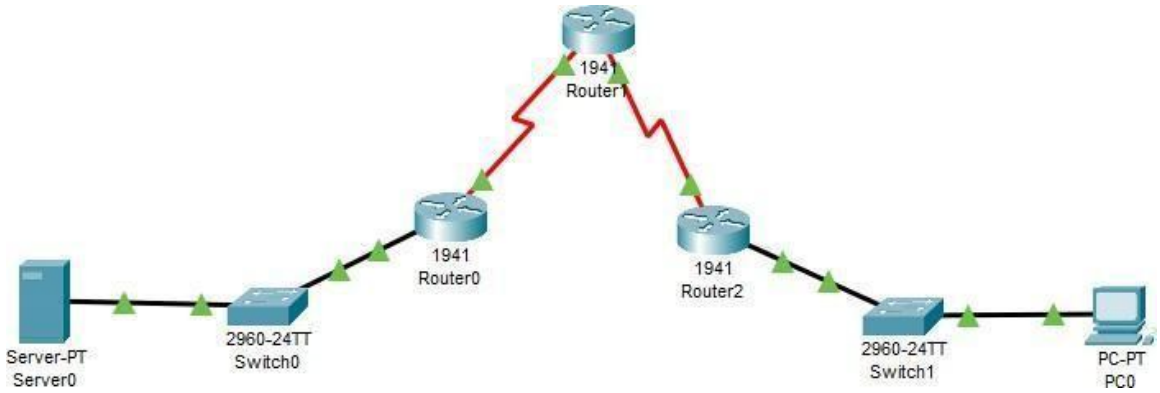
Standard IP ACLs
Can only filter on source IP addresses

Extended IP ACLs Can filter on:

1) Source IP address
2) Destination IP address
3) Protocol (TCP, UDP)
4) Port Numbers (Telnet – 23, http – 80, etc.) and other parameters

An access list is a sequential series of commands or filters. These lists tell the router what types of packets to: accept or deny Acceptance and denial can be based on specified conditions. ACLs applied on the router's interfaces

We use the following topology to study the present case



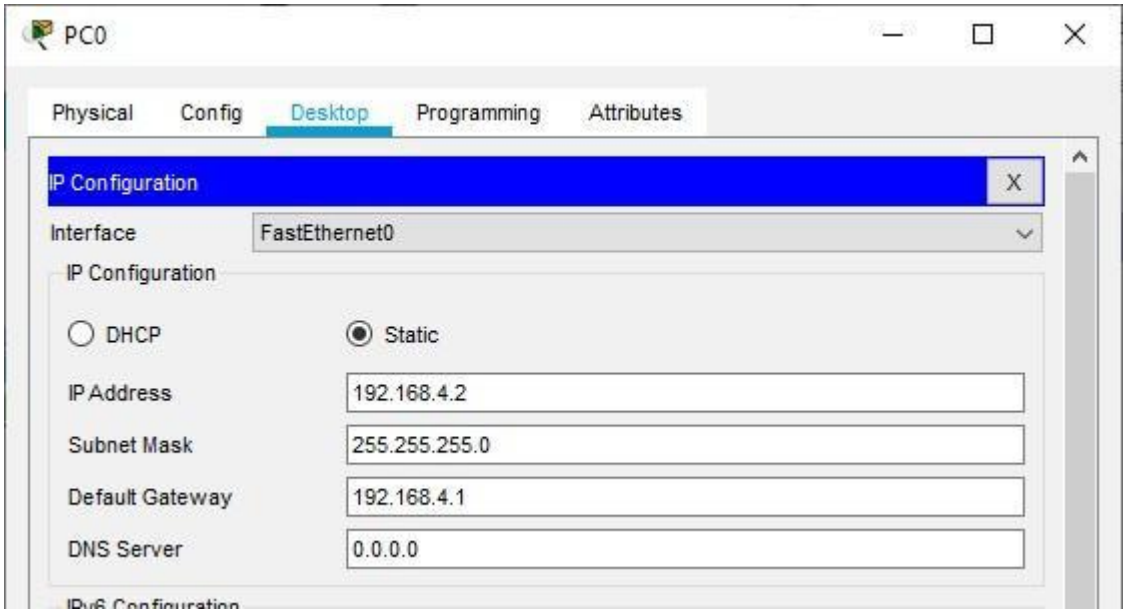
Let us consider the following Address table to configure the network devices:

Device	Interface	IP Address	Subnet Mask	Default gateway	Switch Port
PC 0	NA	192.168.4.2	255.255.255.0	192.168.4.1	Switch1 F0/1
Server0	NA	192.168.1.2	255.255.255.0	192.168.1.1	Switch0 F0/1
Router0	GE0/0	192.168.1.1	255.255.255.0	NA	Switch0 F0/5
	S0/1/0	192.168.2.1	255.255.255.0	NA	NA
Router1	S0/1/0	192.168.2.2	255.255.255.0	NA	NA
	S0/1/1	192.168.3.1	255.255.255.0	NA	NA
Router2	S0/1/1	192.168.3.2	255.255.255.0	NA	NA
	GE0/0	192.168.4.1	255.255.255.0	NA	Switch1 F0/5

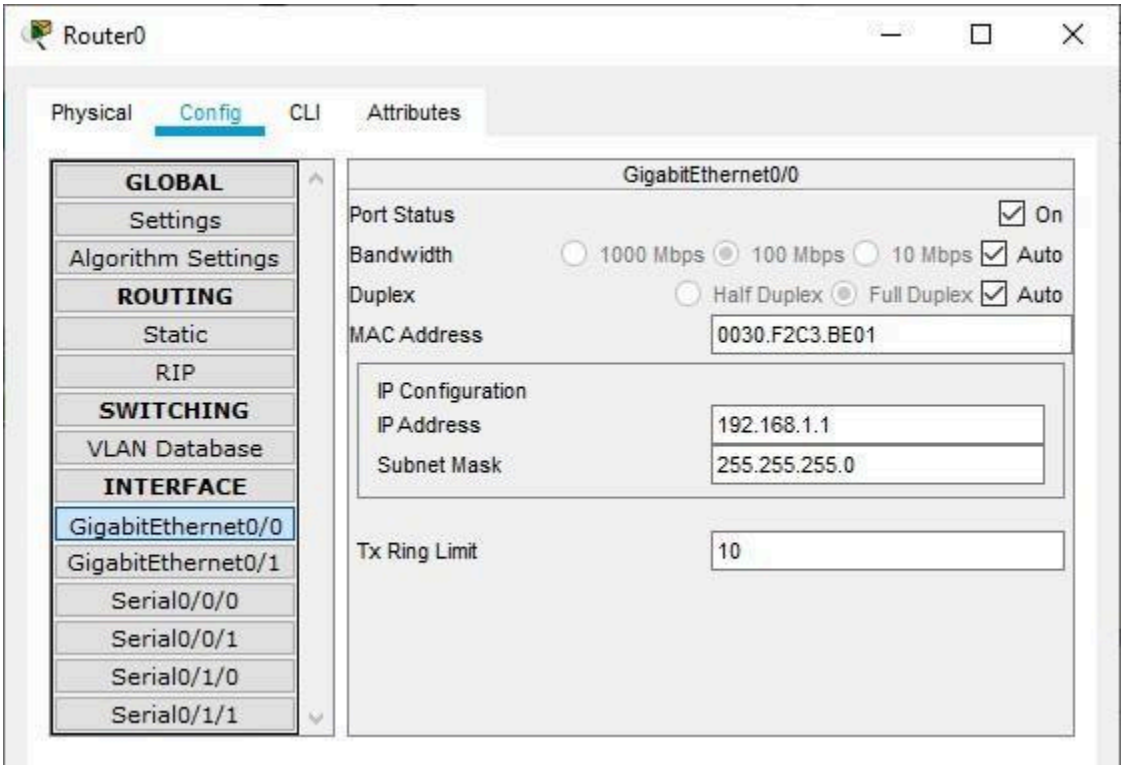
Part 1 - Verify connectivity among devices before firewall configuration

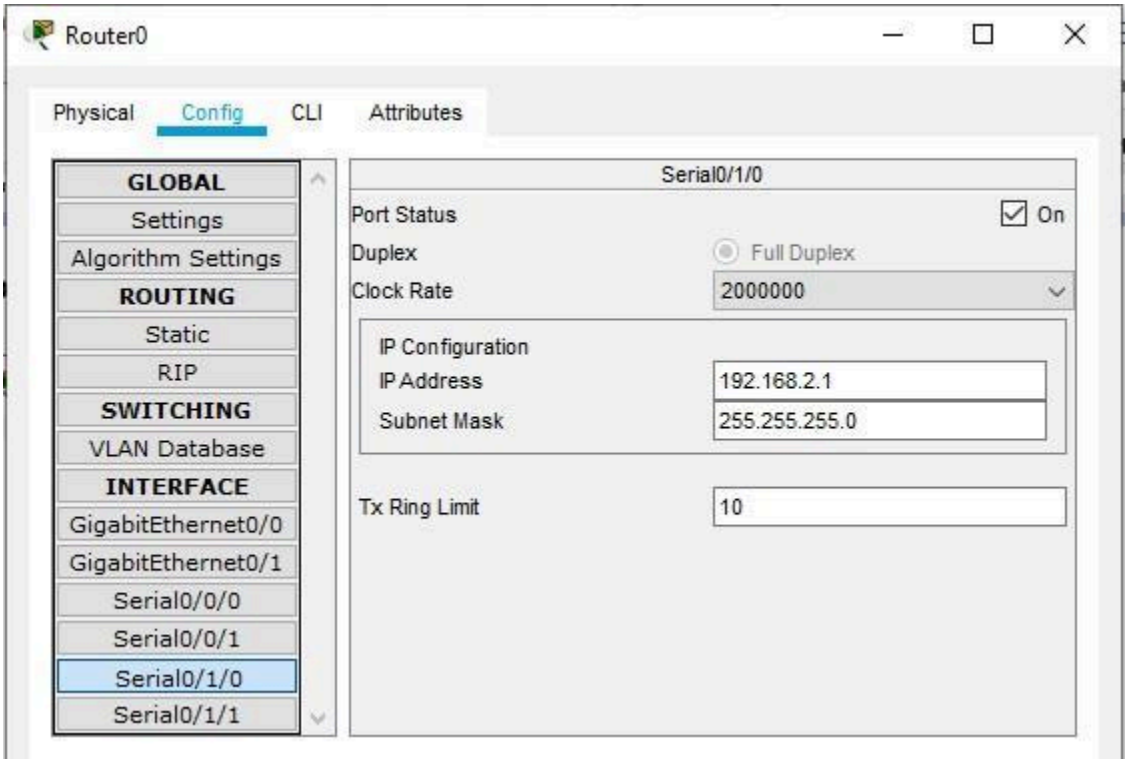
Configuring Server 0

Configuring PC0

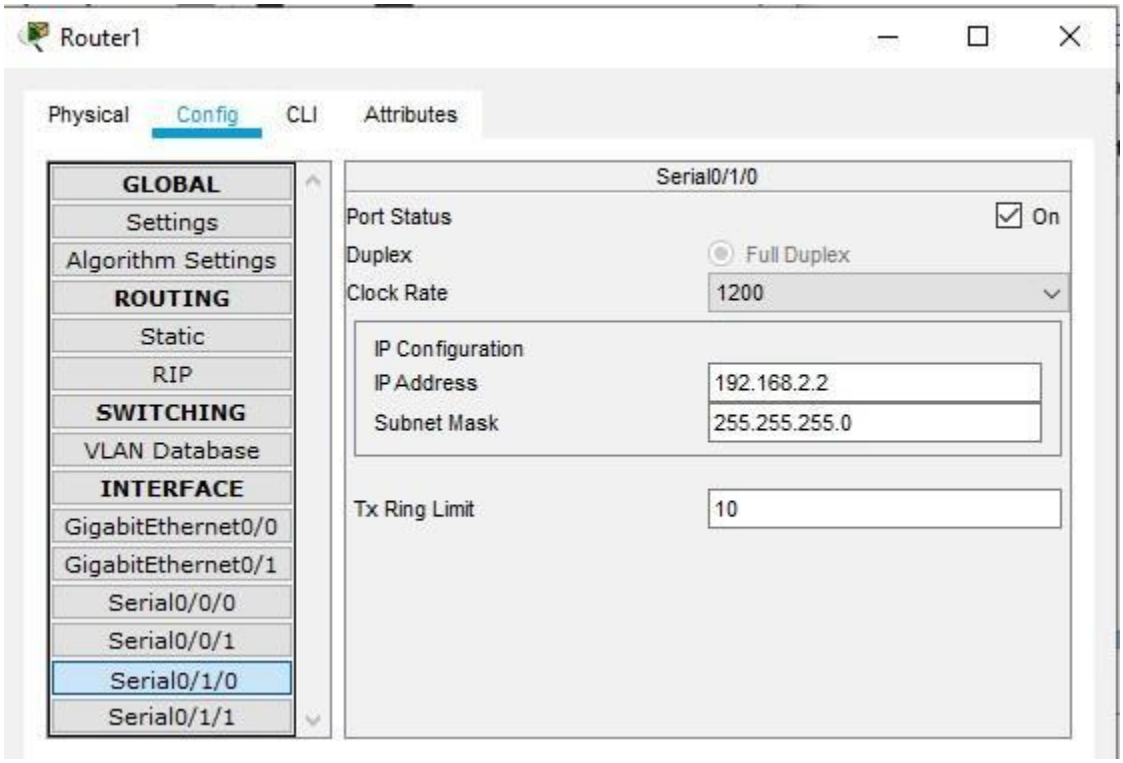


Configuring Router0

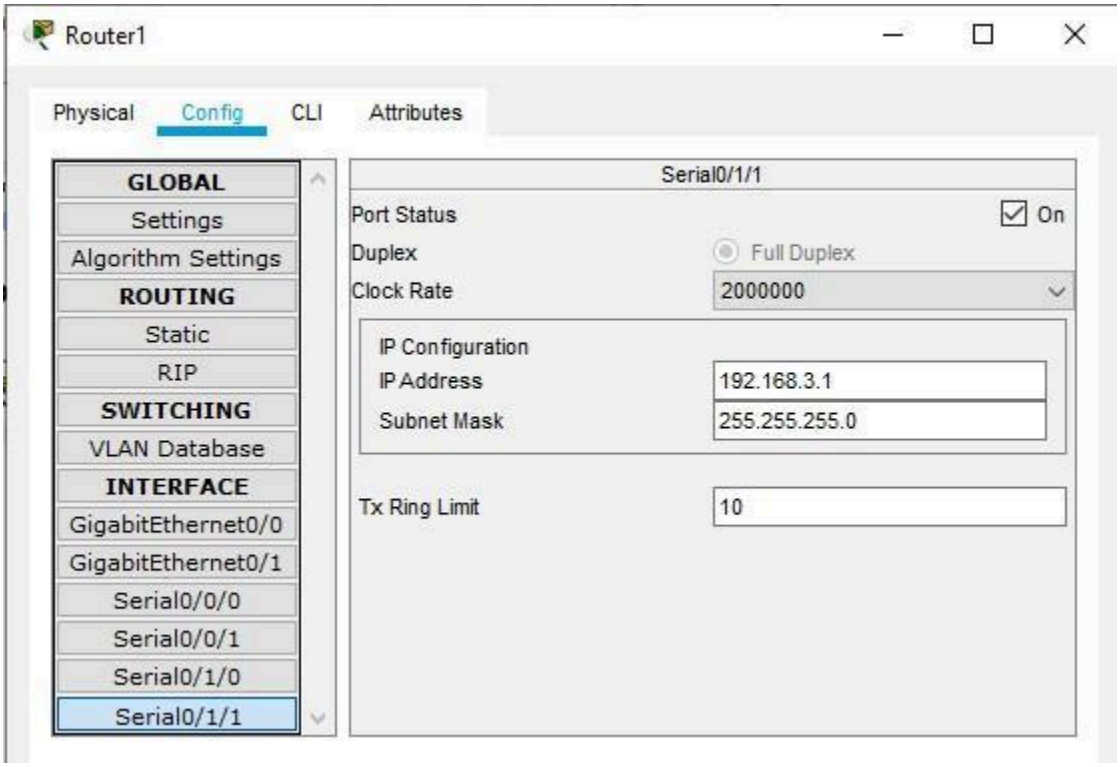


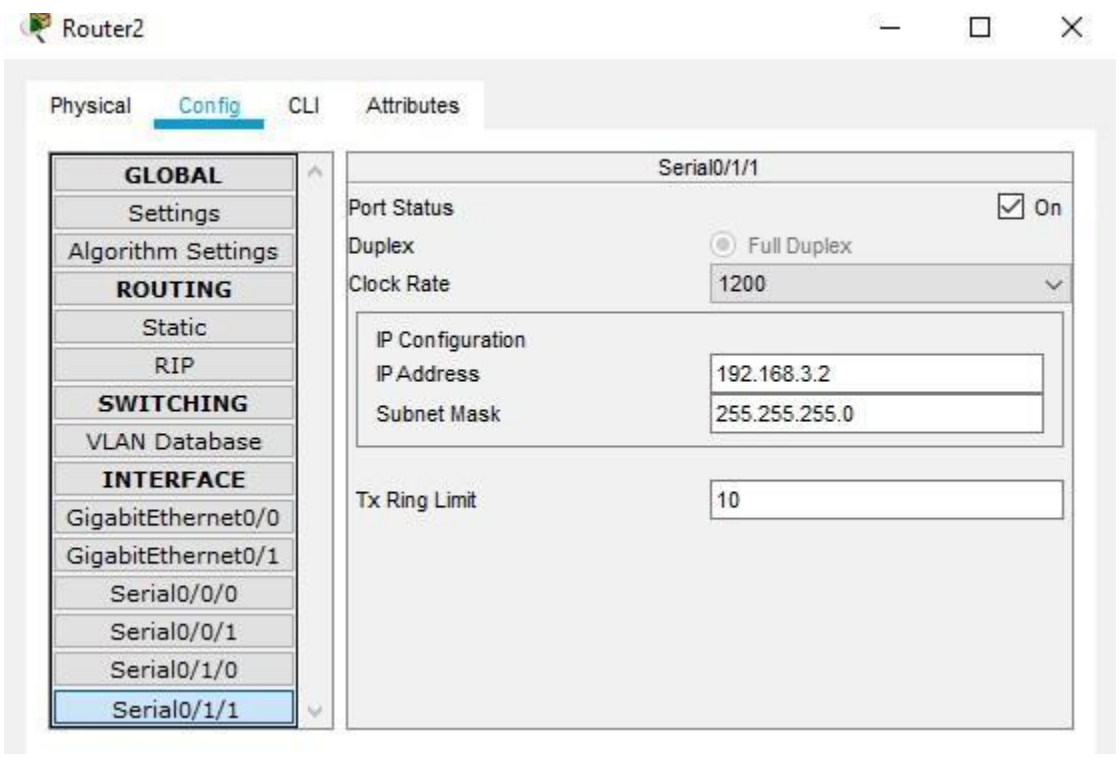


Configuring Router1

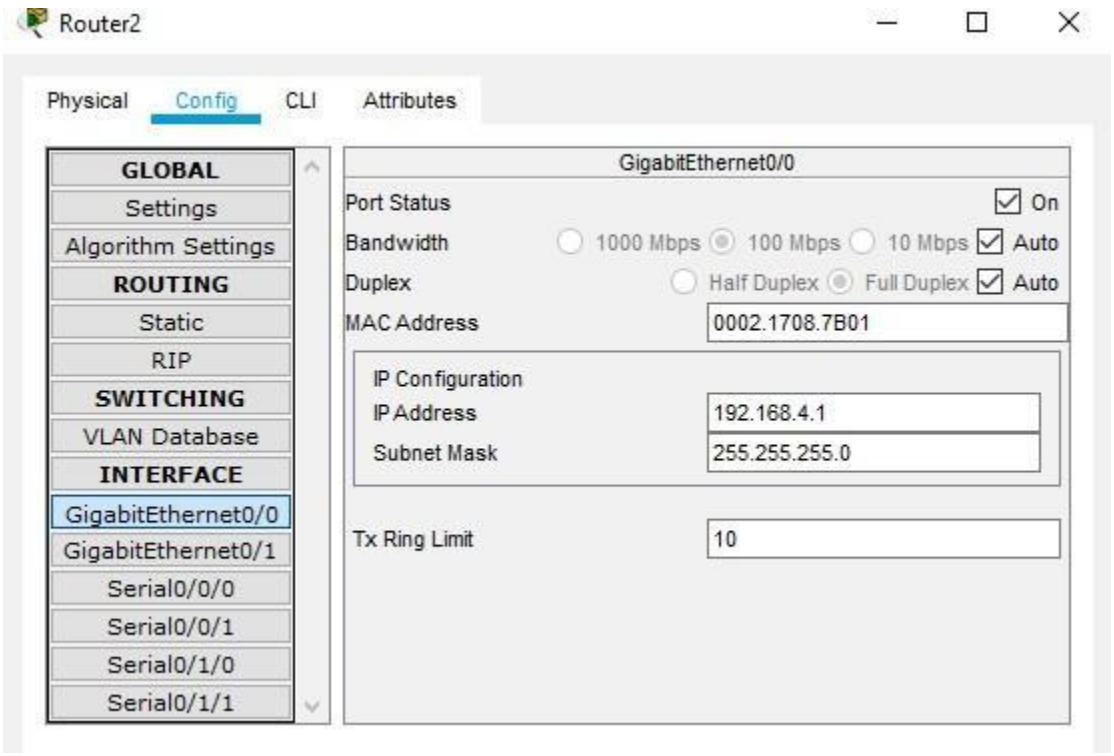


Configuring Router2



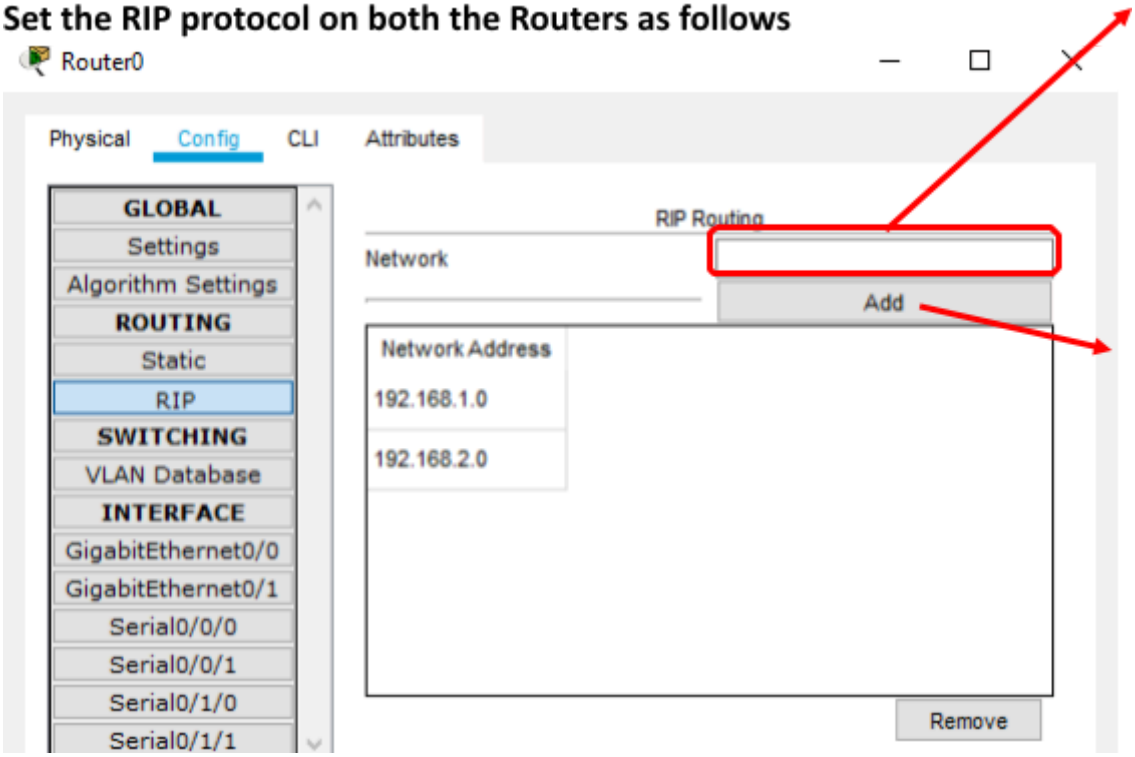


Configuring Router3



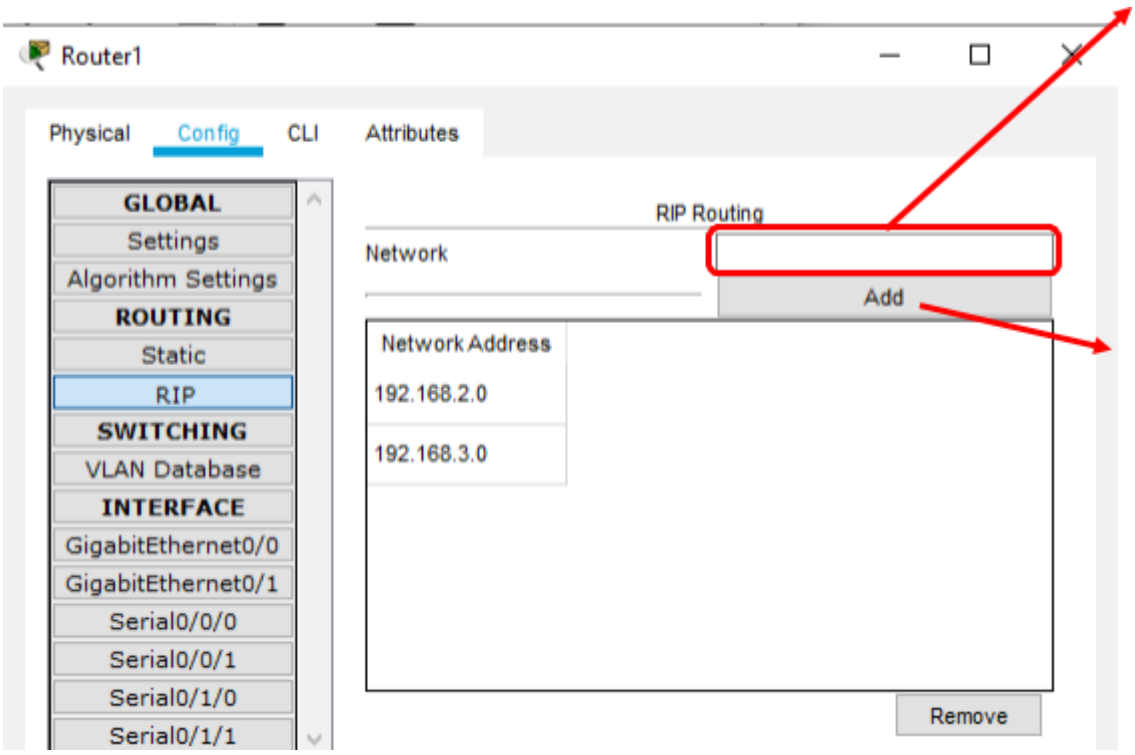
Set the RIP protocol on both the Routers as follows

Type the IP address



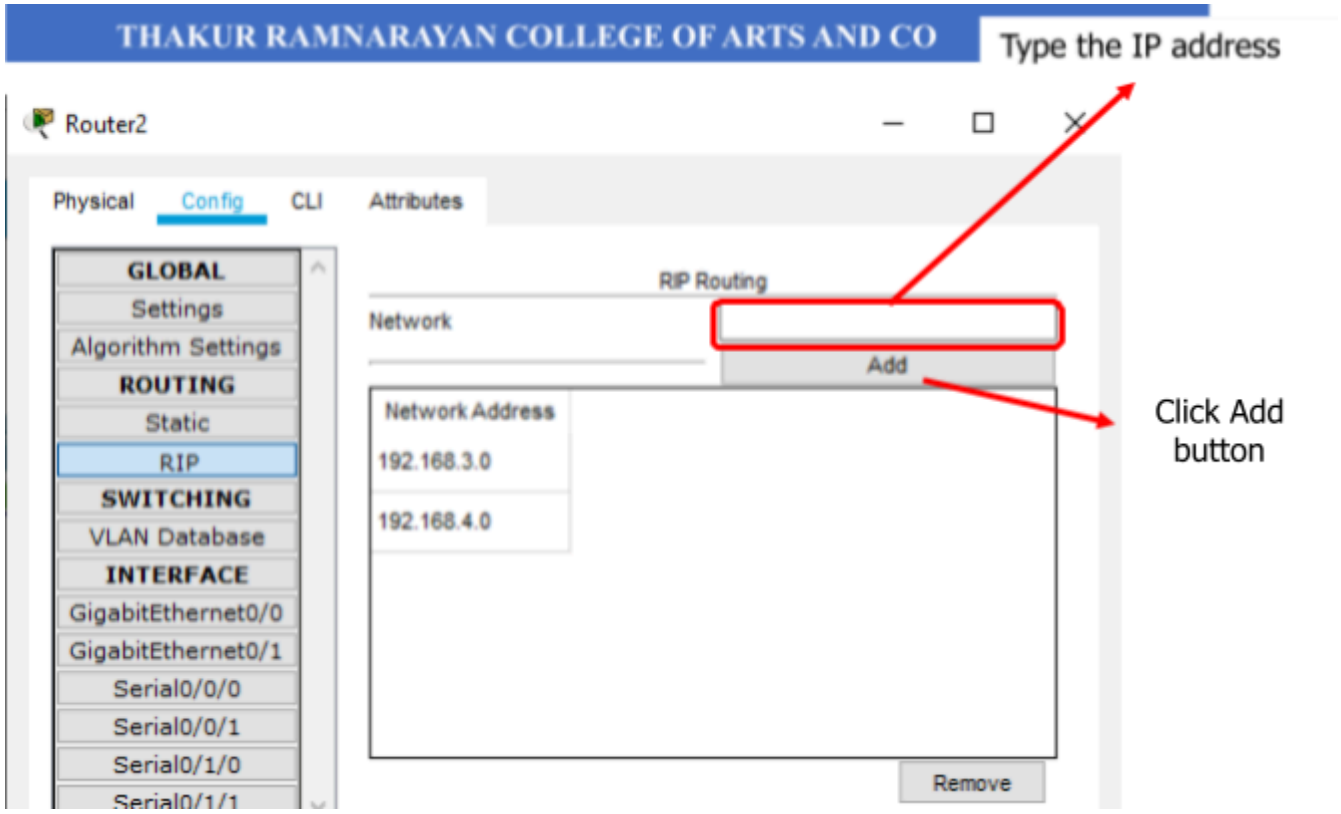
Cli

Type the IP address

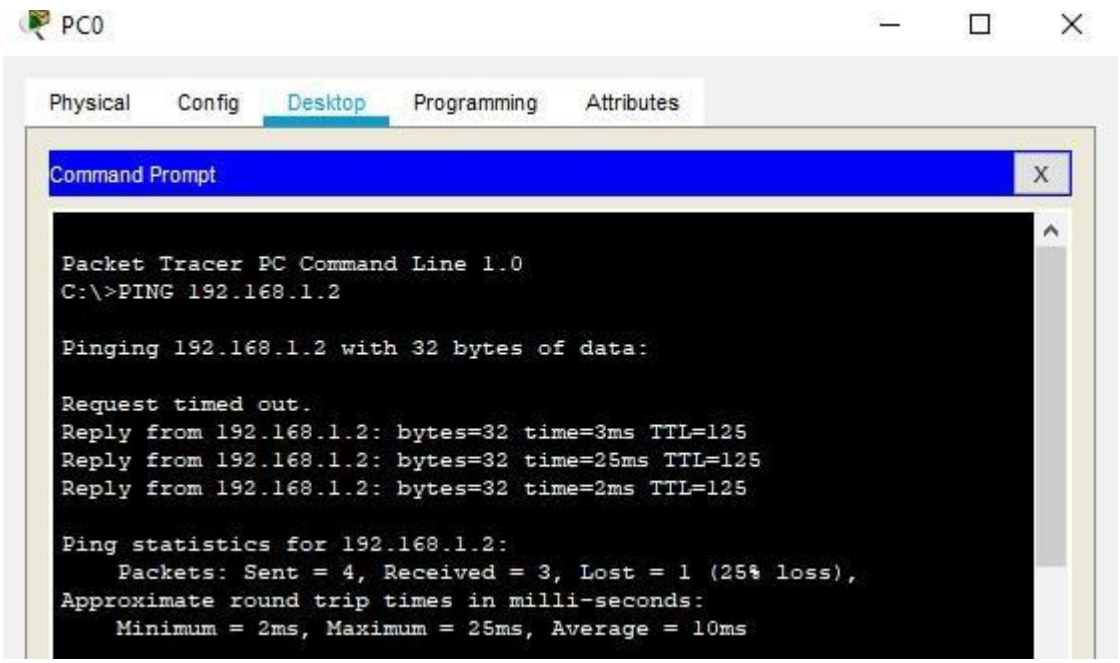


Cli

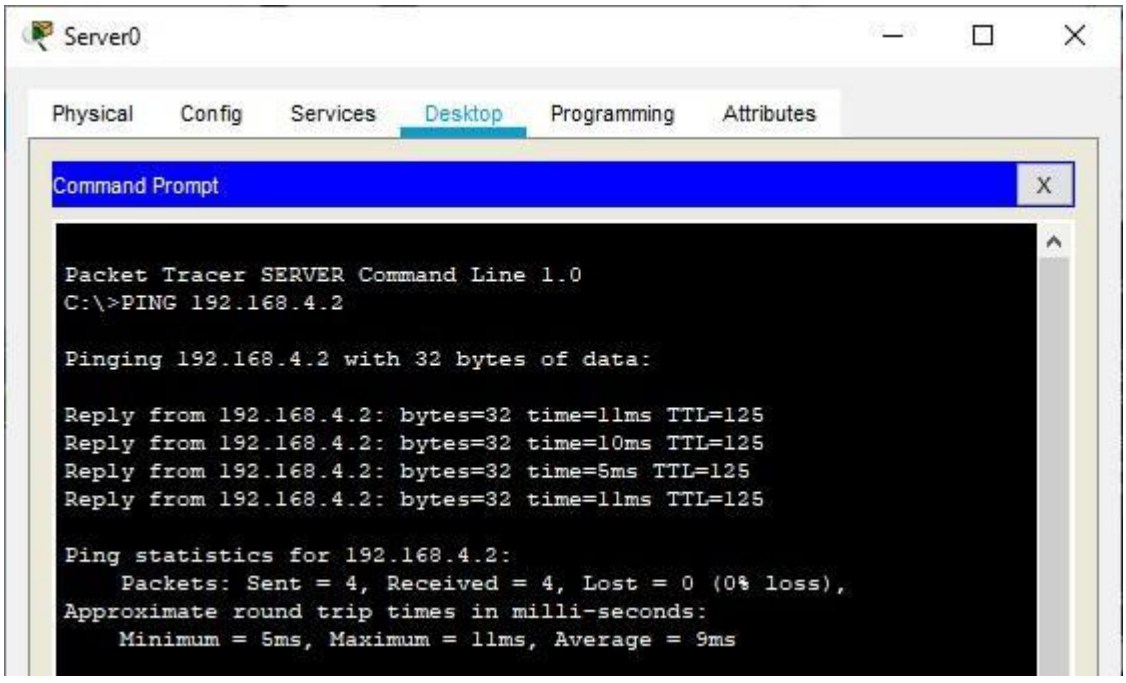
THAKUR RAMNARAYAN COLLEGE OF ARTS AND CO



We can now verify the connectivity by pinging Server from PC



We can now verify the connectivity by pinging PC from Server



Part 2 – Secure Access to Routers

We configure ACL 10 to block all remote access to the Routers and allow remote access only from PC. We type the following commands in all the Routers (Router0, Router1, and Router2). This part is divided in 2 subparts

Set up the SSH protocol

Enter the following commands in CLI mode of Router0

```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router0
Router0(config)#
Router0(config)#crypto key generate rsa
Router0(config)#line vty 0 4
Router0(config-line)#transport input ssh
Router0(config-line)#login local
Router0(config-line)#exit
Router0(config)#username SSHadmin privilege 15 password ismail
Router0(config)#exit Router0#
```

Enter the following commands in CLI mode of Router1

```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router1
Router1(config)#
Router1(config)#crypto key generate rsa
Router1(config)#line vty 0 4
Router1(config-line)#transport input ssh
Router1(config-line)#login local
Router1(config-line)#exit
Router1(config)#username SSHadmin privilege 15 password ismail
Router1(config)#exit Router1#
```

Enter the following commands in CLI mode of Router2

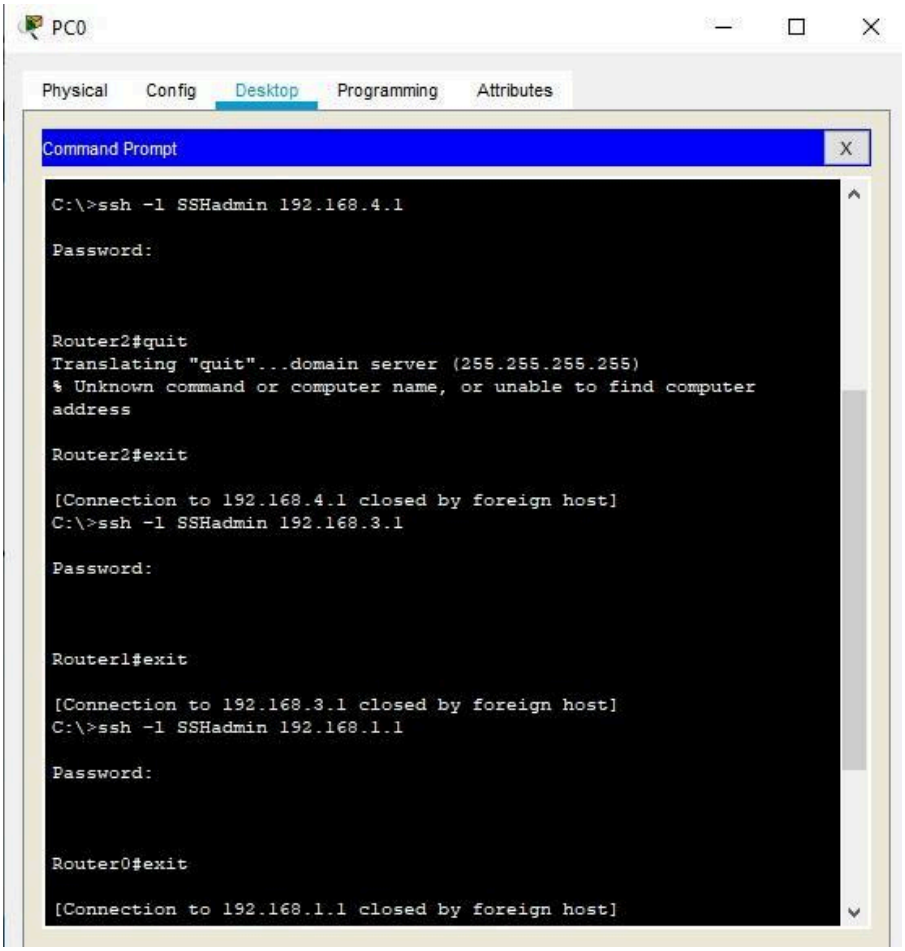
```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router2
Router2(config)#
Router2(config)#crypto key generate rsa
Router2(config)#line vty 0 4
Router2(config-line)#transport input ssh
Router2(config-line)#login local
Router2(config-line)#exit
Router2(config)#username SSHadmin privilege 15 password ismail
Router2(config)#exit Router2#
```

Create an ACL 10 to permit remote access to PC only

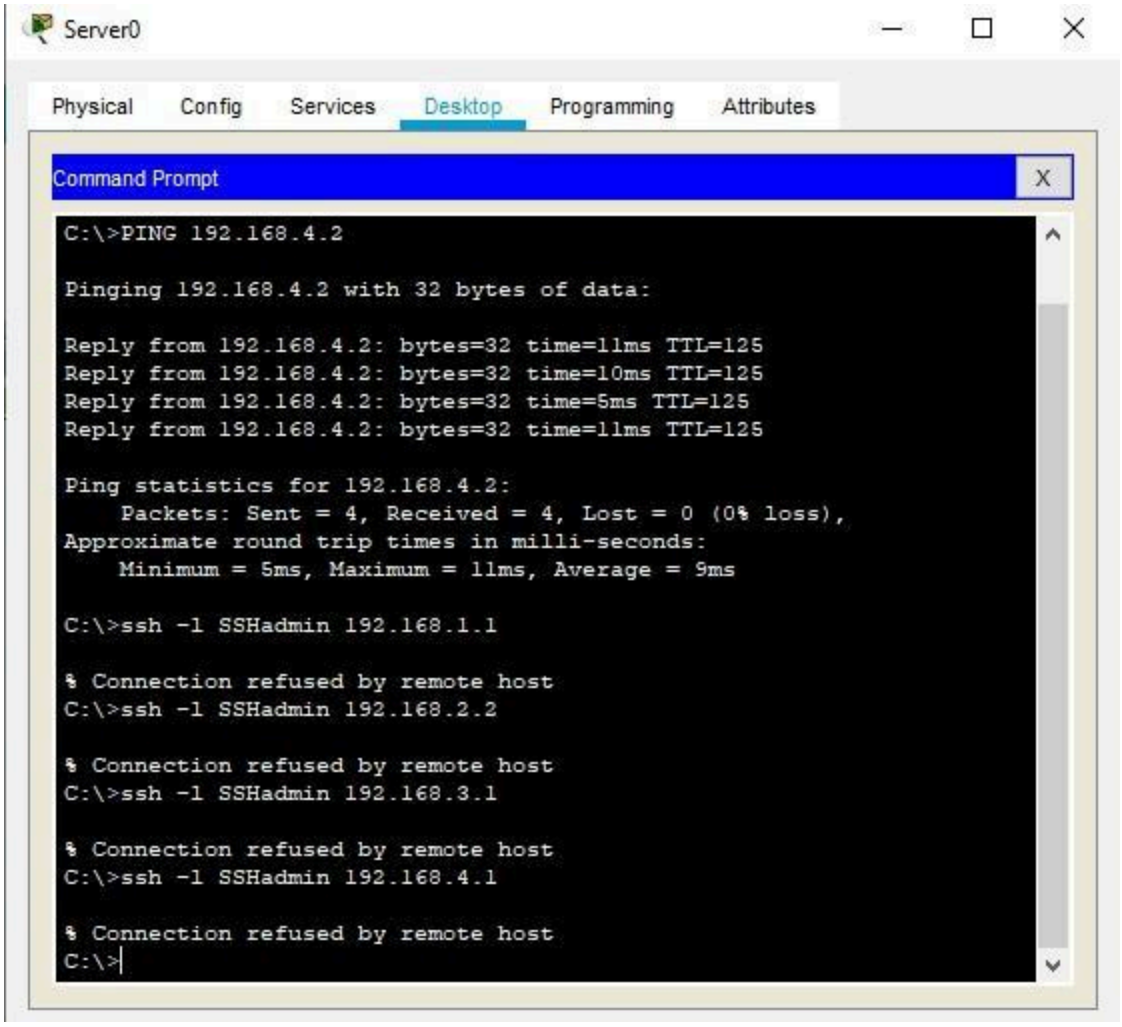
Enter the following commands in CLI mode of all Routers

```
Router>enable
Router#configure terminal
Router(config)#access-list 10 permit host 192.168.4.2
Router(config)#line vty 0 4
Router(config-line)#access-class 10 in
```

Now we verify the remote access from PC using the following and find it to be successful



Now we verify the remote access from Server using the following and find it to be failure


Server0
Physical
Config
Services
Desktop
Programming
Attributes
Command Prompt
X
C:\>PING 192.168.4.2
Pinging 192.168.4.2 with 32 bytes of data:
Reply from 192.168.4.2: bytes=32 time=11ms TTL=125
Reply from 192.168.4.2: bytes=32 time=10ms TTL=125
Reply from 192.168.4.2: bytes=32 time=5ms TTL=125
Reply from 192.168.4.2: bytes=32 time=11ms TTL=125
Ping statistics for 192.168.4.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 11ms, Average = 9ms
C:\>ssh -l SSHadmin 192.168.1.1
% Connection refused by remote host
C:\>ssh -l SSHadmin 192.168.2.2
% Connection refused by remote host
C:\>ssh -l SSHadmin 192.168.3.1
% Connection refused by remote host
C:\>ssh -l SSHadmin 192.168.4.1
% Connection refused by remote host
C:\>

Part 3 - Create a Numbered IP ACL 120 on R1

We need to perform the following in this part

1) Create an IP ACL numbered 120 on R1 using the following rules 2) Permit any outside host to access DNS, SMTP, and FTP services on server

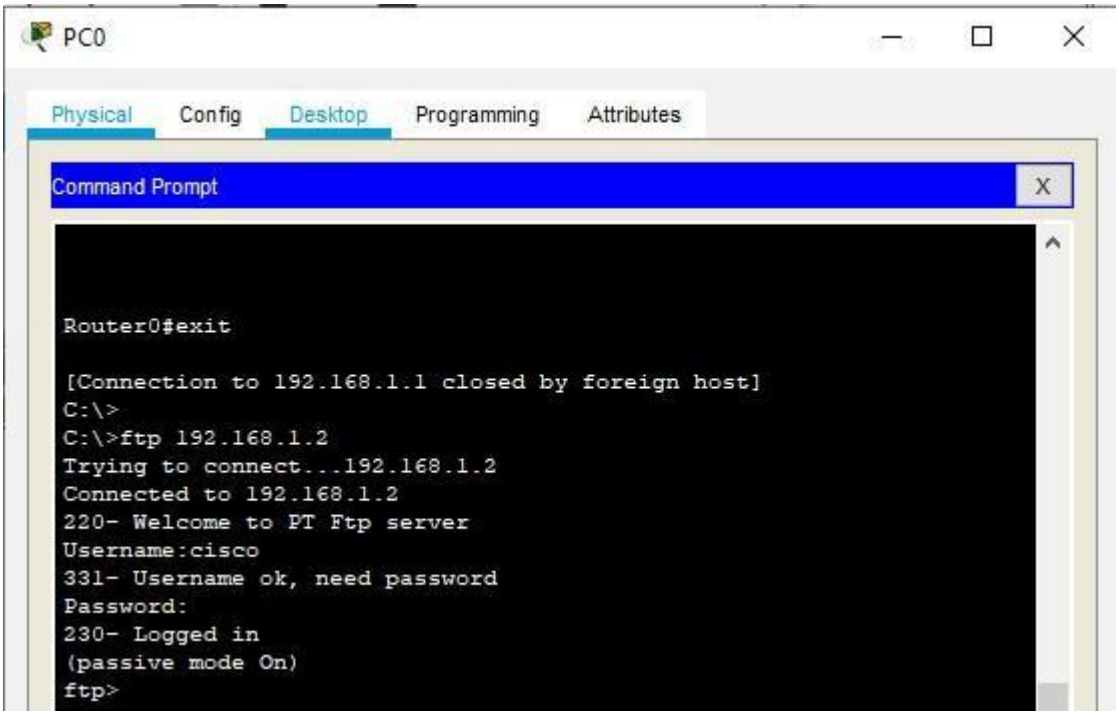
3) Deny any outside host access to HTTPS services on **server**

4) Permit PC to access Router1 via SSH. (Done in previous part)

Enter the following commands in the CLI mode of Router1

```
Router1>enable
Router1#
Router1#configure terminal
Router1(config)#access-list 120 permit udp any host 192.168.1.2 eq domain
Router1(config)#access-list 120 permit tcp any host 192.168.1.2 eq smtp
Router1(config)#access-list 120 permit tcp any host 192.168.1.2 eq ftp
Router1(config)#access-list 120 deny tcp any host 192.168.1.2 eq
443 Router1(config)#exit
Router1#configure terminal
Router1(config)#interface Serial0/1/1
Router1(config-if)#ip access-group 120 in
```

Verify the above entering the following commands in the PC



Hence, we have applied and verified all the required ACLs

Configuring IPv6 ACLs

Access Control Lists for IPv6 Traffic Filtering

The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4. Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list** command with the deny and **permit** keywords in global configuration mode.

IPv6 extended ACLs augments standard IPv6 ACL functionality to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

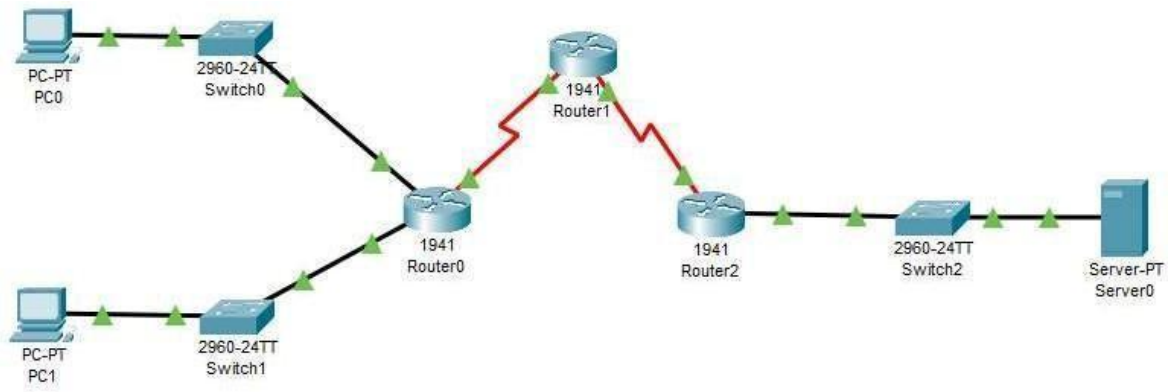
IPv6 Packet Inspection

The following header fields are used for IPv6 inspection: traffic class, flow label, payload length, next header, hop limit, and source or destination IP address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

Access Class Filtering in IPv6

Filtering incoming and outgoing connections to and from the device based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access- class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against the local device address on the interface. If the IPv6 ACL is applied to outbound traffic, the source address in the ACL is matched against the local device address on the interface and the destination address in the ACL is matched against the outgoing connection source address. We recommend that identical restrictions are set on all the virtual terminal lines because a user can attempt to connect to any of them.

We use the following topology



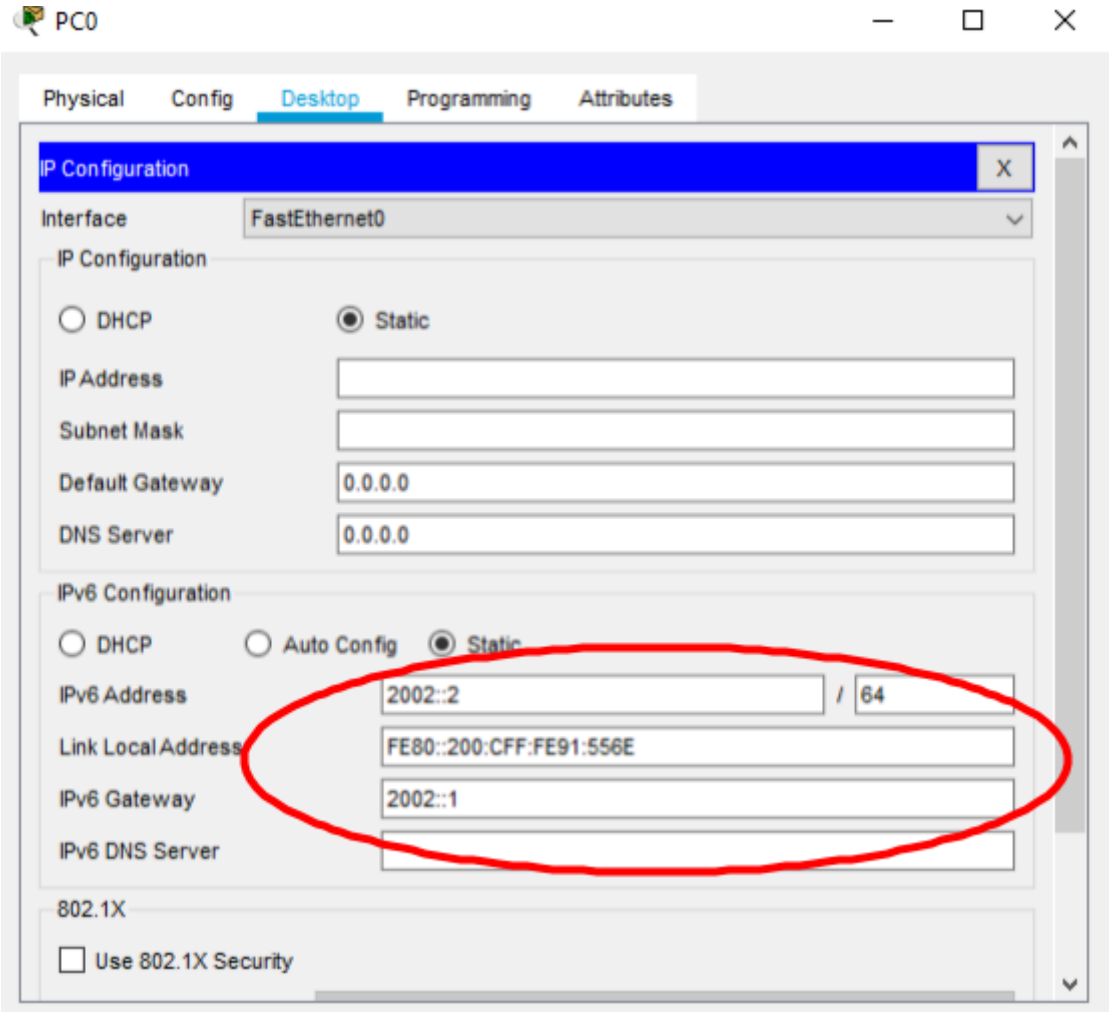
Let us consider the following Address table to configure the network devices:

Device	Interface	IPv6 Address	IPv6 gateway	Switch Port
PC 0	NA	2002::2 / 64	2002::1	Switch0 F0/1

PC 1	NA	2001::2 / 64	2001::1	Switch1 F0/1
Server0	NA	2005::2 / 64	2005::1	Switch2 F0/1
Router0	GE0/0	2002::1 / 64	NA	Switch0 F0/5
	GE0/1	2001::1 / 64	NA	Switch1 F0/5
	S0/1/0	2003::1 / 64	NA	NA
Router1	S0/1/0	2003::1 / 64	NA	NA
	S0/1/1	2004::1 / 64	NA	NA
Router2	S0/1/1	2004::2 / 64	NA	NA
	GE0/0	2005::1 / 64	NA	Switch2 F0/5

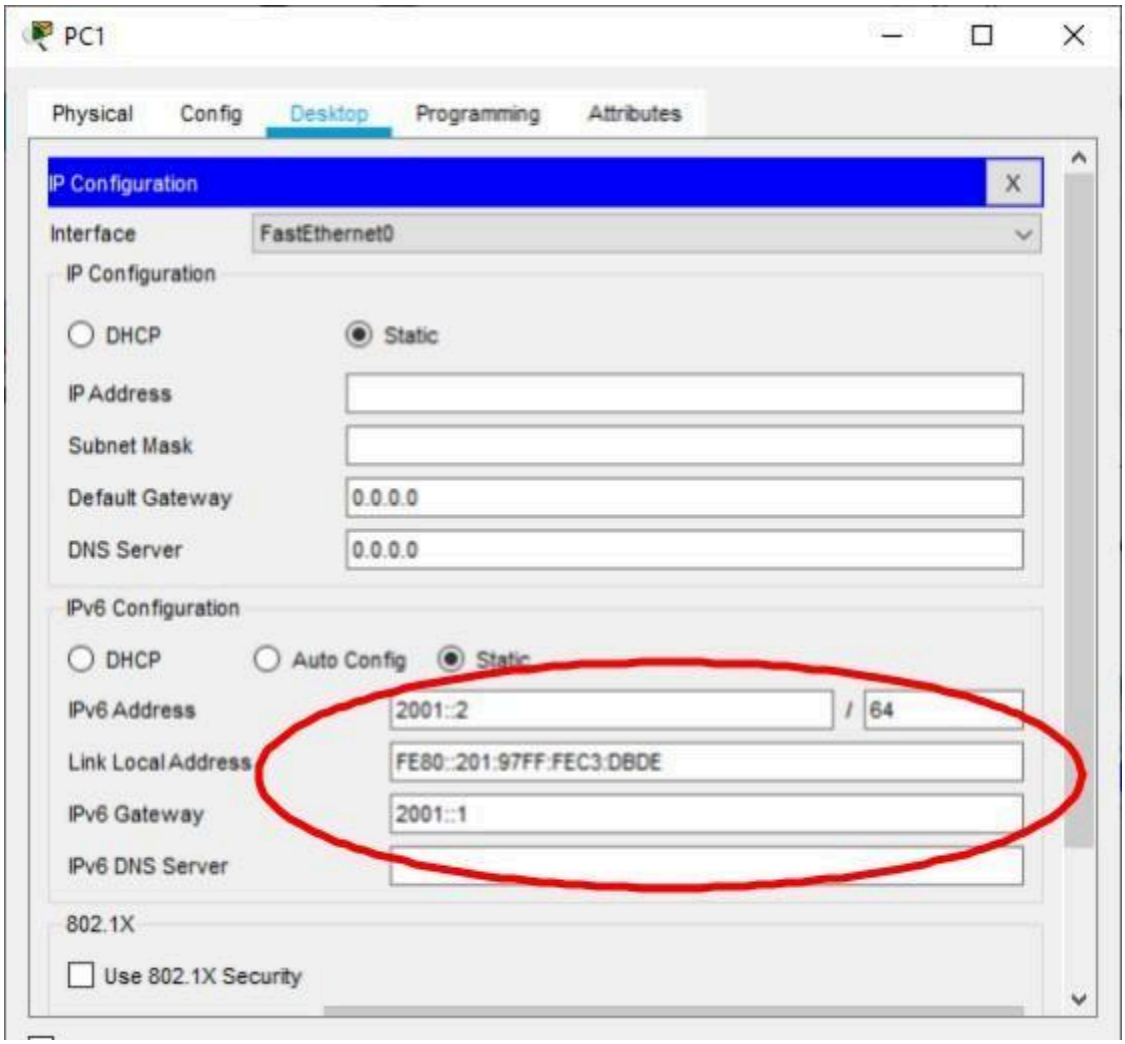
Configuring

PC0



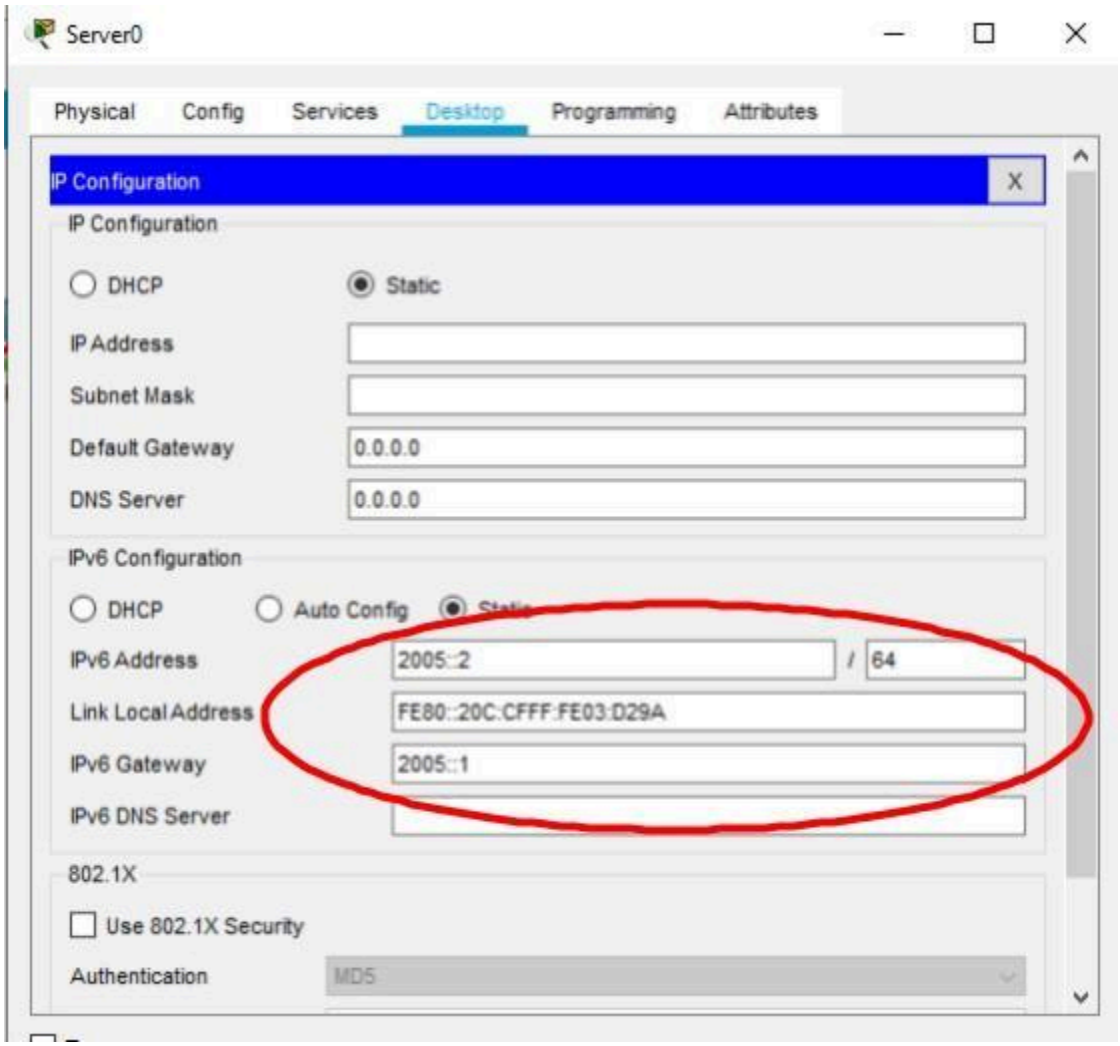
PC1

Configuring



Server0

Configuring



For setting the ipv6 addresses we need to use the CLI mode for each Router as follows

Configuring Router0

```
Router>
Router>enable
Router#
Router#configure terminal
Router(config)#ipv6 unicast-routing

Router(config)#interface
GigabitEthernet0/0 Router(config-if)#ipv6
address 2002::1/64 Router(config-if)#ipv6 rip a
enable Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#

Router(config)#interface
GigabitEthernet0/1 Router(config-if)#ipv6
address 2001::1/64 Router(config-if)#ipv6 rip a
enable Router(config-if)#no shutdown
Router(config-if)#exit
 Router(config)#

Router(config)#interface Serial0/1/0
Router(config-if)#ipv6 address 2003::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Configuring Router1

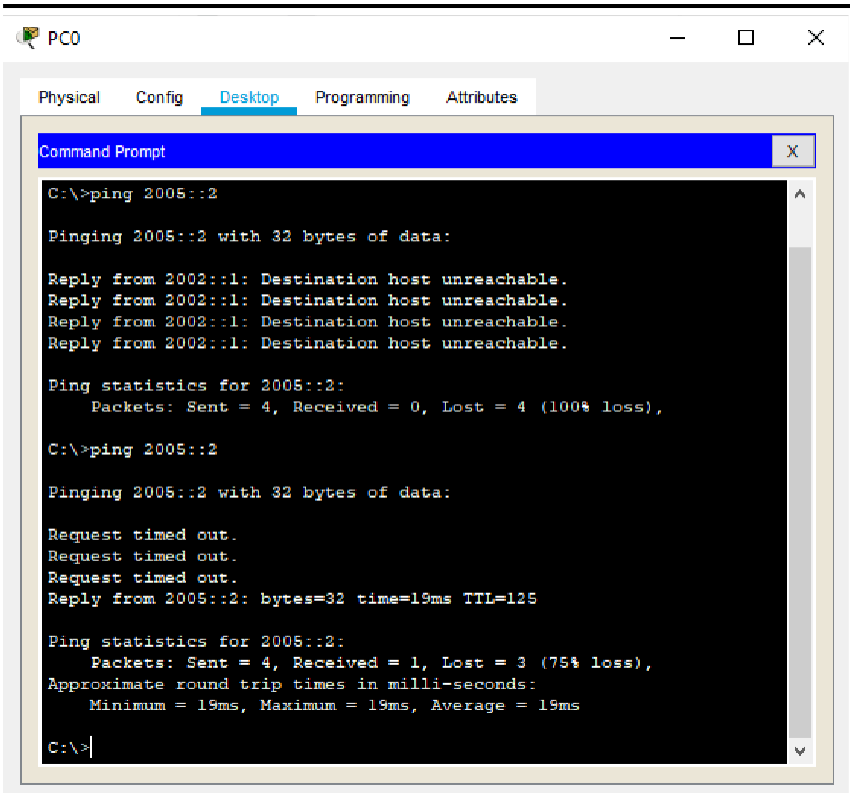
```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing
Router(config)#
```

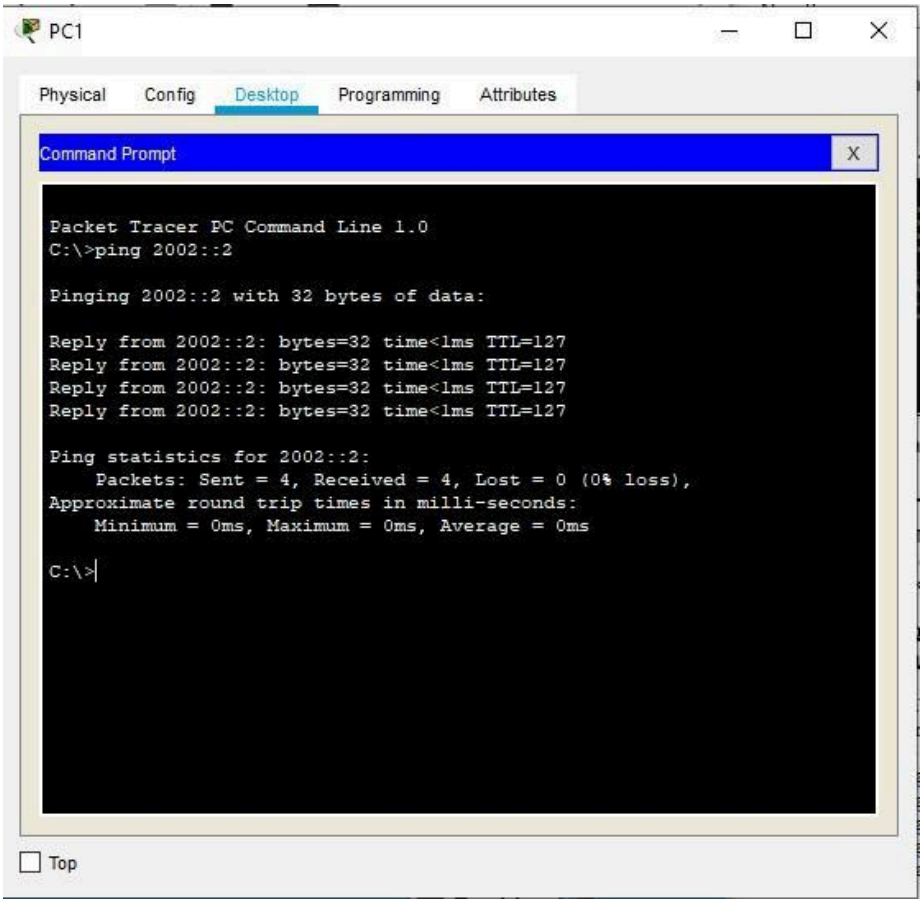
```
Router(config)#interface Serial0/1/0
Router(config-if)#ipv6 address 2003::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#
Router(config-if)#exit
Router(config)#

Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 address 2004::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Configuring Router2

```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing
Router(config)#

Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 address 2004::2/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown Router(config-if)#exit

Router(config)#interface GigabitEthernet0/0
Router(config-if)#ipv6 address 2005::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Check the connectivity by pinging from PCs to Server


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
C:\>ping 2005::2
Pinging 2005::2 with 32 bytes of data:
Reply from 2002::1: Destination host unreachable.
Reply from 2002::1: Destination host unreachable.
Reply from 2002::1: Destination host unreachable.
Reply from 2002::1: Destination host unreachable.
Ping statistics for 2005::2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 2005::2
Pinging 2005::2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Reply from 2005::2: bytes=32 time=19ms TTL=125
Ping statistics for 2005::2:
Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
Approximate round trip times in milli-seconds:
Minimum = 19ms, Maximum = 19ms, Average = 19ms
C:\>



And we see that the connectivity is established

We configure the ACL and apply it to the Router1 with the following conditions

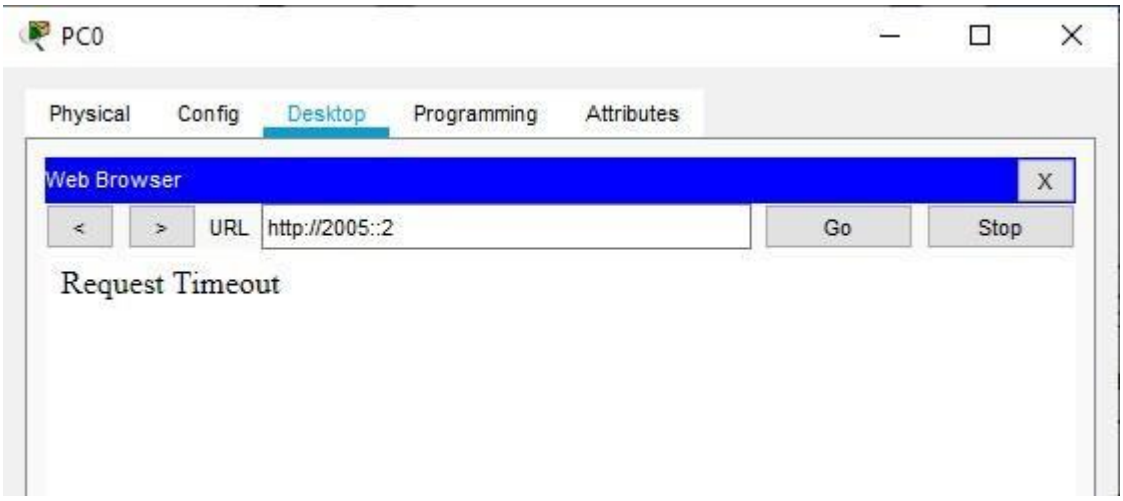
1) No HTTP or HTTPS allowed on server by any host
2) No www service accessible on the server by any host
3) Only ipv6 packets allowed towards the server

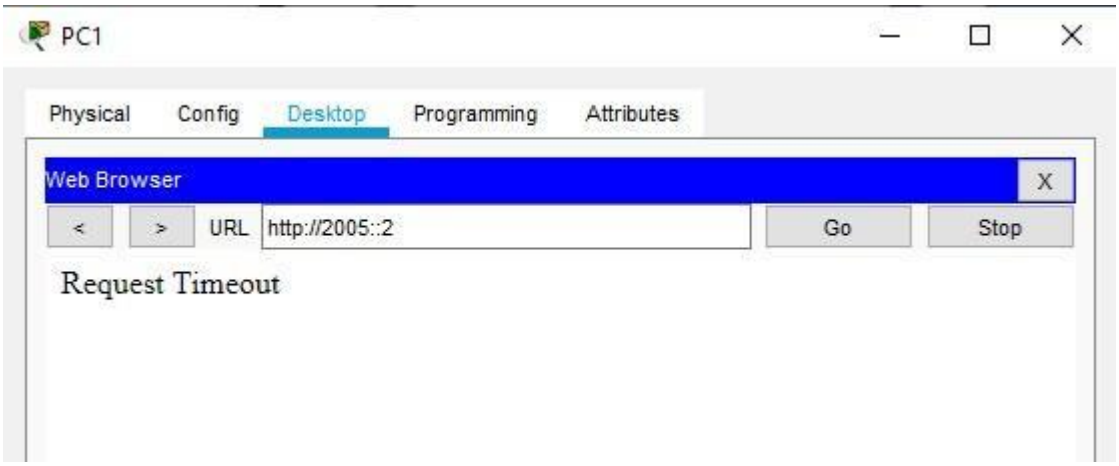
We enter the following commands in the CLI mode of the Router1 and Router2, apply it at the proper interface

Router>
Router>enable
Router#configure terminal
Router(config)#ipv6 access-list smile
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq www
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq 443
Router(config-ipv6-acl)#permit ipv6 any any
Router(config-ipv6-acl)# Router(config-ipv6-acl)#exit

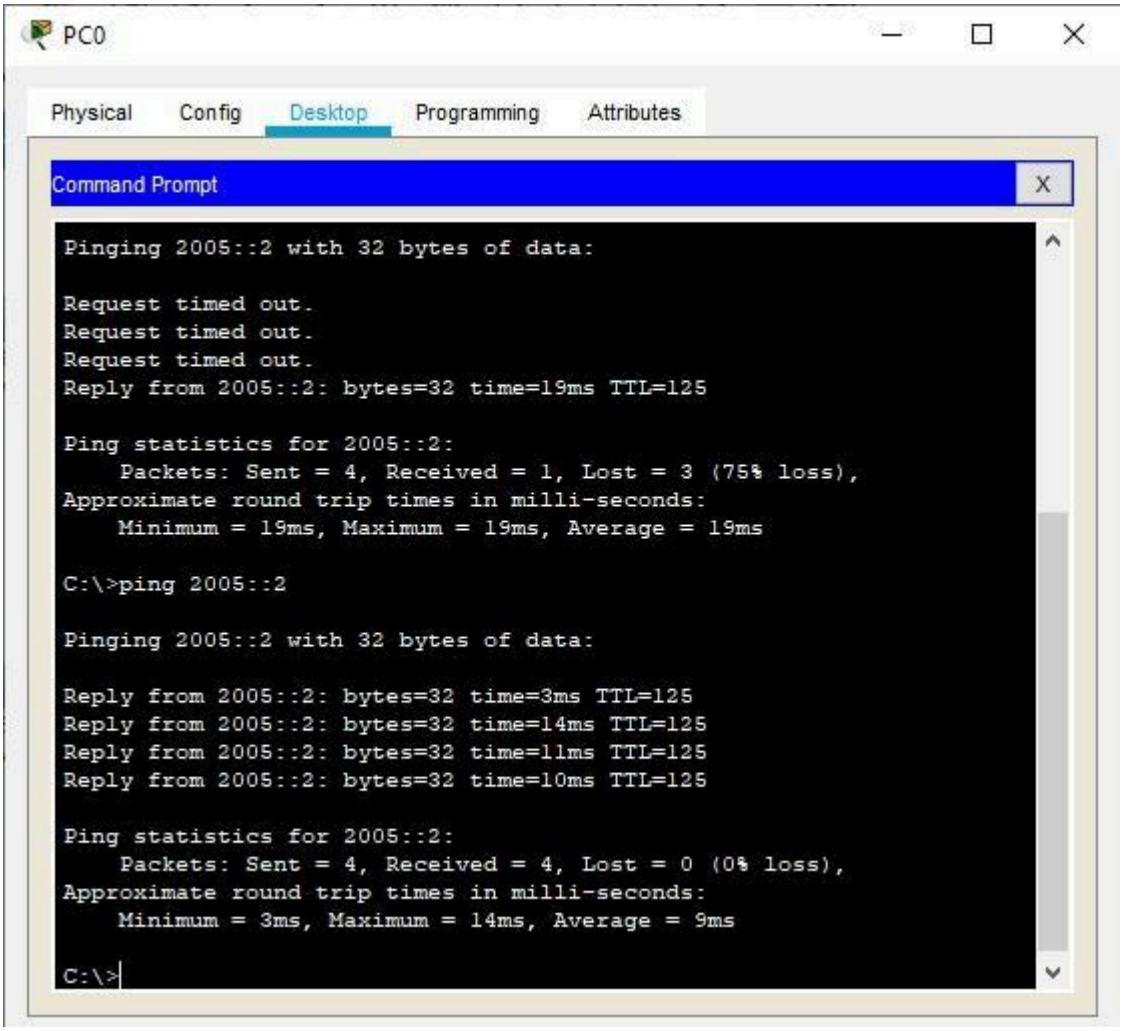
```
Router(config)# Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 traffic-filter smile in
Router(config-if)#exit
Router(config)#
```

We verify the configuration by first accessing the www service from the browser of both PCs and get failure





Next we verify whether the ipv6 protocol works by pinging server from any of the PC (it must be successful)



Hence the given ACLs have been applied and verified on host running on ipv6 protocol.

PRACTICAL NO 5: Configuring a Zone-Based Policy Firewall (ZPF)

Cisco IOS® Software Release 12.4(6)T introduced Zone-Based Policy Firewall (ZFW), a new configuration model for the Cisco IOS Firewall feature set. This new configuration model offers intuitive policies for multiple-interface routers, increased granularity of firewall policy application, and a default deny-all policy that prohibits traffic between firewall security zones until an explicit policy is applied to allow desirable traffic.
Nearly all classic Cisco IOS Firewall features implemented before Cisco IOS Software Release 12.4(6)T are supported in the new zone-based policy inspection interface:

1) Stateful packet inspection
2) VRF-aware Cisco IOS Firewall
3) URL filtering
4) Denial-of-Service (DoS) mitigation

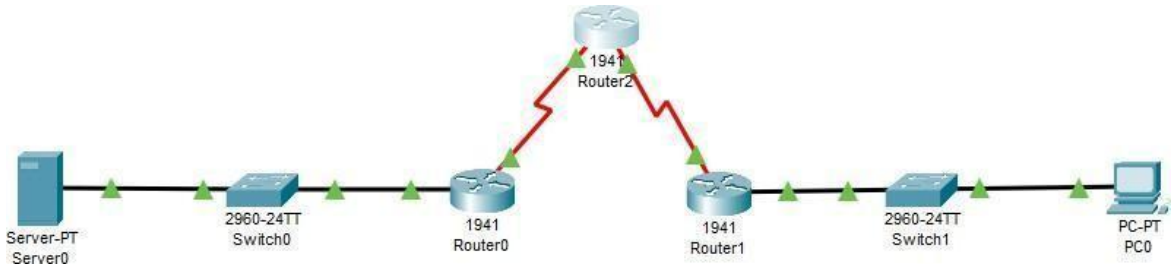
Cisco IOS Software Release 12.4(9)T added ZFW support for per-class session/connection and throughput limits, as well as application inspection and control:

1) HTTP
2) Post Office Protocol (POP3),
3) Internet Mail Access Protocol (IMAP),
4) Simple Mail Transfer Protocol / Enhanced Simple Mail Transfer Protocol (SMTP/ESMTP)
5) Sun Remote Procedure Call (RPC)
6) Instant Messaging (IM) applications:
   i) Microsoft Messenger ii) Yahoo! Messenger iii) AOL Instant Messenger
7) Peer-to-Peer (P2P) File Sharing:
   i) Bittorrent ii) KaZaA iii) Gnutella iv) eDonkey

Cisco IOS Software Release 12.4(11)T added statistics for easier DoS protection tuning.
Some Cisco IOS Classic Firewall features and capabilities are not yet supported in a ZFW in Cisco IOS Software Release 12.4(15)T: i) Authentication proxy ii) Stateful firewall failover iii) Unified firewall MIB iv) IPv6 stateful inspection
v) TCP out-of-order support

ZFW generally improves Cisco IOS performance for most firewall inspection activities. Neither Cisco IOS ZFW or Classic Firewall include stateful inspection support for multicast traffic.

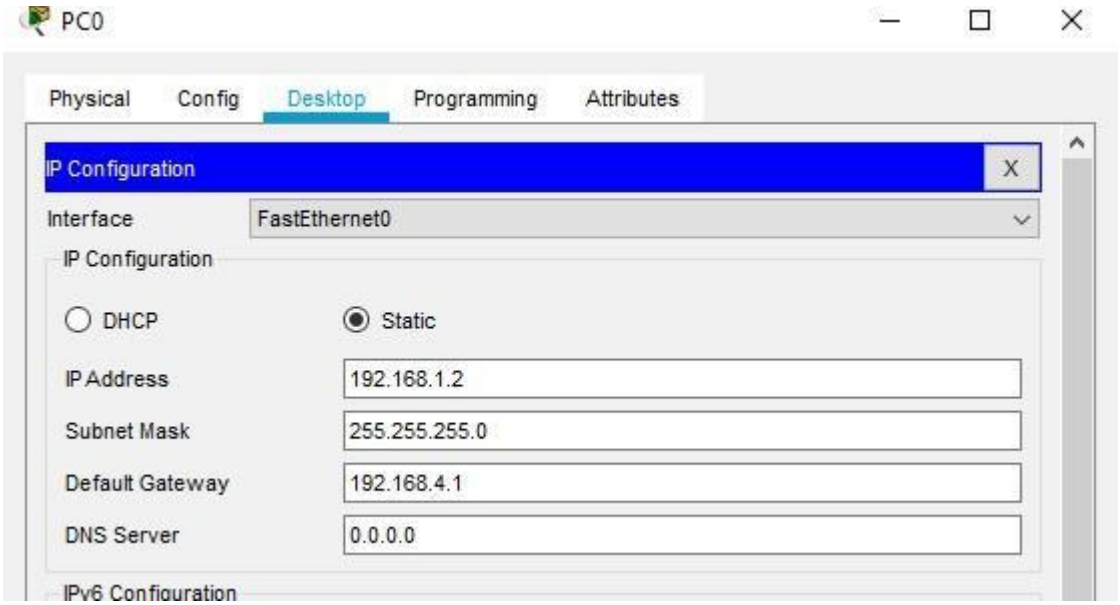
We use the following topology



Let us consider the following Address table to configure the network devices:

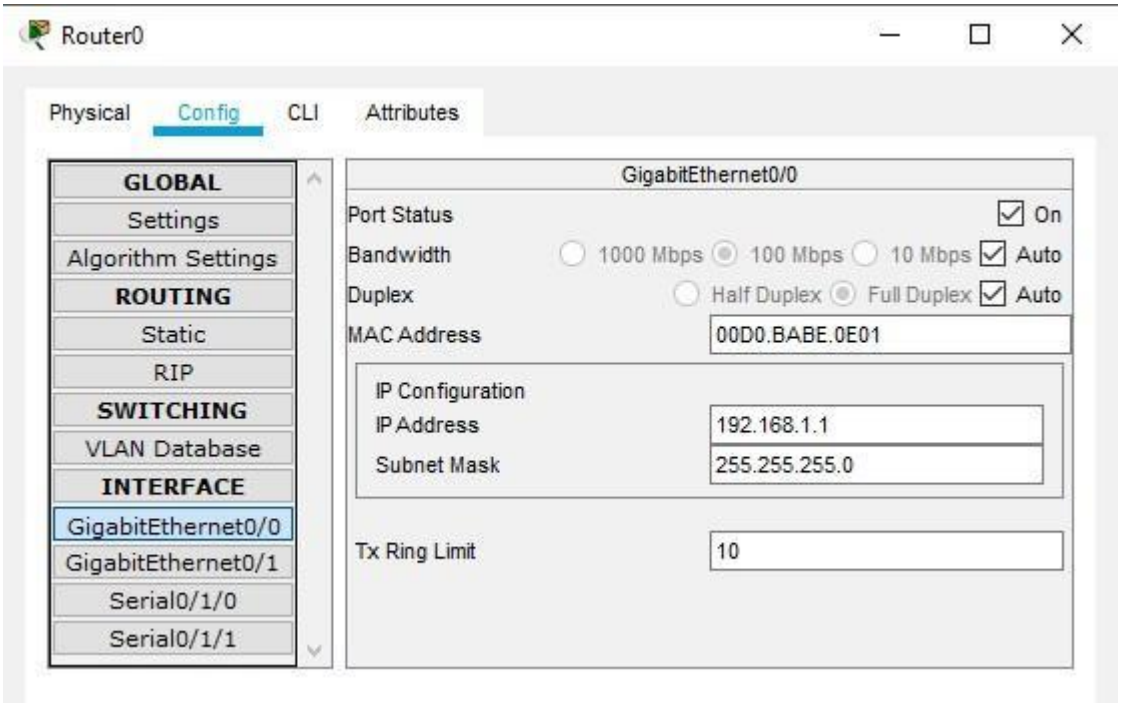
Device	Interface	IP Address	Subnet Mask	Default gatewa y	Switch Port
PC 0	NA	192.168.4.2	255.255.255.0	192.168.4.1	Switch1 F0/1
Server0	NA	192.168.1.2	255.255.255.0	192.168.1.1	Switch0 F0/1
Router0	GE0/0	192.168.1.1	255.255.255.0	NA	Switch0 F0/5
	S0/1/0	192.168.2.1	255.255.255.0	NA	NA
Router2	S0/1/0	192.168.2.2	255.255.255.0	NA	NA
	S0/1/1	192.168.3.1	255.255.255.0	NA	NA
Router1	S0/1/1	192.168.3.2	255.255.255.0	NA	NA
	GE0/0	192.168.4.1	255.255.255.0	NA	Switch1 F0/5

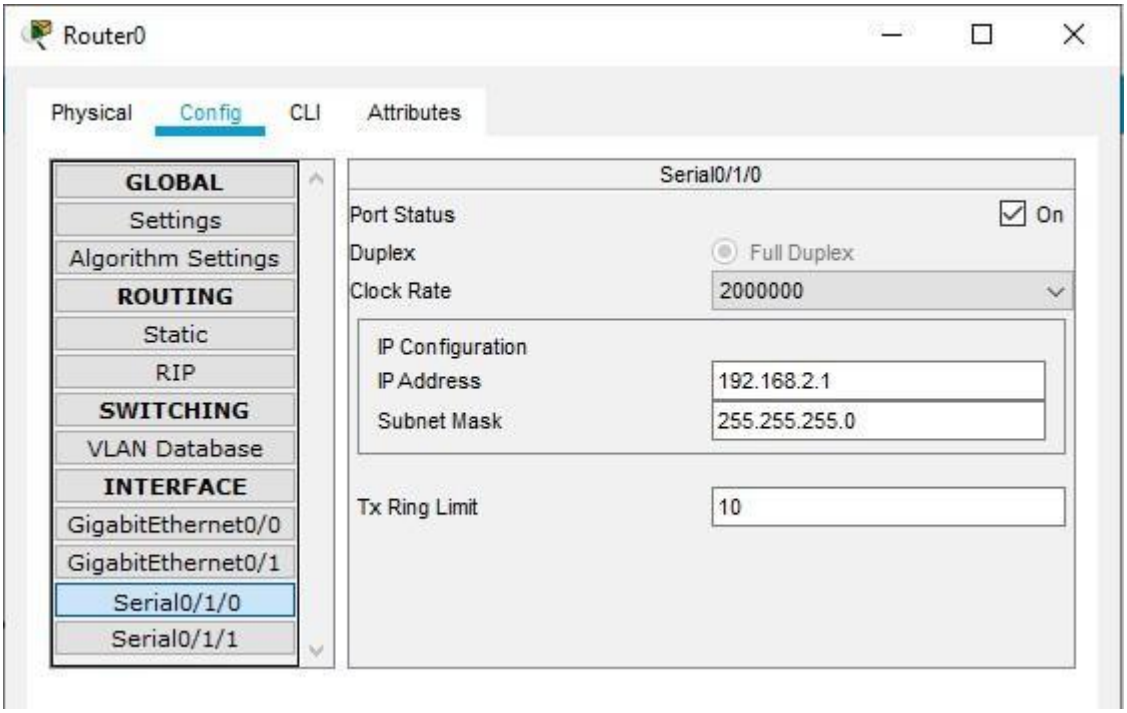
Configuring PC0



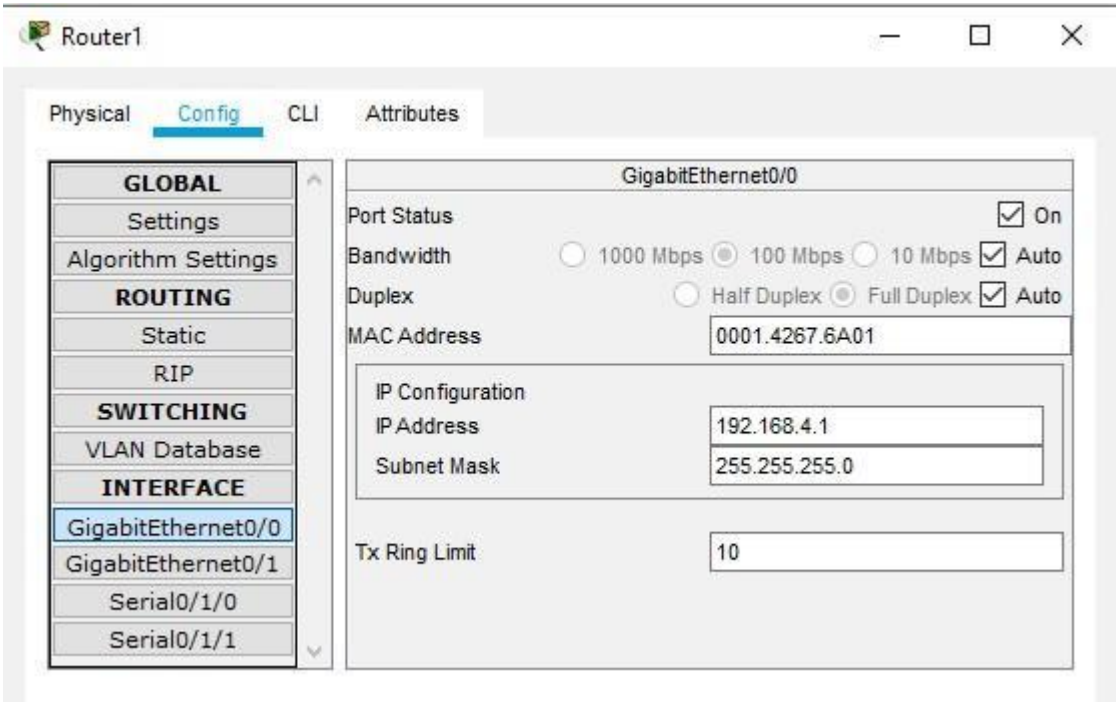
Configuring Server0

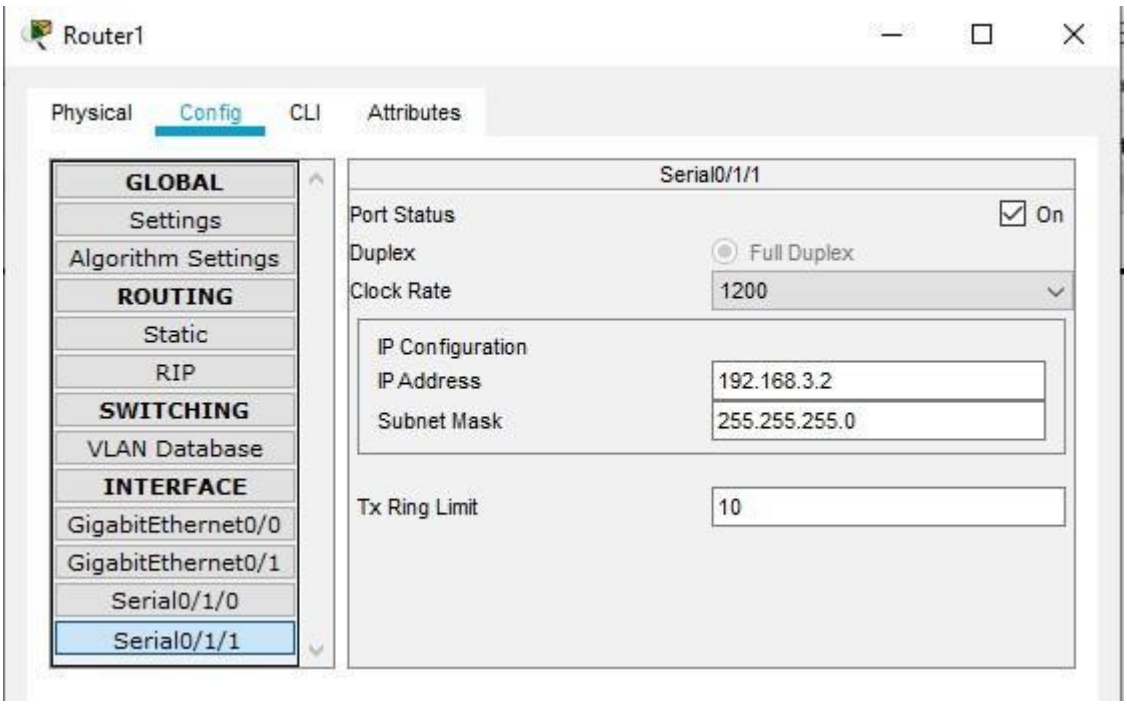
Configuring Router0



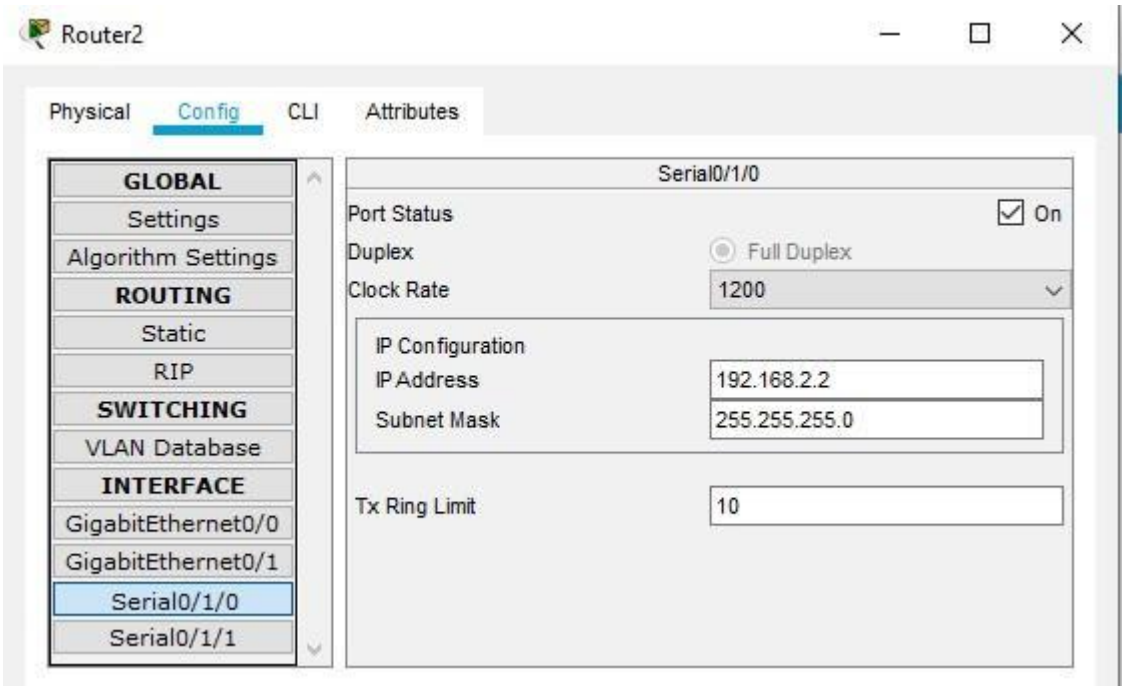


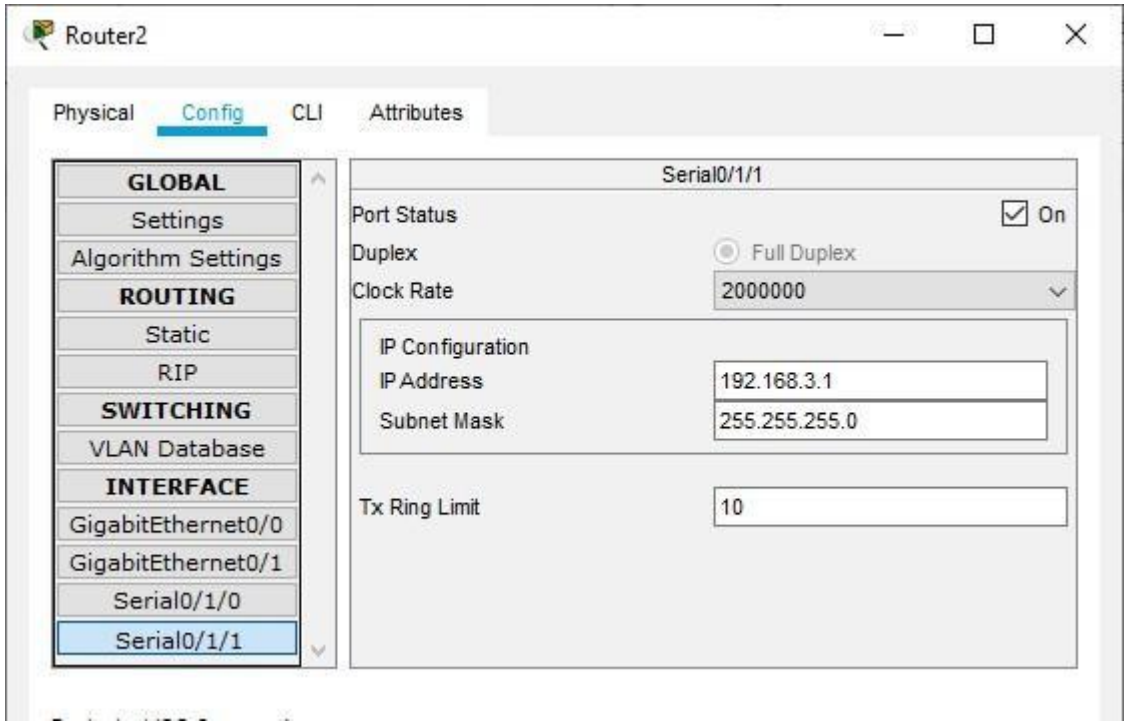
Configuring Router1


Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
GigabitEthernet0/0
Port Status
On
Bandwidth
1000 Mbps
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0001.4267.6A01
IP Configuration
IP Address
192.168.4.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10


Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/1
Port Status
On
Duplex
Full Duplex
Clock Rate
1200
IP Configuration
IP Address
192.168.3.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10

Configuring Router2

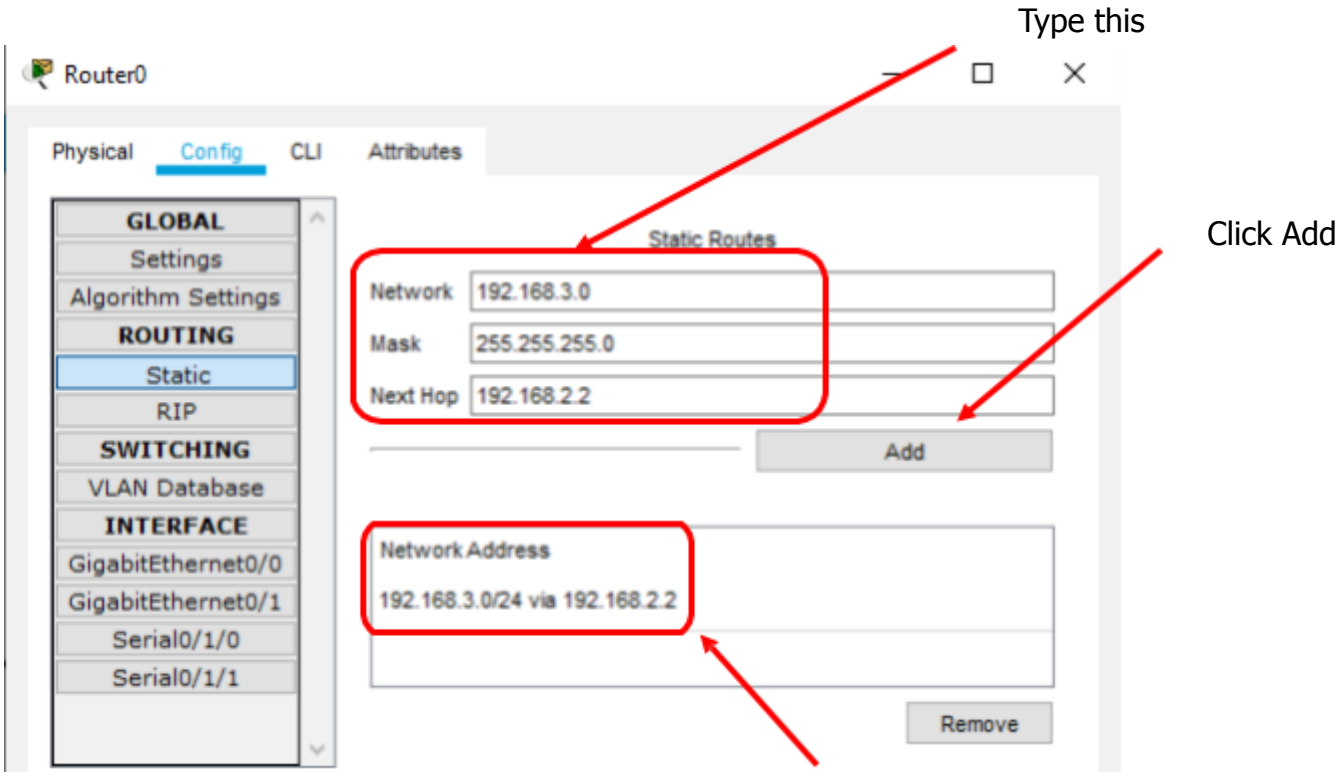


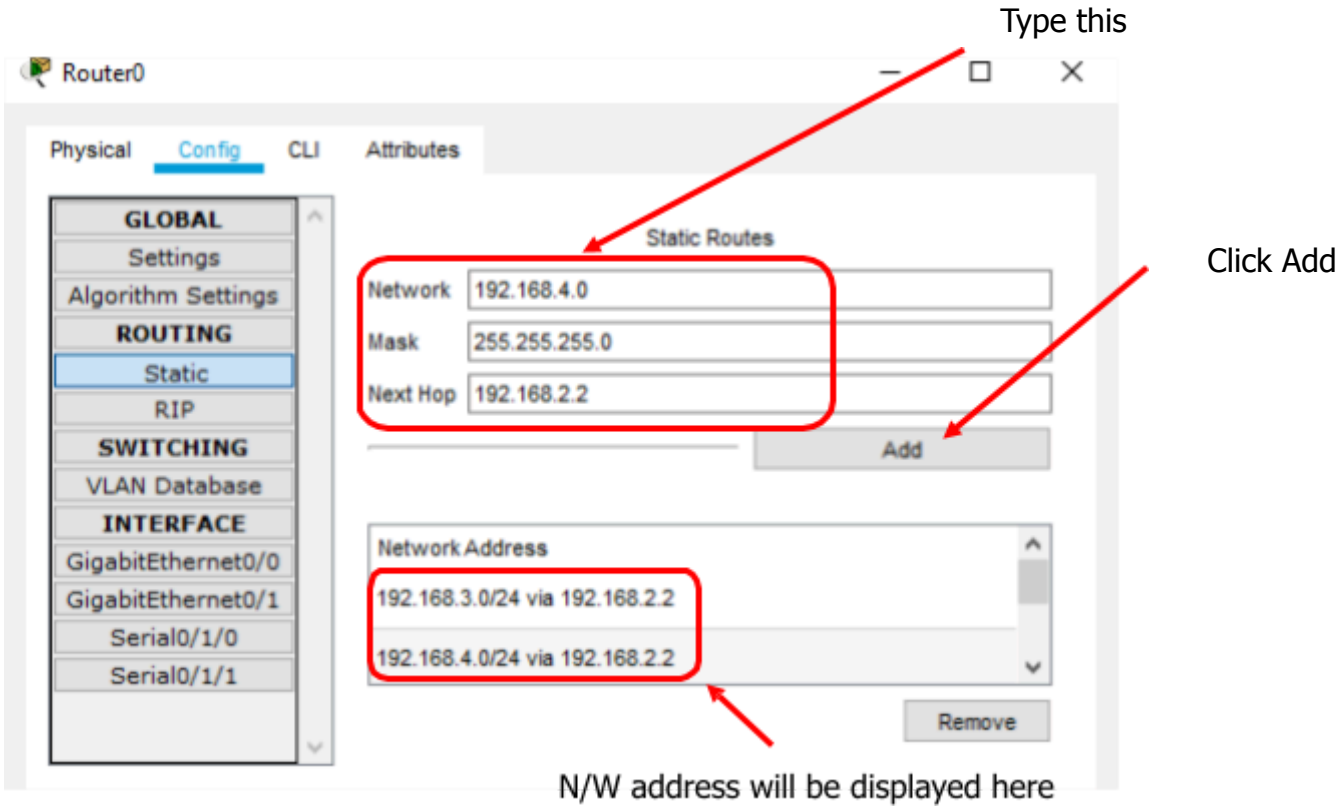

Router2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/1
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IP Address
192.168.3.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10

Part 1: Static Routing

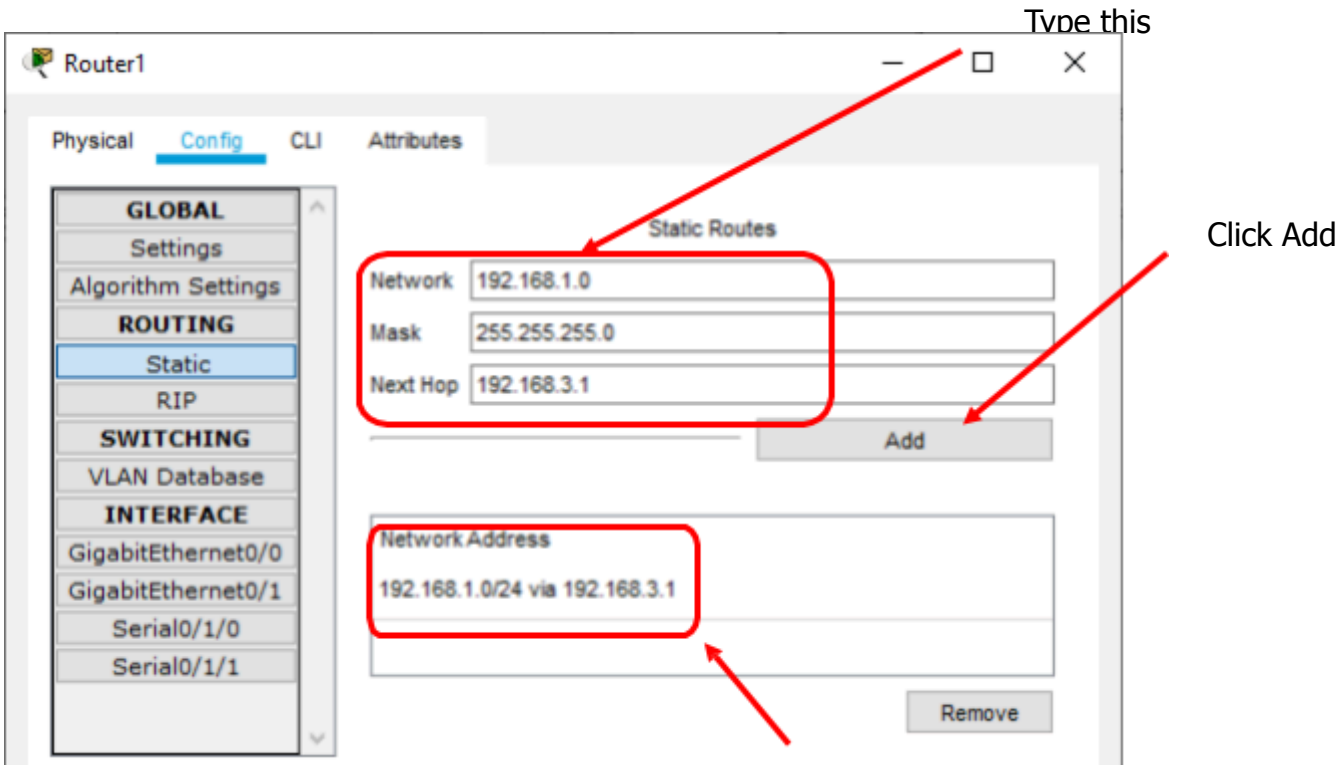
Static Routing is done using the following procedure for each Router

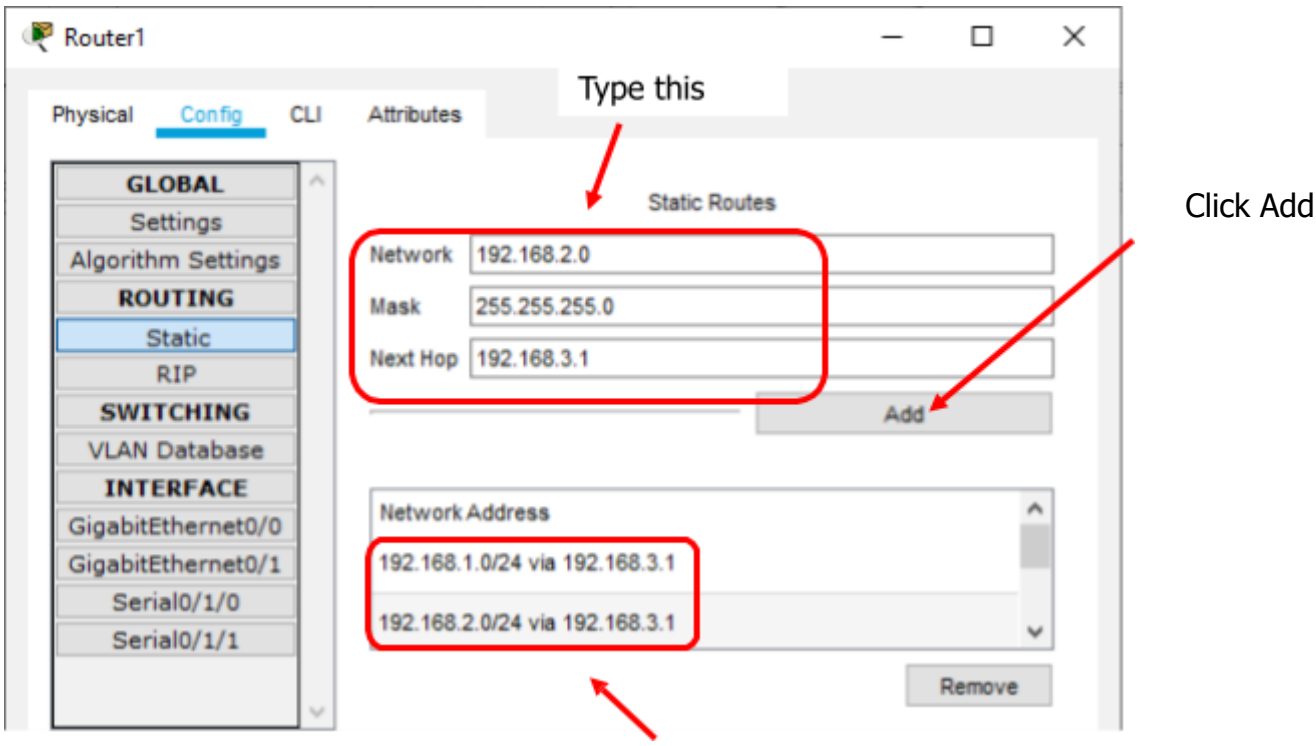
Router 0: Add the following in the Static mode of Router0





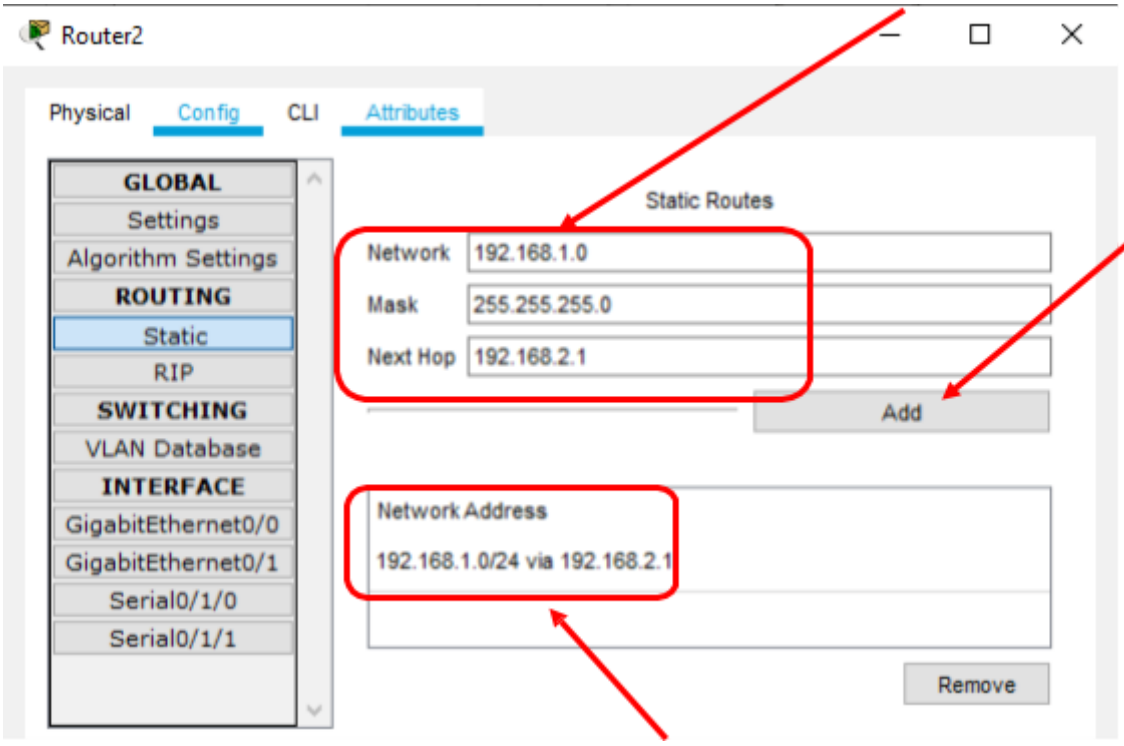
Router 1: Add the following in the Static mode of Router1





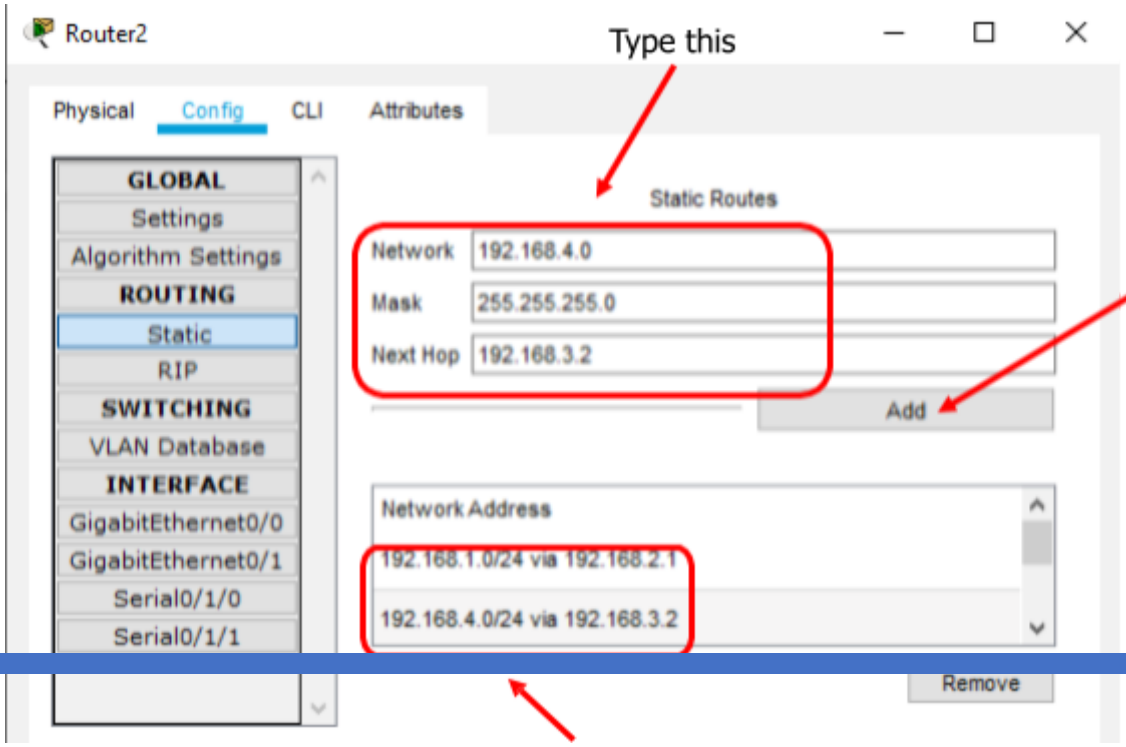
N/W address will be displayed here

Router 2: Add the following in the Static mode of Router2 Type this



Click Add

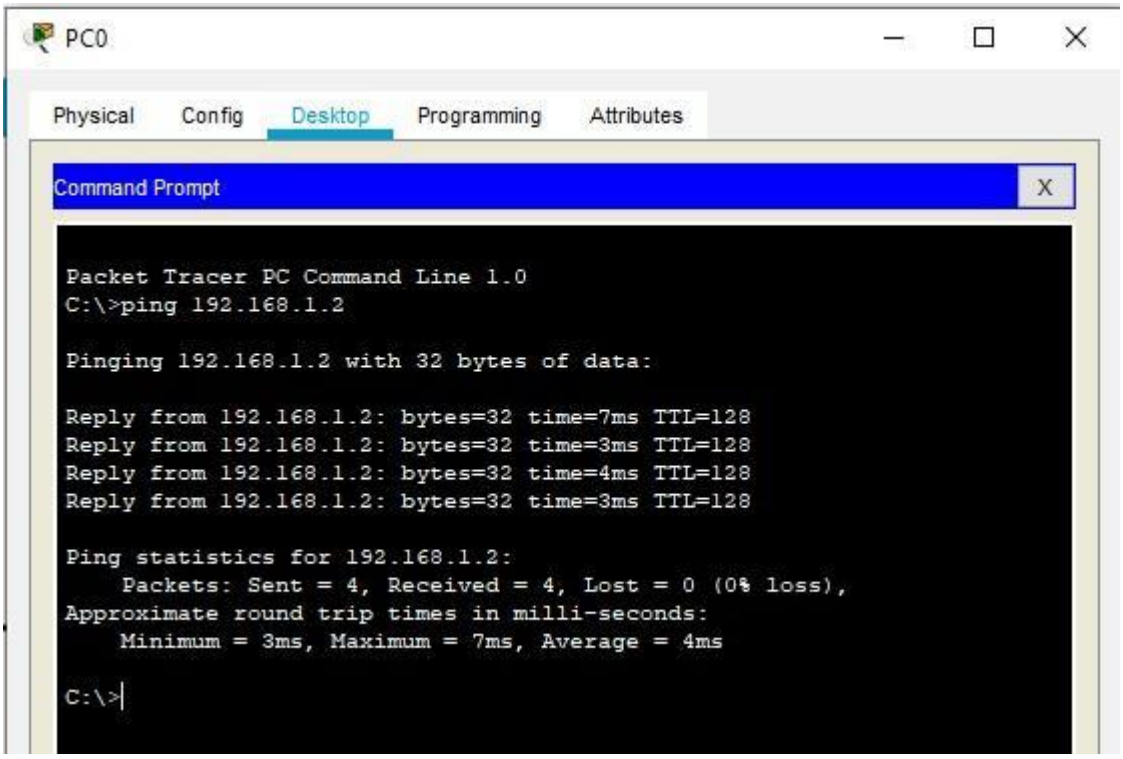
N/W address will be displayed here

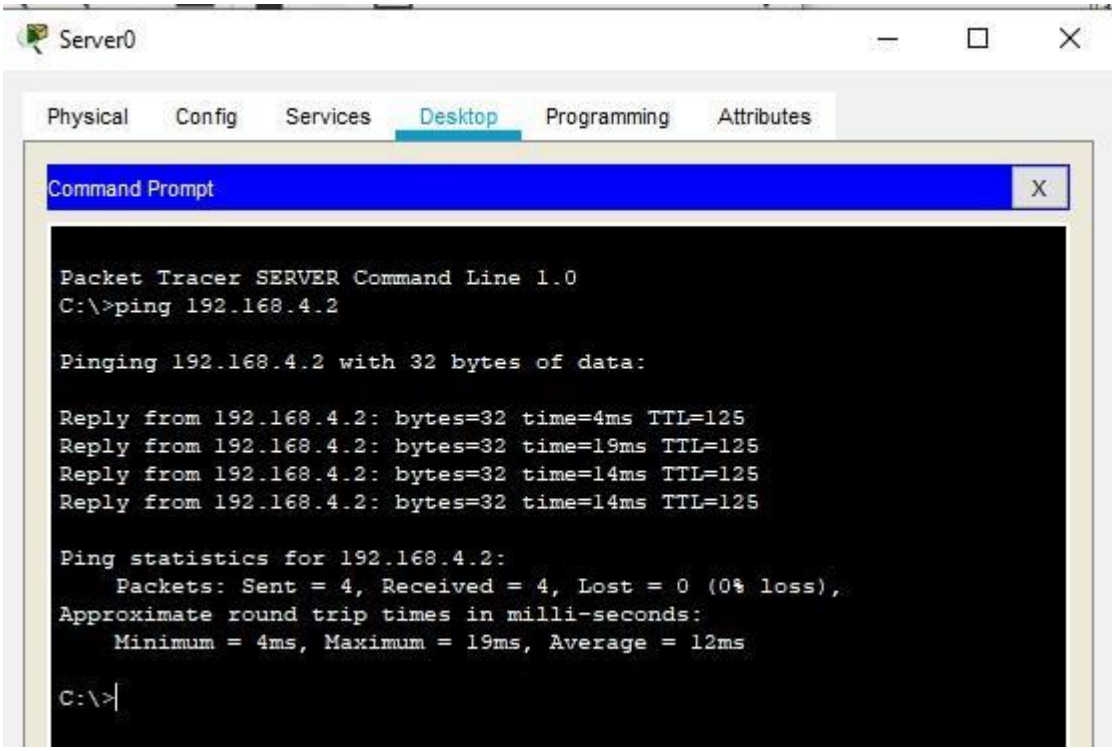


Click Add

N/W address will be displayed here

Now we check the connectivity by pinging the Server from the PC and from PC to Server





Part 2: Configuring SSH on Router 2

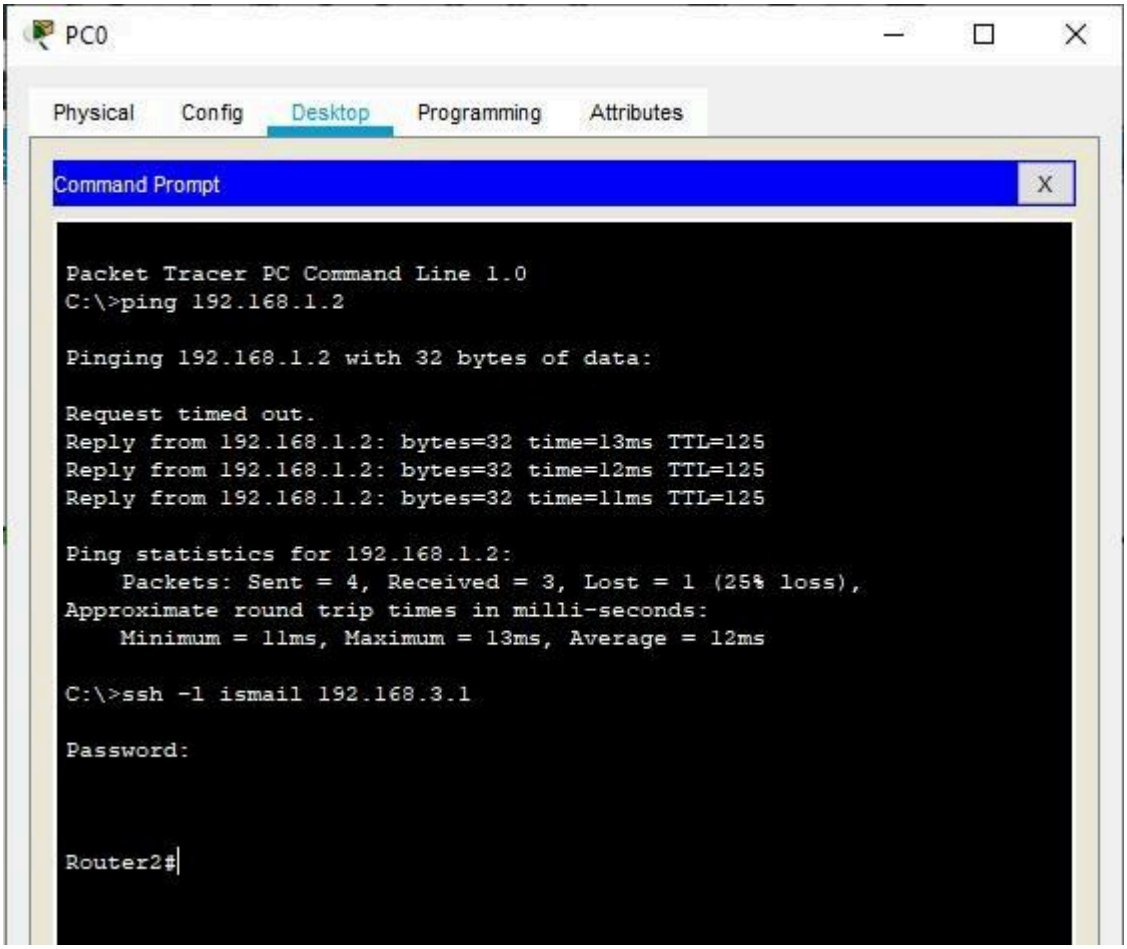
Type the following commands in the CLI mode of Router2

Router>enable
Router#configure terminal
Router(config)#ip domain-name .com
Router(config)#hostname Router2
Router2(config)#crypto key generate rsa

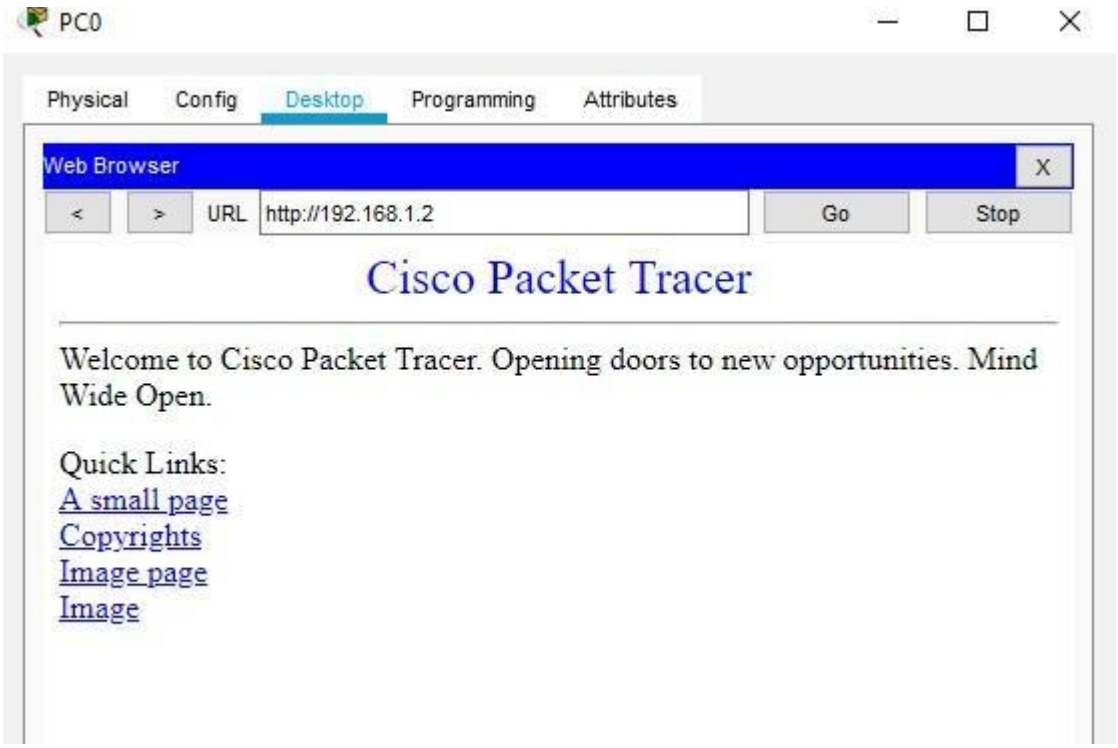
Router2 (config)#line vty 0 4
Router2 (config-line)#transport input ssh
Router2 (config-line)#login local
Router2 (config-line)#exit

Router2 (config)#username ismail privilege 15 password cisco

Now verify ssh from PC0 by typing the following command ssh -l ismail 192.168.3.1


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.2: bytes=32 time=13ms TTL=125
Reply from 192.168.1.2: bytes=32 time=12ms TTL=125
Reply from 192.168.1.2: bytes=32 time=11ms TTL=125
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 11ms, Maximum = 13ms, Average = 12ms
C:\>ssh -l ismail 192.168.3.1
Password:
Router2#

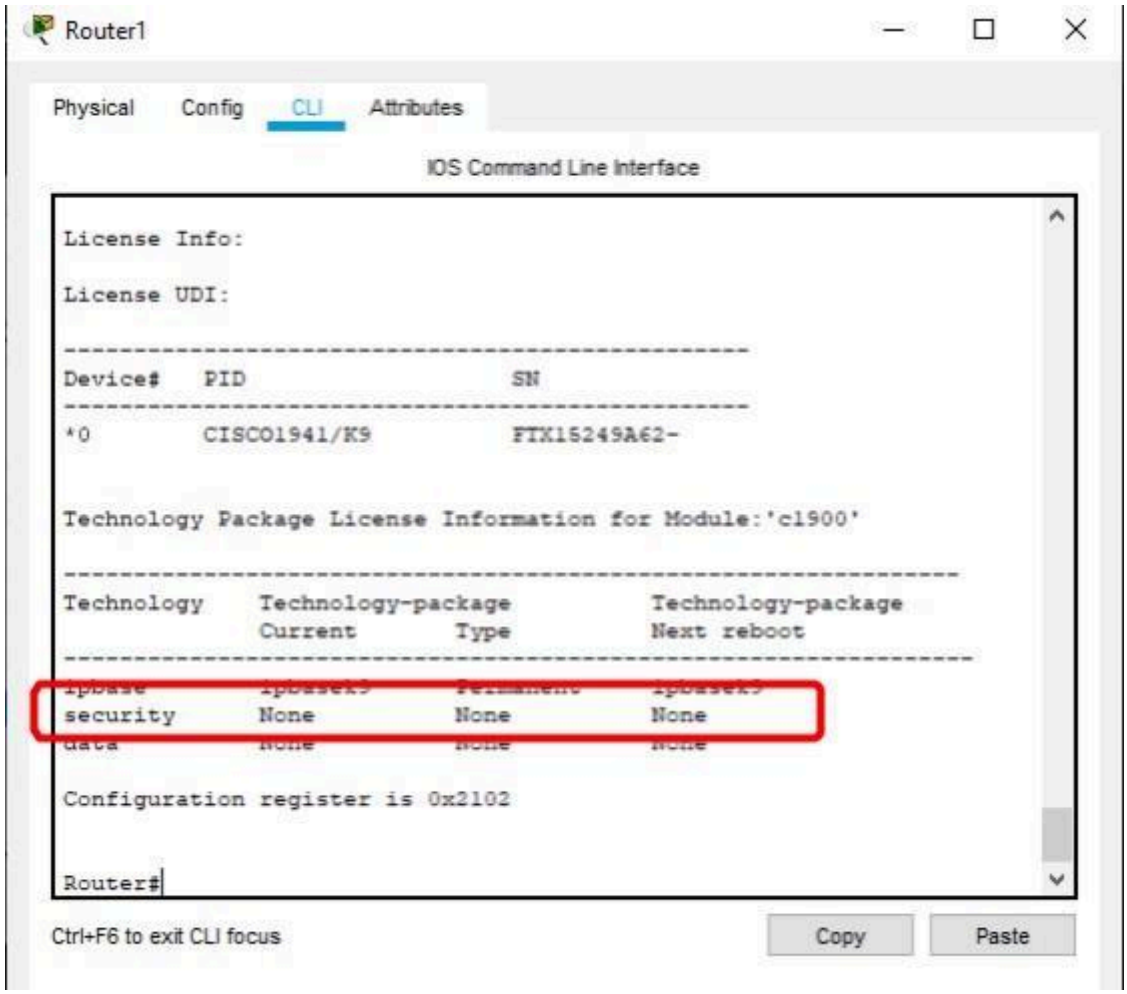
Next we access the web services of the Server using the web browser of PC using the following



Part 3: Create the Firewall Zones on Router1

Type the following commands in the CLI mode of Router1

Router>enable
Router#configure terminal
Router(config)#show version



Router#configure terminal
Router (config)#license boot module c1900 technology-package securityk9 ACCEPT? [yes/no]: y

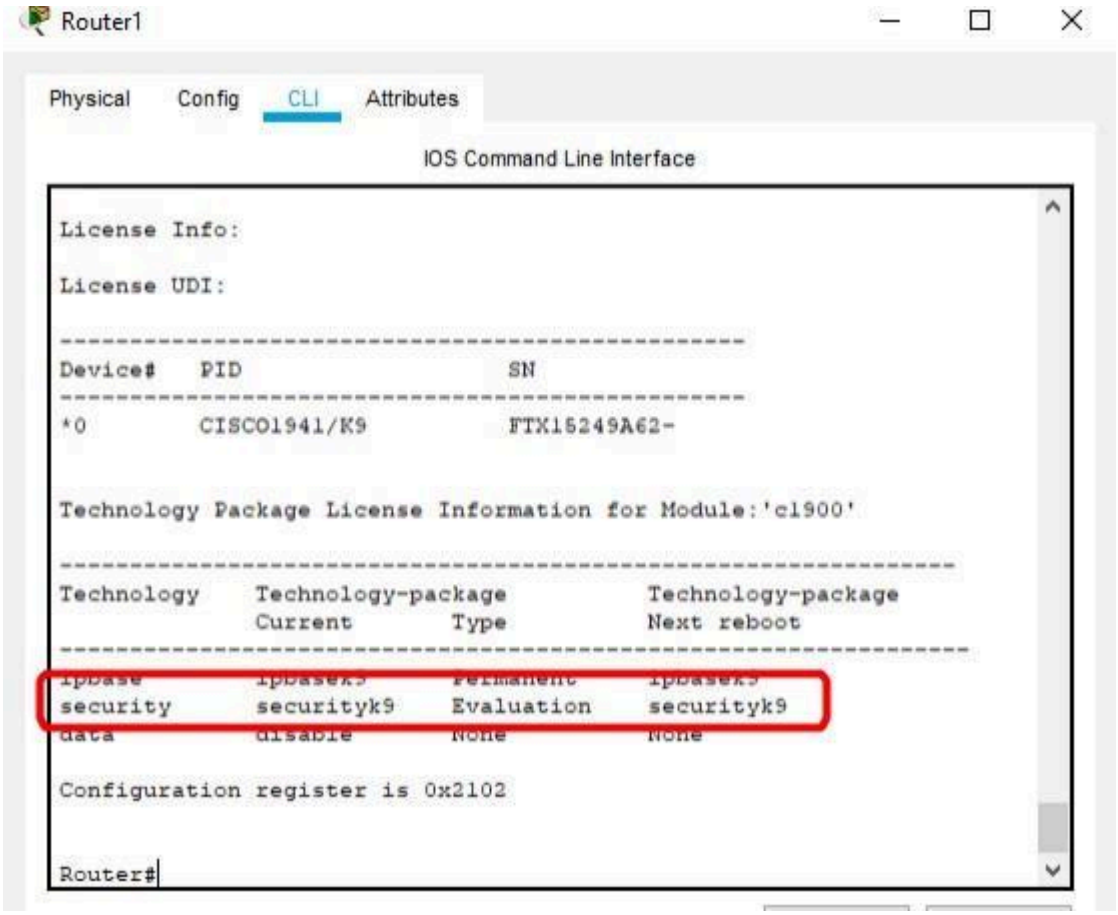
Router(config)#exit
Router>enable

Router#reload

Router>enable

THAKUR RAMNARAYAN COLLEGE OF ARTS AND COMMERCE

Router#show version



Router# Router#configure terminal

Router(config)#zone security in-zone
Router(config-sec-zone)#exit

Router(config)#zone security out-zone Router(config-sec-zone)#exit

Router(config)#access-list 101 permit ip 192.168.4.0
0.0.0.255 any Router(config)#class- map type inspect

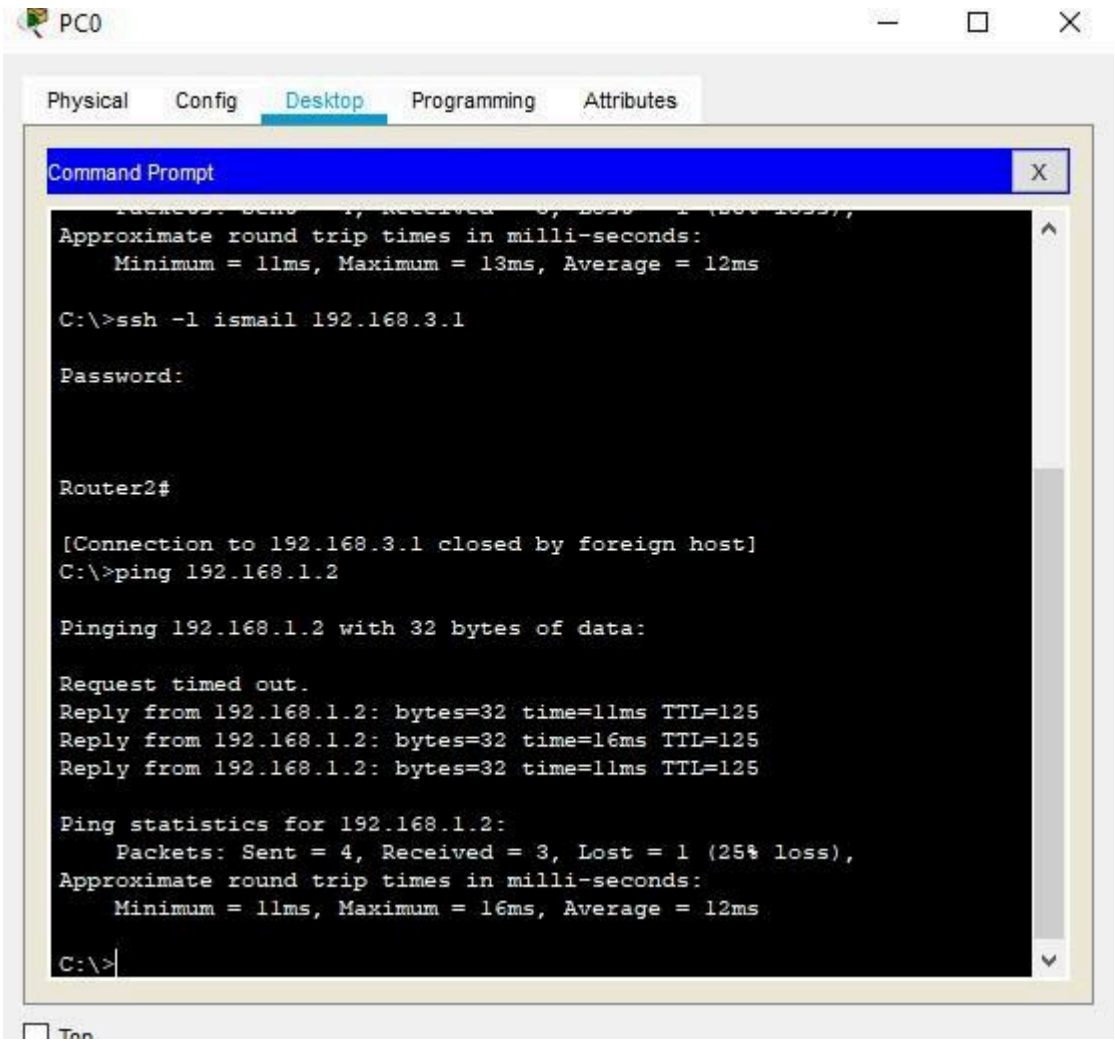
match-all in-map Router(config- cmap)#match
access-group 101 Router(config-cmap)#exit

Router(config)#policy-map type inspect
in-out Router(config- pmap)#class type
inspect in-

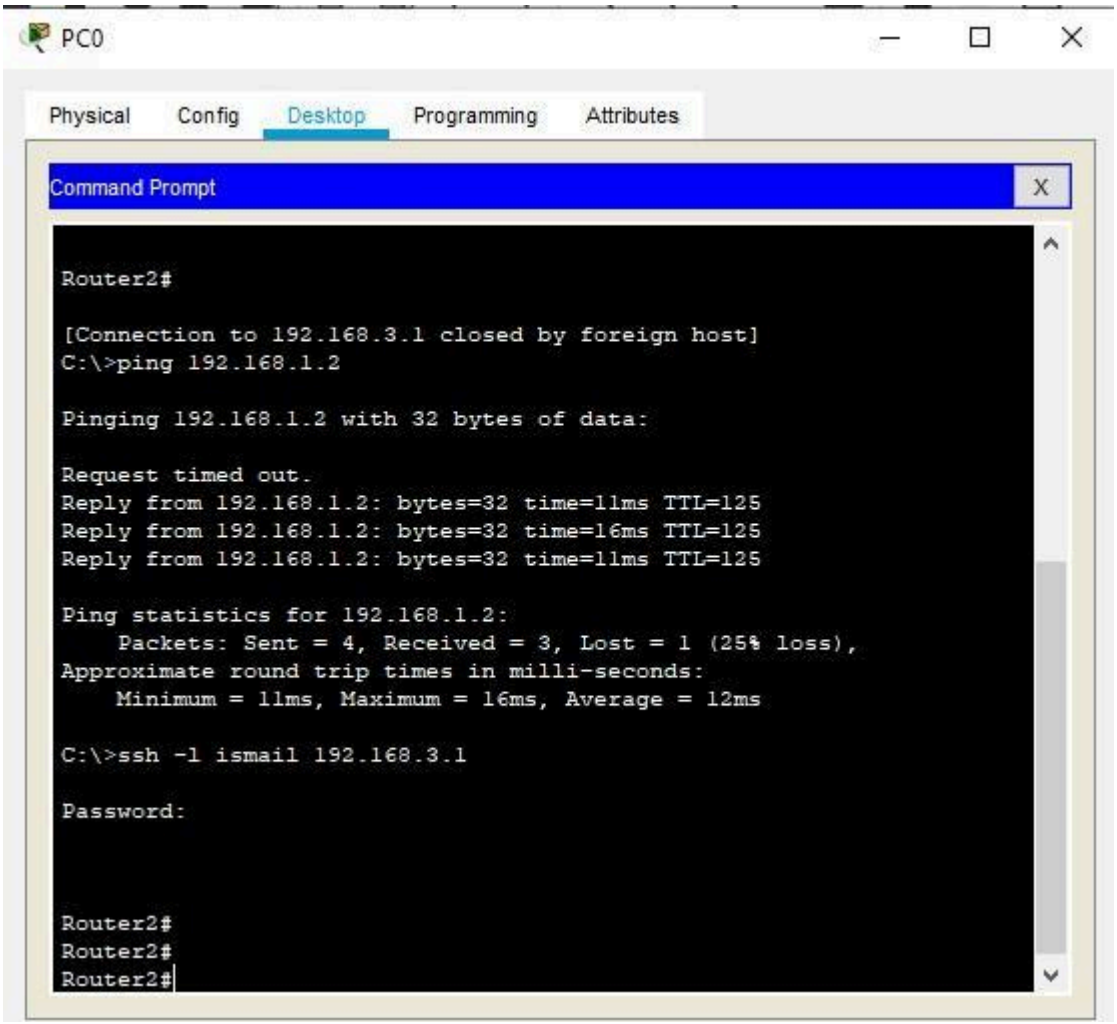
```
map Router(config-pmap- c)#inspect
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)#

Router(config)#zone-pair security in-out-zone source in-zone destination out-zone
Router(config-sec-zone-pair)#service-policy type inspect in-out
Router(config-sec-zone-pair)#exit
Router(config)#

Router(config)#interface GigabitEthernet0/0
Router(config-if)#zone-member security in-zone
Router(config-if)#exit Router(config)#

Router(config)#interface Serial0/1/1
Router(config-if)#zone-member security out-zone
Router(config-if)#exit
Router(config)#exit

Router#copy running-config startup-config
```

Part 4: Testing the Firewall Functionality (from in-zone to out-zone) by the following steps

Step 1: Pinging SERVER from PC (it will succeed)



Step 2: Start an SSH session from PC to Router 2 (192.168.3.1)

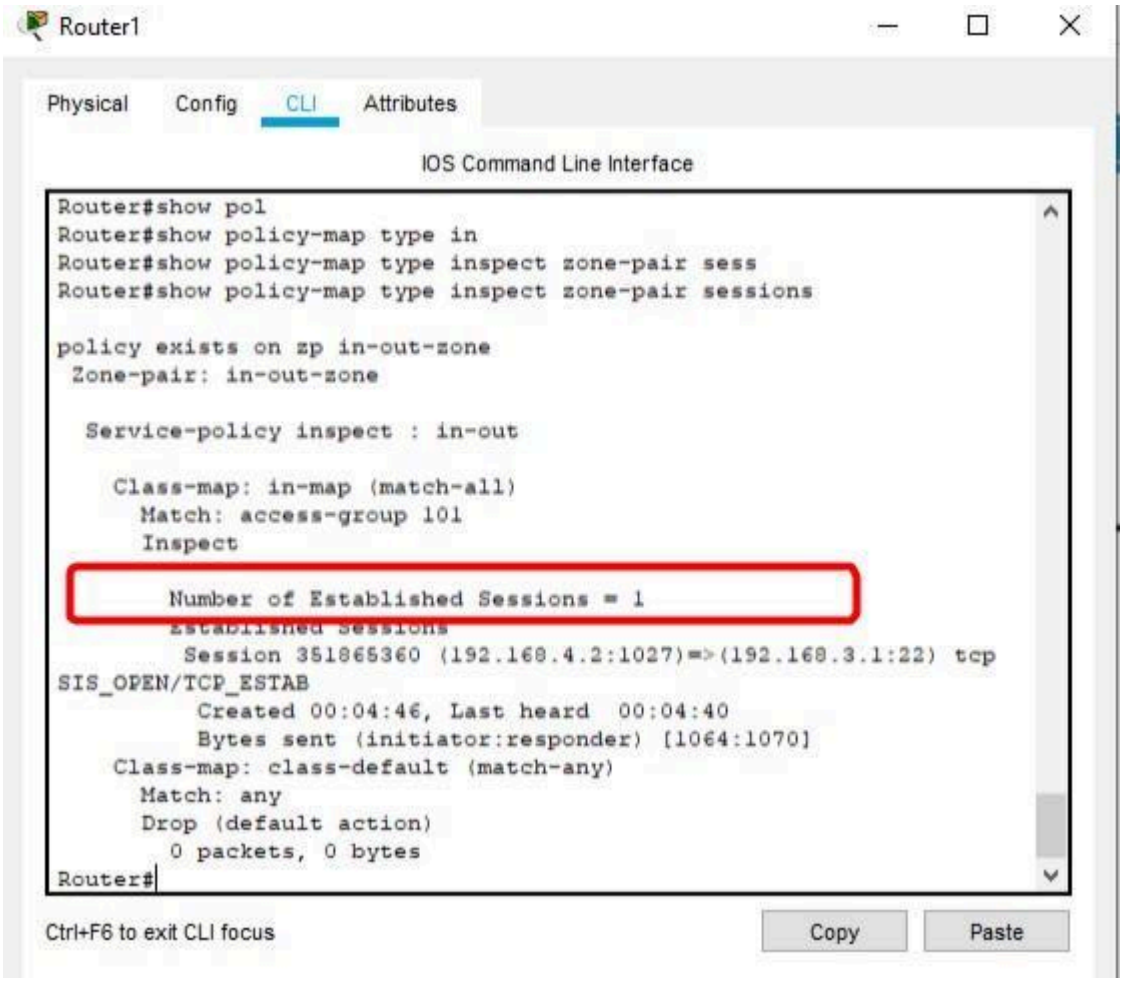


As seen above the session becomes active and we get access to Router2 (Do not exit and the session and continue to Step 3)

Step 3: Type the following command in the CLI mode of Router1

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

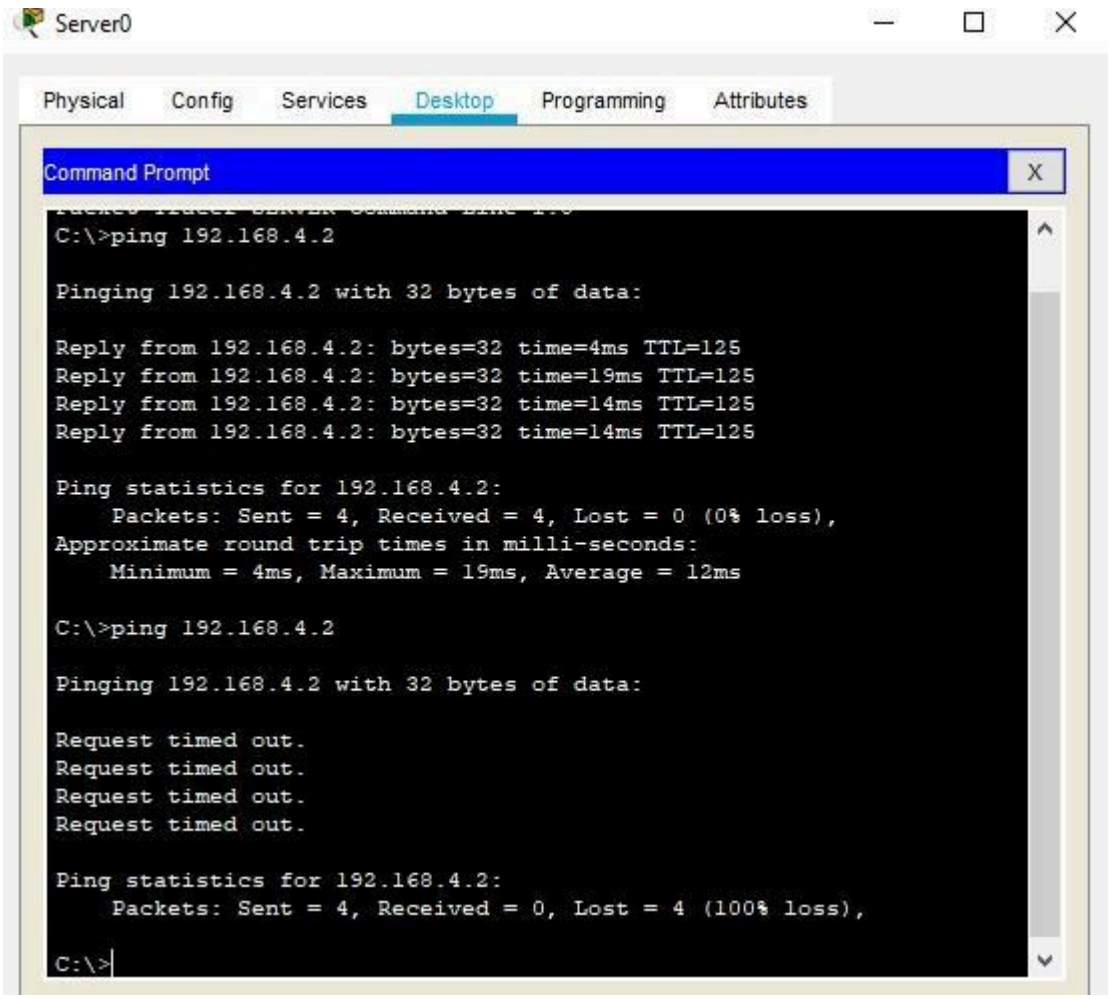
We will get the following output



Step 4: We close the SSH connection and open the web browser and access the server address (192.168.1.2) and get the following

Part 5: Testing the Firewall Functionality (from out-zone to in-zone) by the following steps

Step 1: Ping PC0 from the SERVER (ip 192.168.4.2) (it will result in Failure)



Hence the Firewall functionality has been verified

PRACTICAL NO 6: Configure IOS Intrusion Prevention System (IPS) Using the CLI

The Cisco IOS IPS acts as an in-line intrusion prevention sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. The Signature Event Action Processor (SEAP) can dynamically control actions that are to be taken by a signature event on the basis of parameters such as fidelity, severity, or target value rating. These parameters have default values but can also be configured through CLI. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate:

1) Send an alarm to a syslog server or a centralized management interface

2) Drop the packet

3) Reset the connection

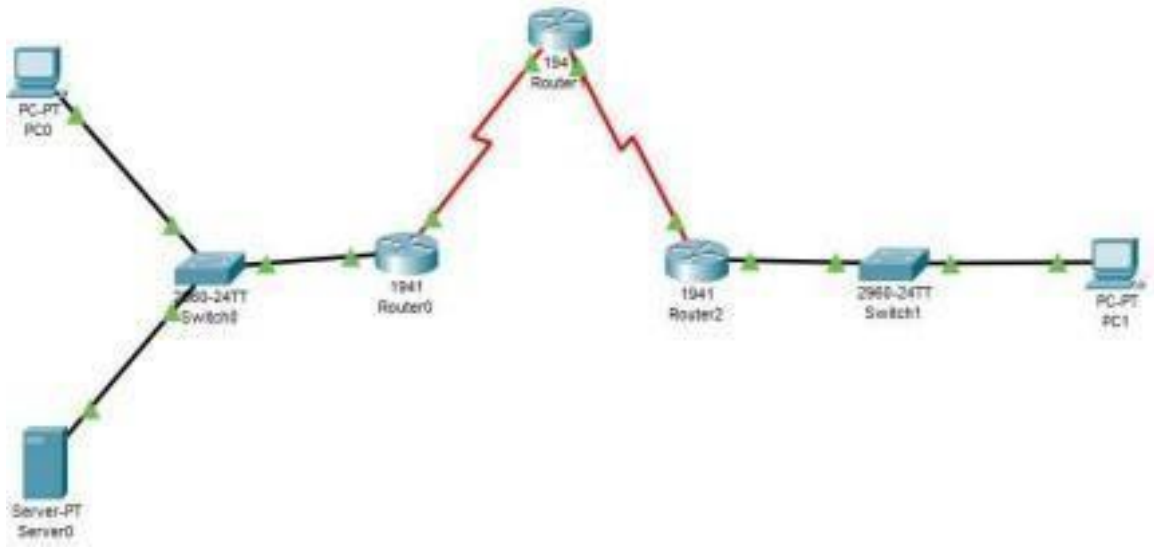
4) Deny traffic from the source IP address of the attacker for a specified amount of time 5) Deny traffic on the connection for which the signature was seen for a specified amount of time

Cisco developed its Cisco IOS software-based intrusion-prevention capabilities and Cisco IOS Firewall with flexibility in mind, so that individual signatures could be disabled in case of false positives. Generally, it is preferable to enable both the firewall and Cisco IOS IPS to support network security policies. However, each of these features may be enabled independently and on different router interfaces.

Signatures:

A signature is a set of rules that an IDS and an IPS use to detect typical intrusive activity, such as DoS attacks. We can easily install signatures using IDS and IPS management software such as Cisco IDM. Sensors enables us to modify existing signatures and define new ones. As sensors scan network packets, they use signatures to detect known attacks and respond with predefined actions. A malicious packet flow has a specific type of activity and signature, and an IDS or IPS sensor examines the data flow using many different signatures. When an IDS or IPS sensor matches a signature with a data flow, the sensor takes action, such as logging the event or sending an alarm to IDS or IPS management software, such as the Cisco SDM

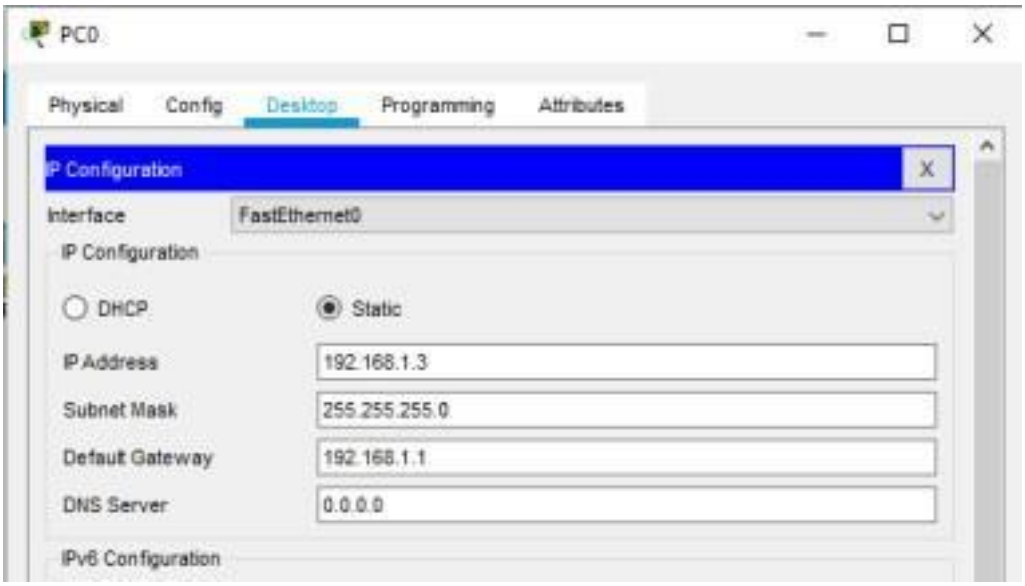
We us the following topology for the present case:



Let us consider the following Address table to configure the network devices:

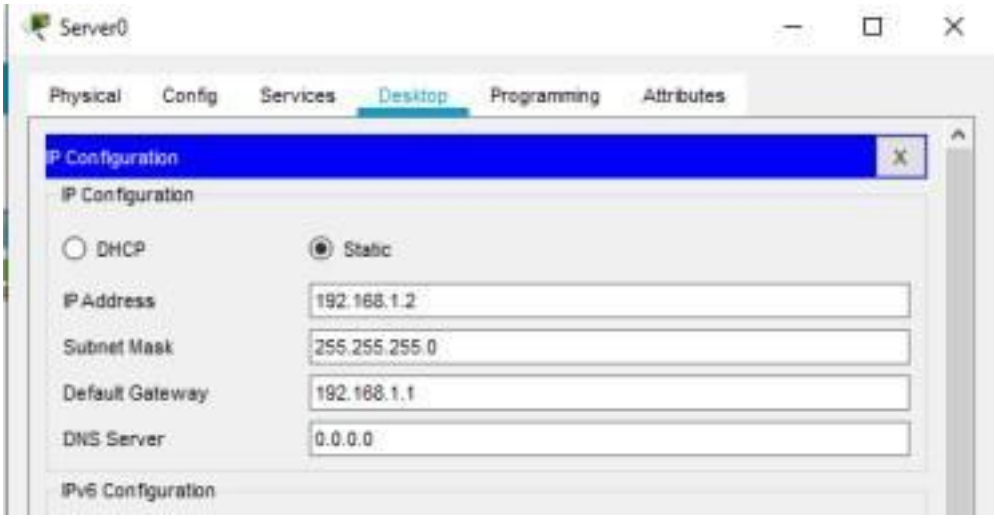
Device	Interface	IP Address	Subnet Mask	Default gateway	Switch Port
PC 0	NA	192.168.1.3	255.255.255.0	192.168.1.1	Switch0 F0/1
PC 1	NA	192.168.4.2	255.255.255.0	192.168.4.1	Switch1 F0/1
Server0	NA	192.168.1.2	255.255.255.0	192.168.1.1	Switch0 F0/2
Router0	GE0/0	192.168.1.1	255.255.255.0	NA	Switch0 F0/5
	S0/1/0	192.168.2.1	255.255.255.0	NA	NA
Router1	S0/1/0	192.168.2.2	255.255.255.0	NA	NA
	S0/1/1	192.168.3.1	255.255.255.0	NA	NA
Router2	S0/1/1	192.168.3.2	255.255.255.0	NA	NA
	GE0/0	192.168.4.1	255.255.255.0	NA	Switch1 F0/5

Configuring PC0

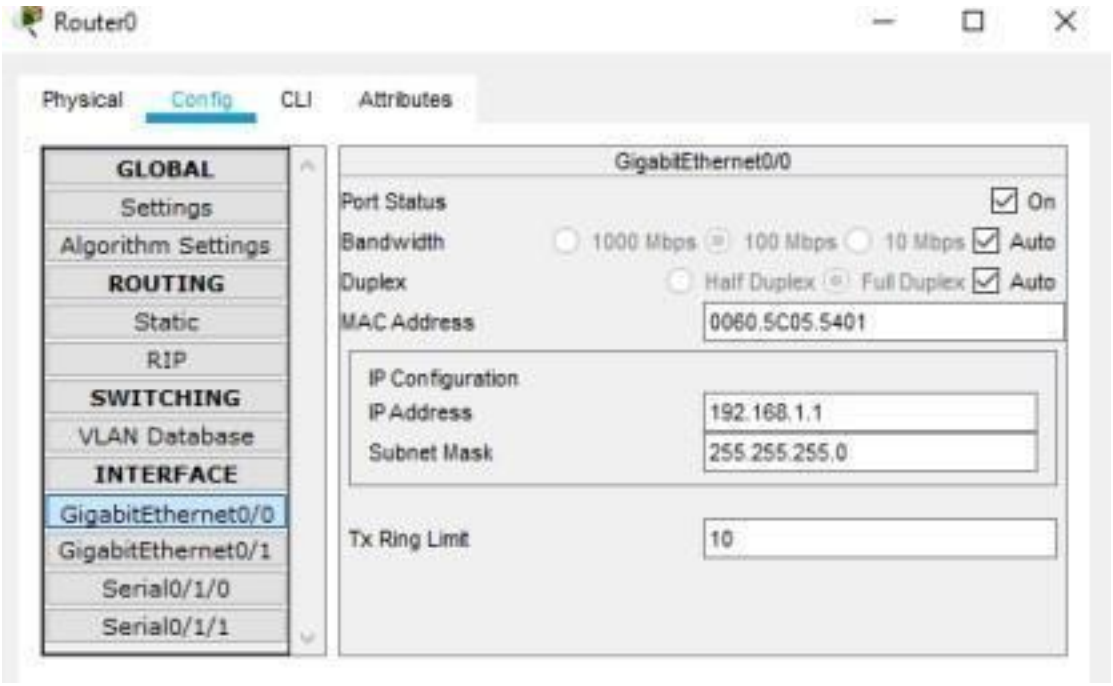


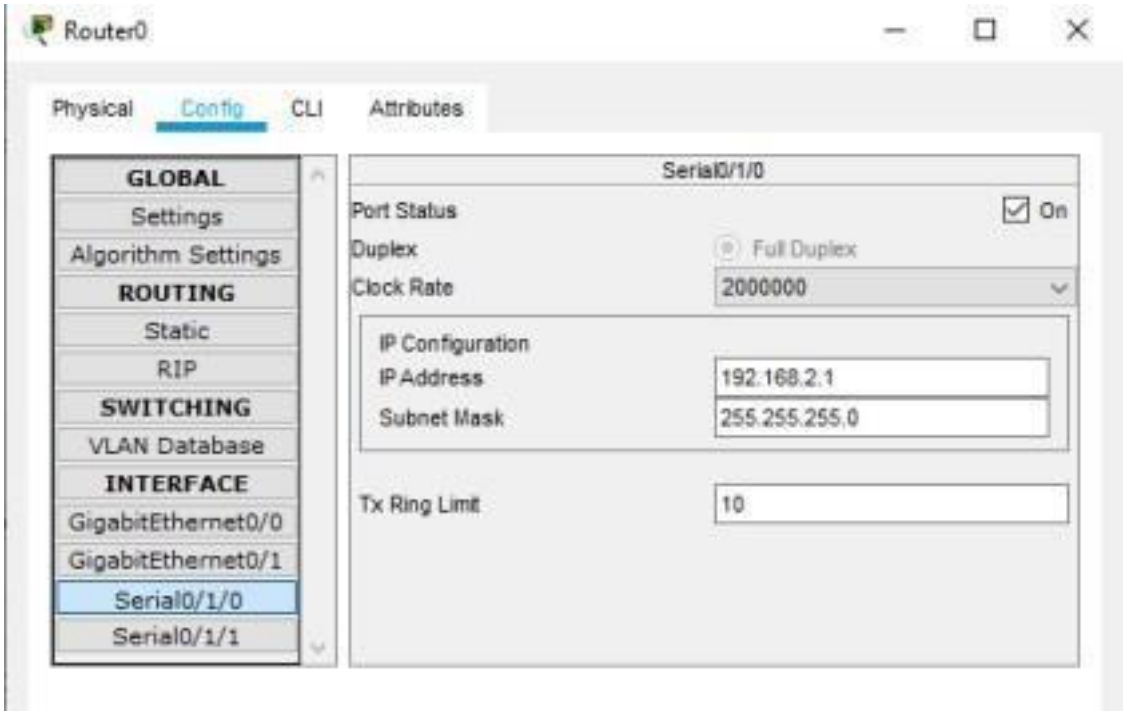
Configuring PC1

Configuring Server0

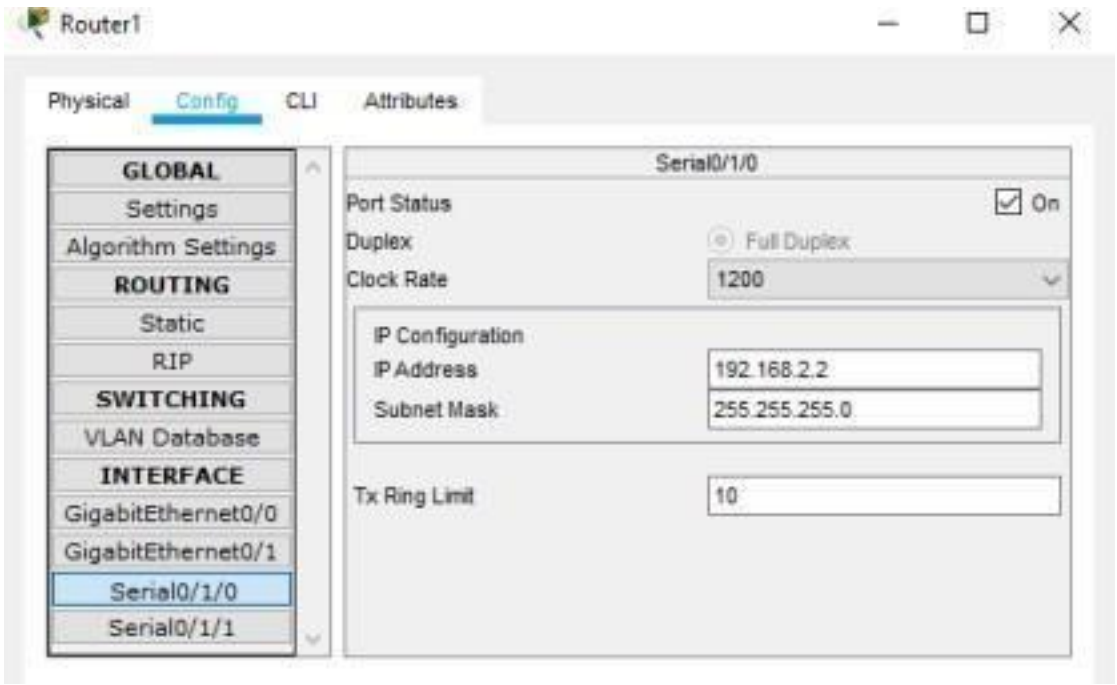


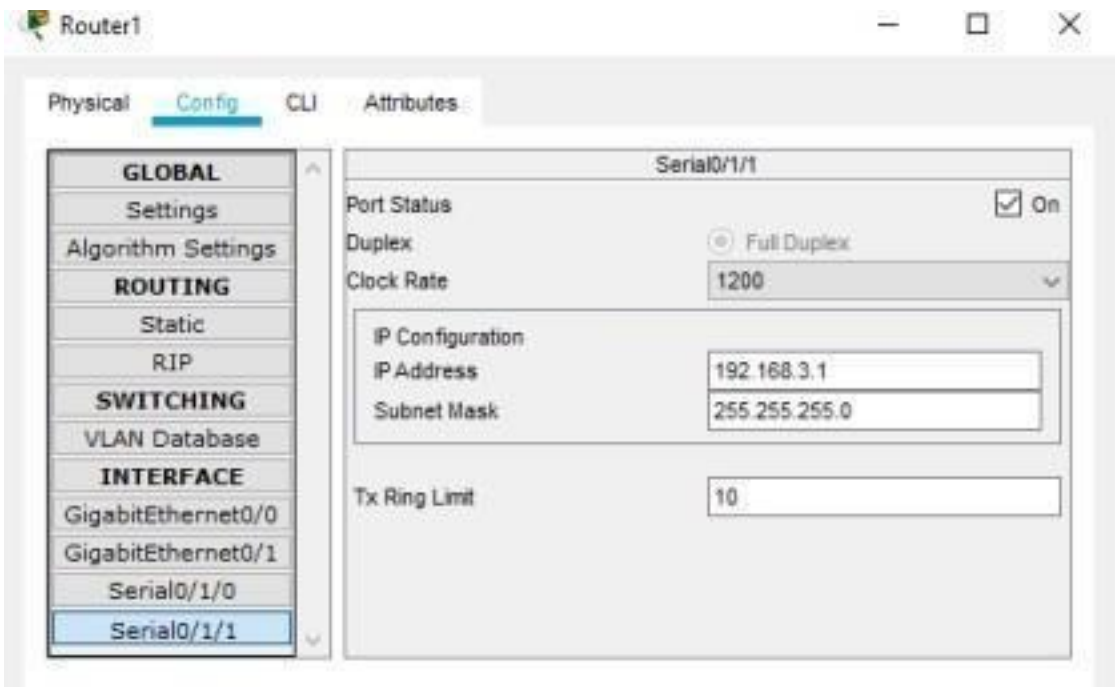
Configuring Router0



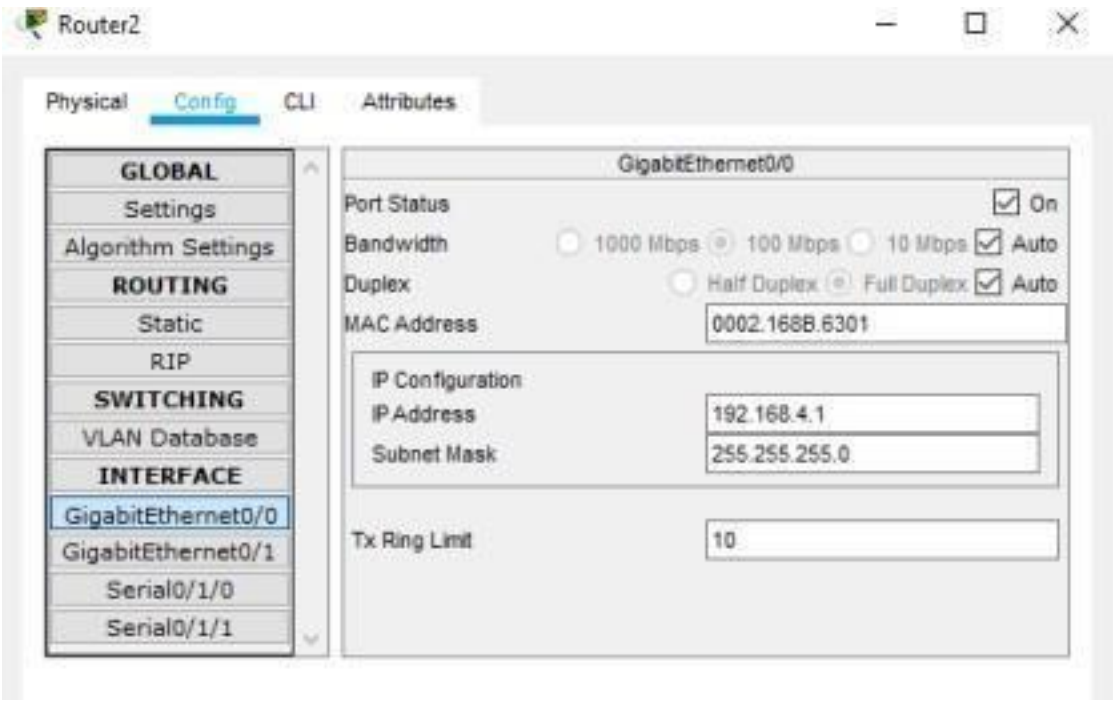


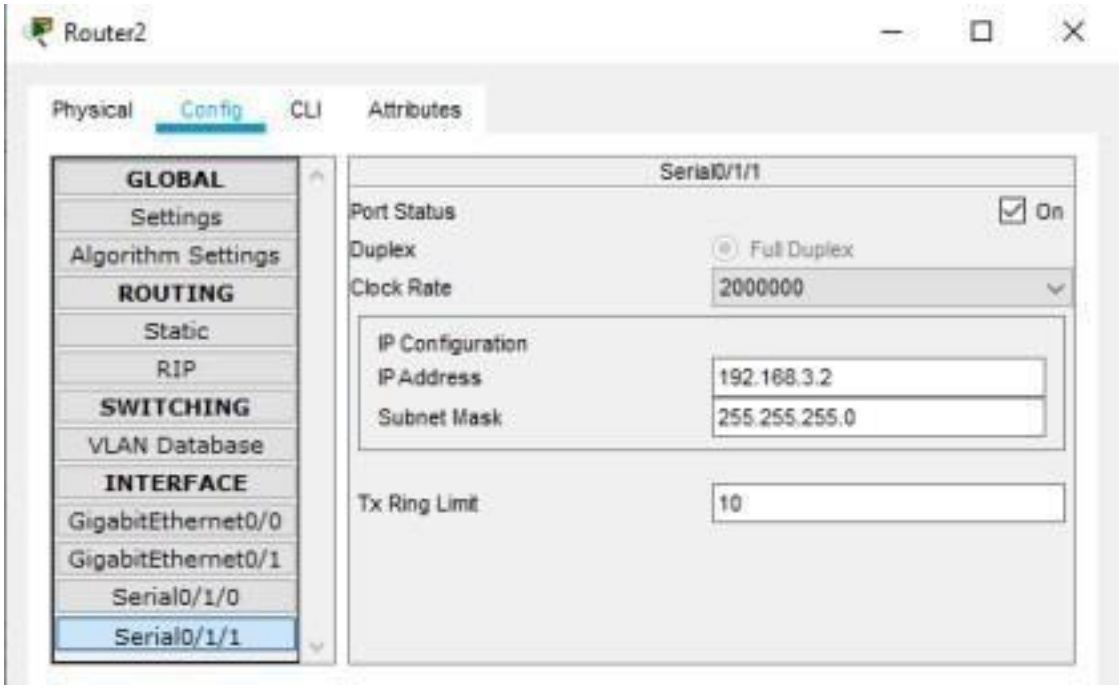
Configuring Router1



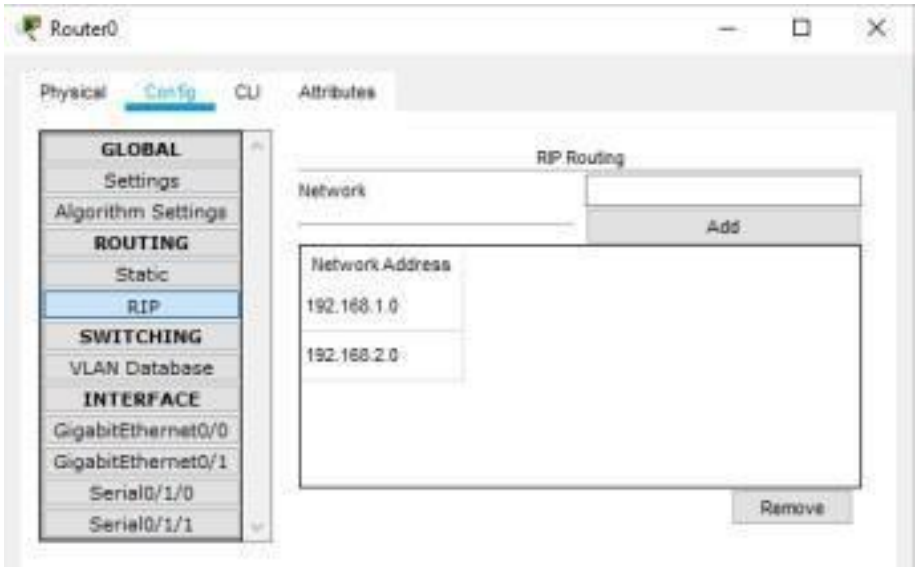


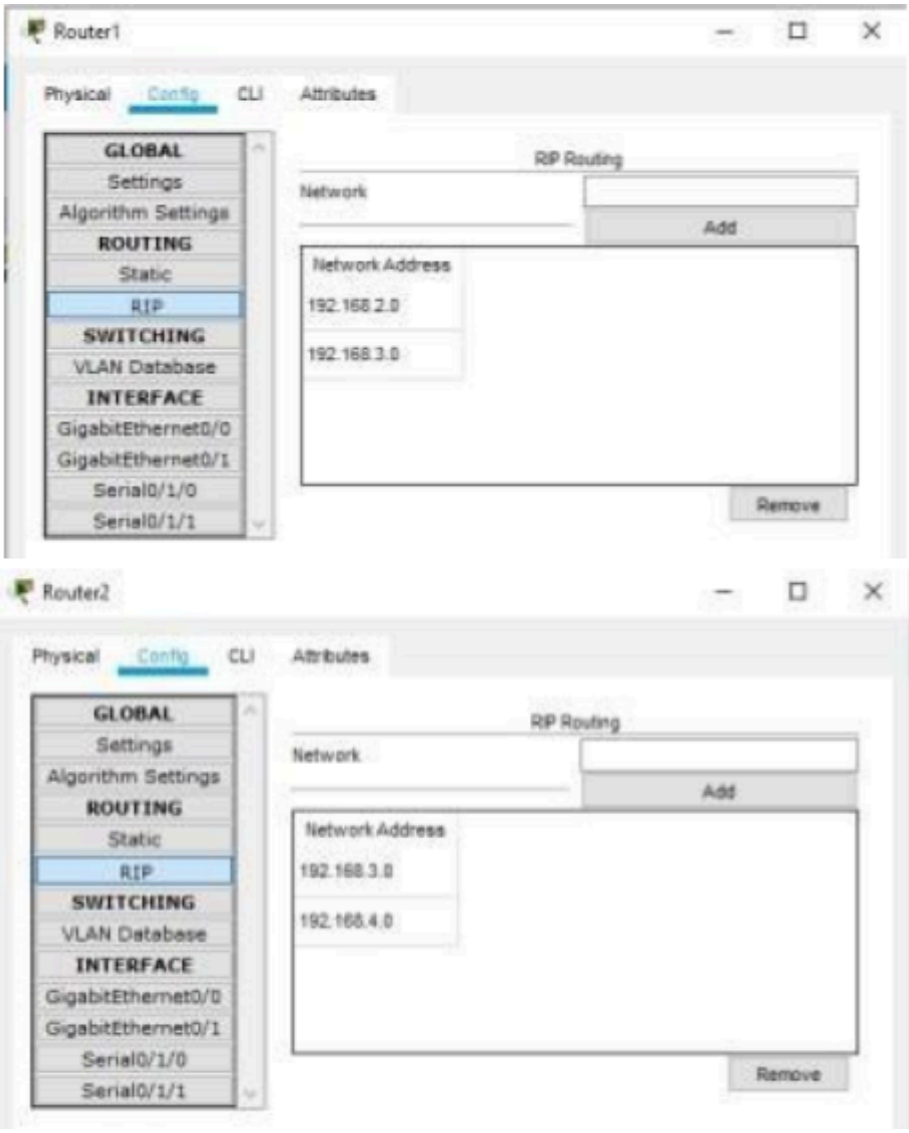
Configuring Router2



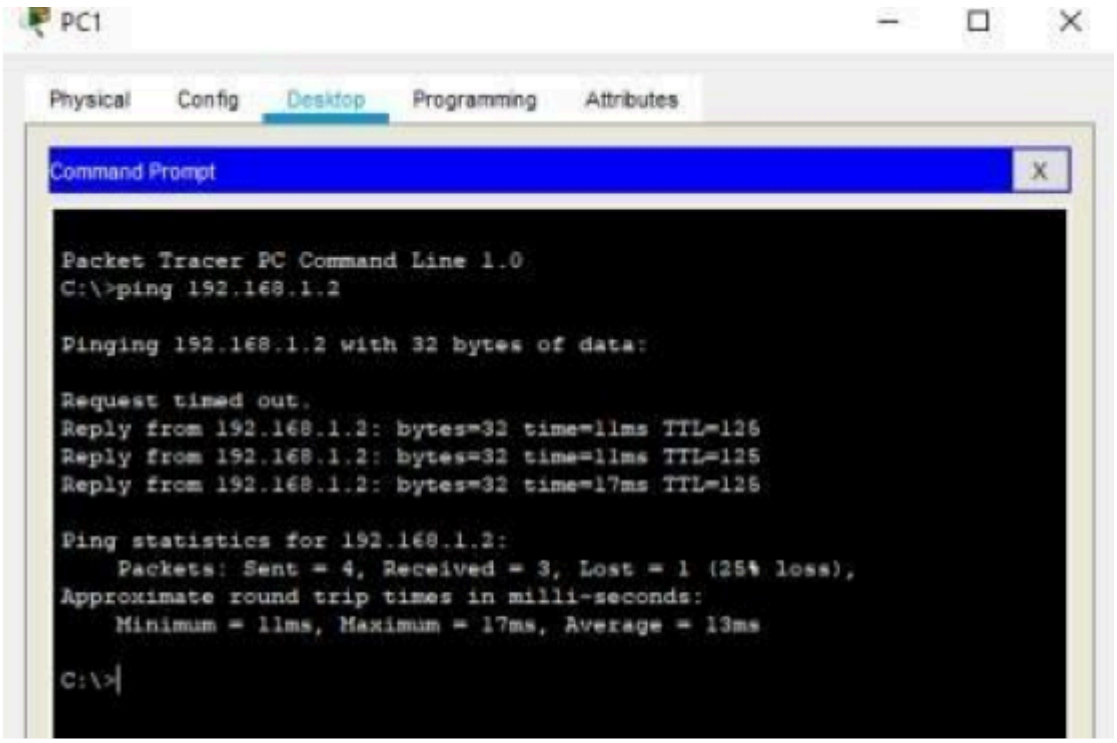


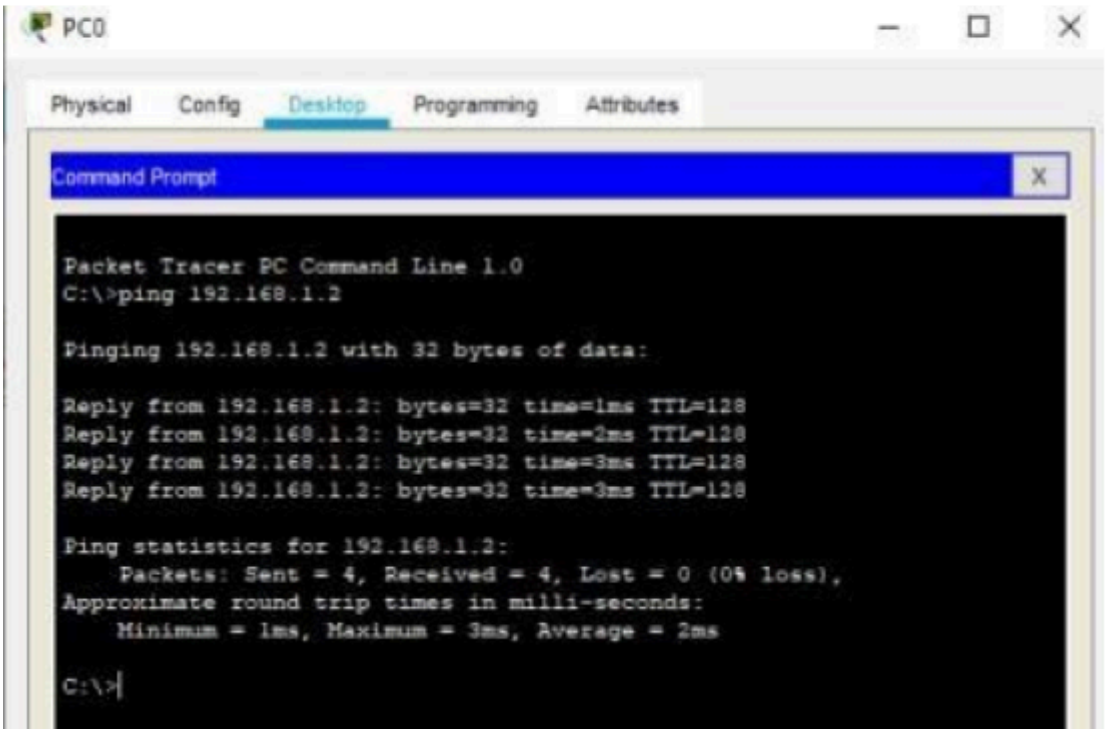
We need to set the Routing table in all the Routers so that each node could send and receive packets from others (RIP is set in all the Routers as follows)





Now we can check the connectivity by sending ping commands from any node to any other node



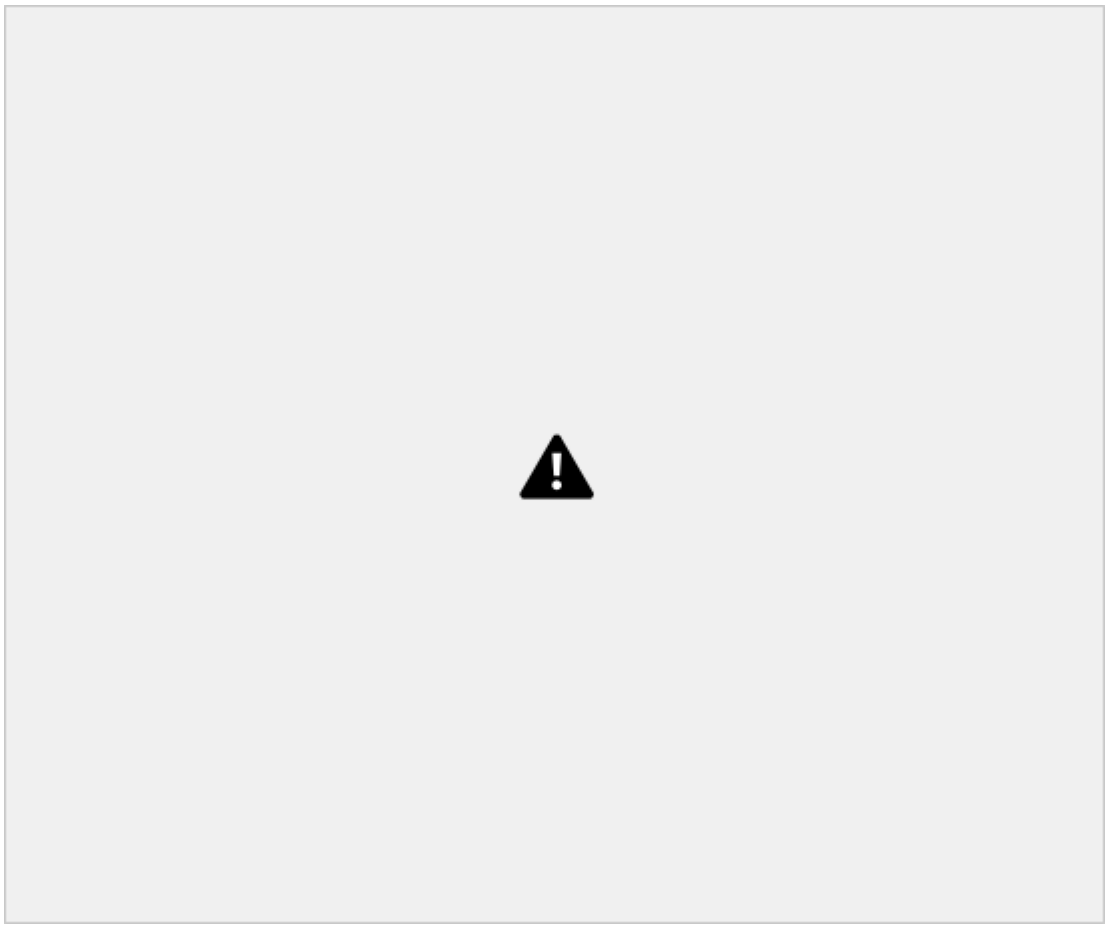


So, we conclude that the connectivity has been established

PART1: Enable the IOS IPS (on Router1)

Type the following command in the CLI mode of Router1

Router#show version
We will get a message informing whether the security Package is enabled or not



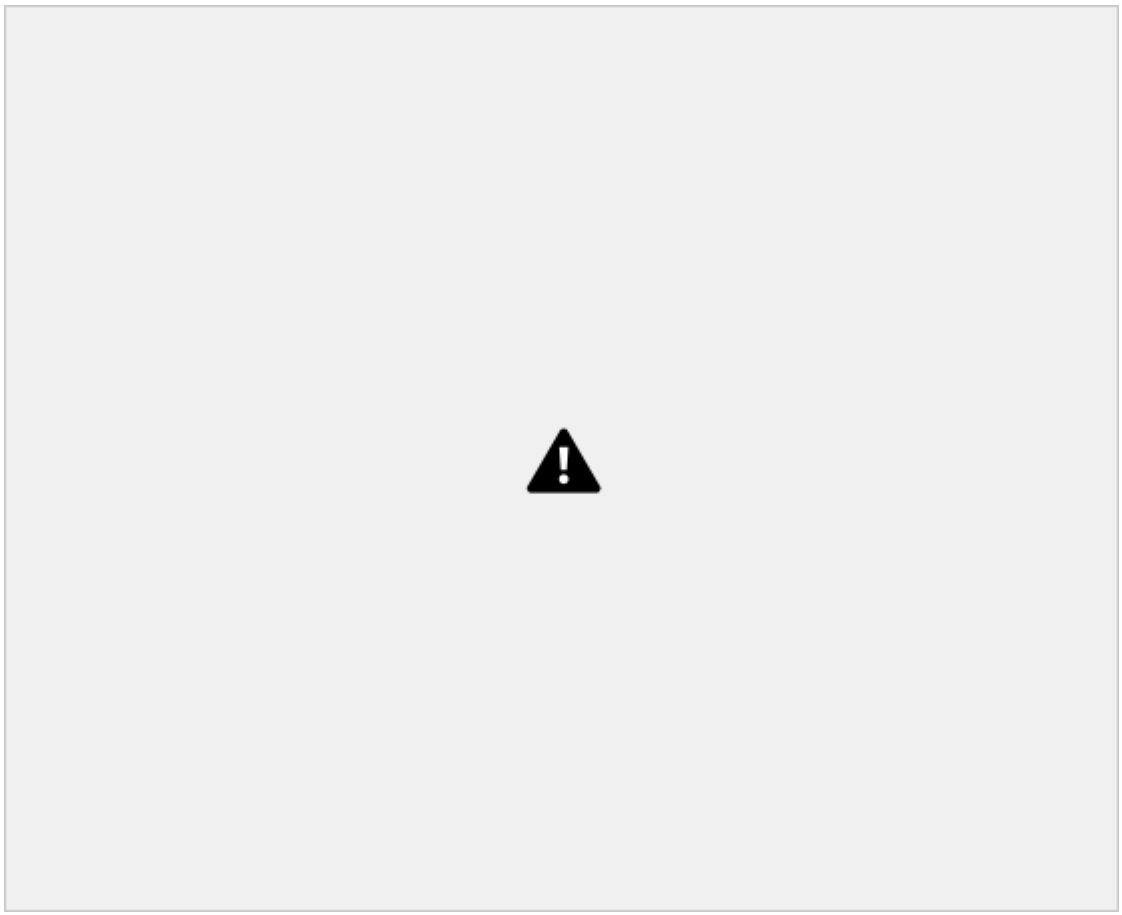
As seen above the security package is not enabled, to enable the security feature, type the following command in Router1

Router#configure terminal
Router(config)#license boot module c1900 technology-package securityk9
ACCEPT? [yes/no]: y

Press enter key

Router#
Router#reload
System configuration has been modified. Save? [yes/no]:y
Proceed with reload? [confirm] Press Enter key
Press RETURN to get started! Press Enter key

Router>enable
Router# Router#show version
We will get a message informing whether the security package is enabled or not



As seen above now the security package has been enabled

Now type the following commands in the CLI mode of Router1

Router#
Router#clock set 10:30:45 march 3 2022 Router#mkdir
smile
Create directory filename [smile]? Press enter key Created
dir flash:smile

Router#

Router#configure terminal
Router(config)#ip ips config location flash:smile

```
Router(config)#ip ips name iosips
Router(config)#ip ips notify log
Router(config)#ip ips signature-category
Router(config-ips-category)#category all

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

Router(config)#interface Serial0/1/0
Router(config-if)#ip ips iosips out
Router(config-if)#
Press enter key
Router(config-if)#exit
Router(config)#
```

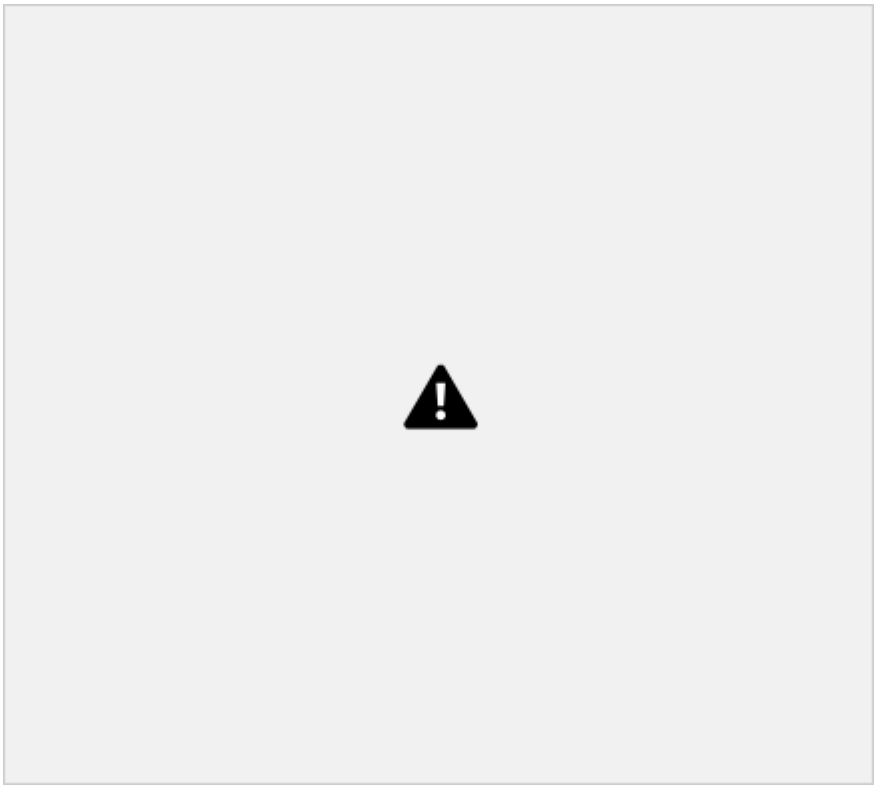
Part 2: Modify the Signature

Type the following commands in the CLI mode of Router1

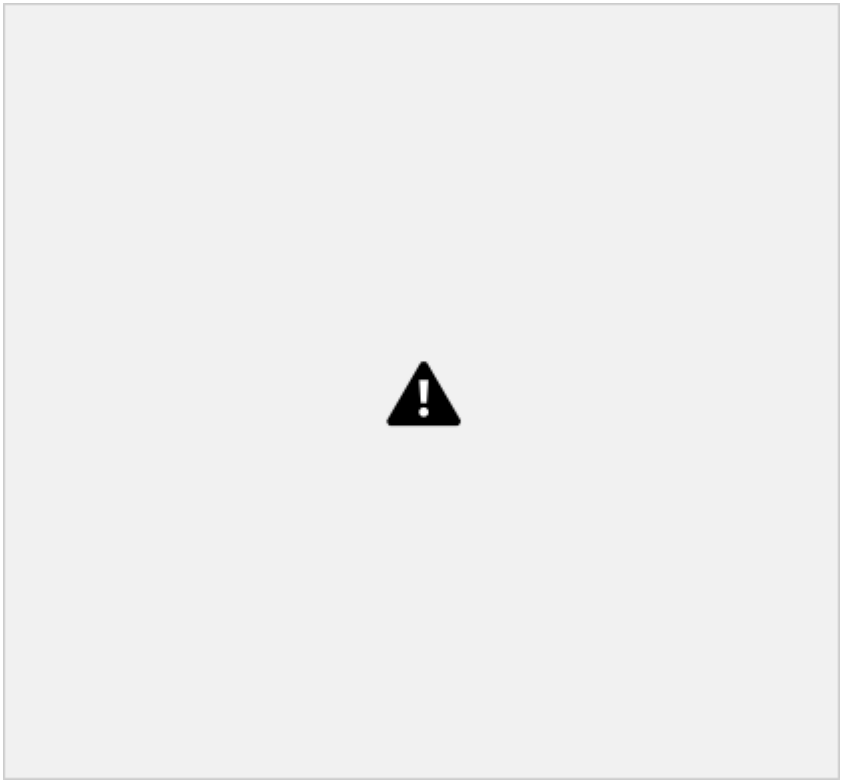
```
Router(config)#
Router(config)#ip ips signature-definition
Router(config-sigdef)#signature 2004 0
Router(config-sigdef-sig)#status
Router(config-sigdef-sig-status)#retired false
Router(config-sigdef-sig-status)#enabled true
Router(config-sigdef-sig-status)#exit
Router(config-sigdef-sig)#engine
Router(config-sigdef-sig-engine)#event-action produce-alert
Router(config-sigdef-sig-engine)#event-action deny-packet-inline
Router(config-sigdef-sig-engine)#exit
Router(config-sigdef-sig)#exit
Router(config-sigdef)#exit
Do you want to accept these changes? [confirm]y Router(config)#
```

Now we need to verify the above IPS configuration, we do it first by pinging PC1 to SERVER and then from SERVER to PC1

PC1 to SERVER – The ping fails



Server to PC1 – The Ping is successful



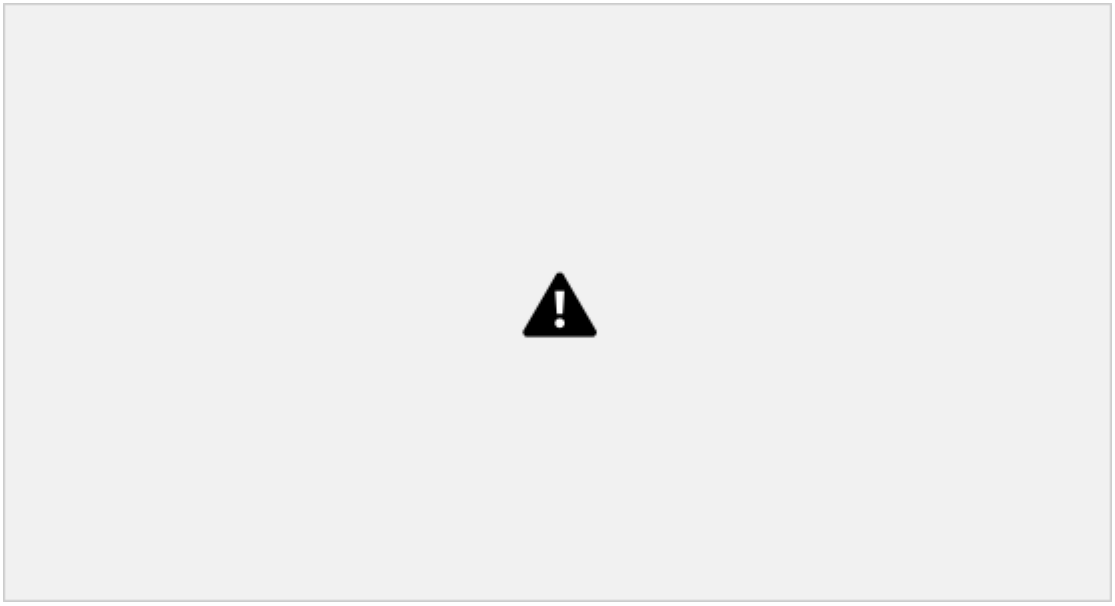
We check the Syslog service on the server to check the logging activity, by typing the following commands in Router0

Router>enable
Router#configure terminal
Router(config)#logging 192.168.1.2
Router(config)# Router(config)#
Router(config)#exit
Router#

Router#ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/3 ms

Router#



Hence, we set the IPS and also verified it on Router1

Practical 7: Packet Tracer - Layer 2 Security Topology

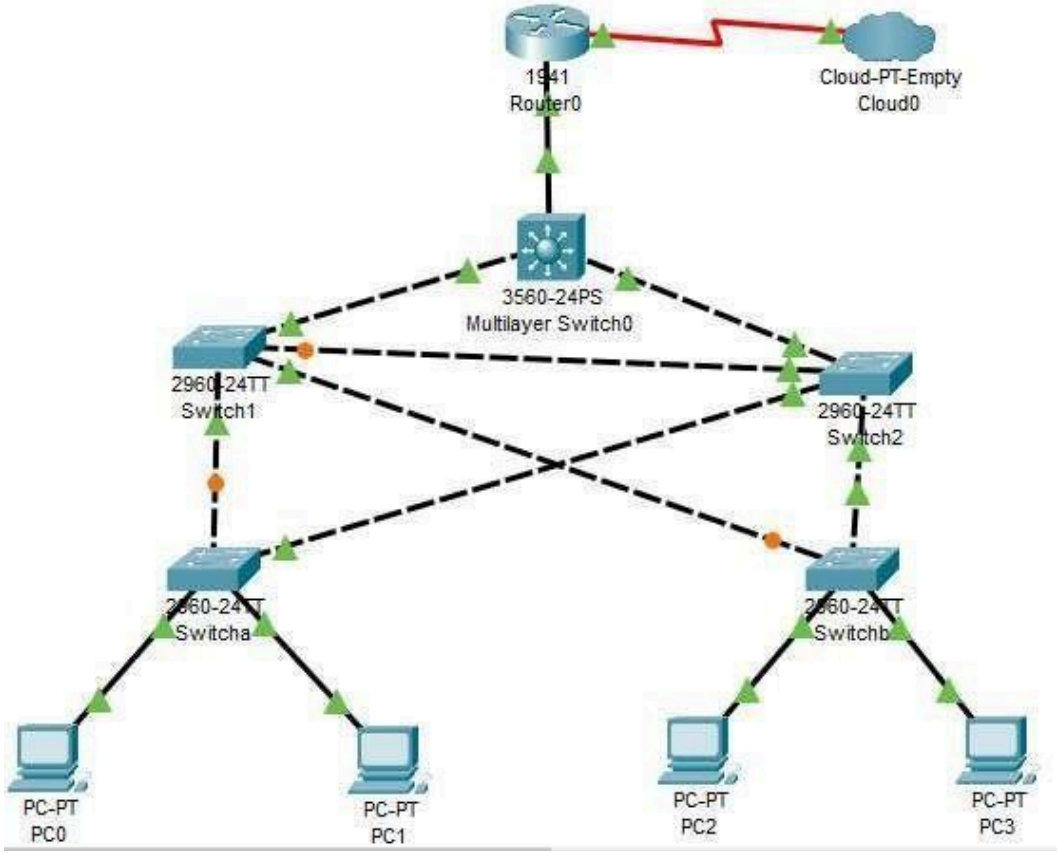
Objectives

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

Background / Scenario

There have been a number of attacks on the network recently. For this reason, the network administrator has assigned you the task of configuring Layer 2 security. For optimum performance and security, the administrator would like to ensure that the root bridge is the 3560 Central switch. To prevent spanning-tree manipulation attacks, the administrator wants to ensure that the STP parameters are secure. To prevent against CAM table overflow attacks, the network administrator has decided to configure port security to limit the number of MAC addresses each switch port can learn. If the number of MAC addresses exceeds the set limit, the administrator would like the port to be shutdown.

Let us consider the following topology to present this case:



Let us consider the following interface table to connect the network devices:

Note: Add one Serial Port in Router 0 and in Empty Cloud 0.

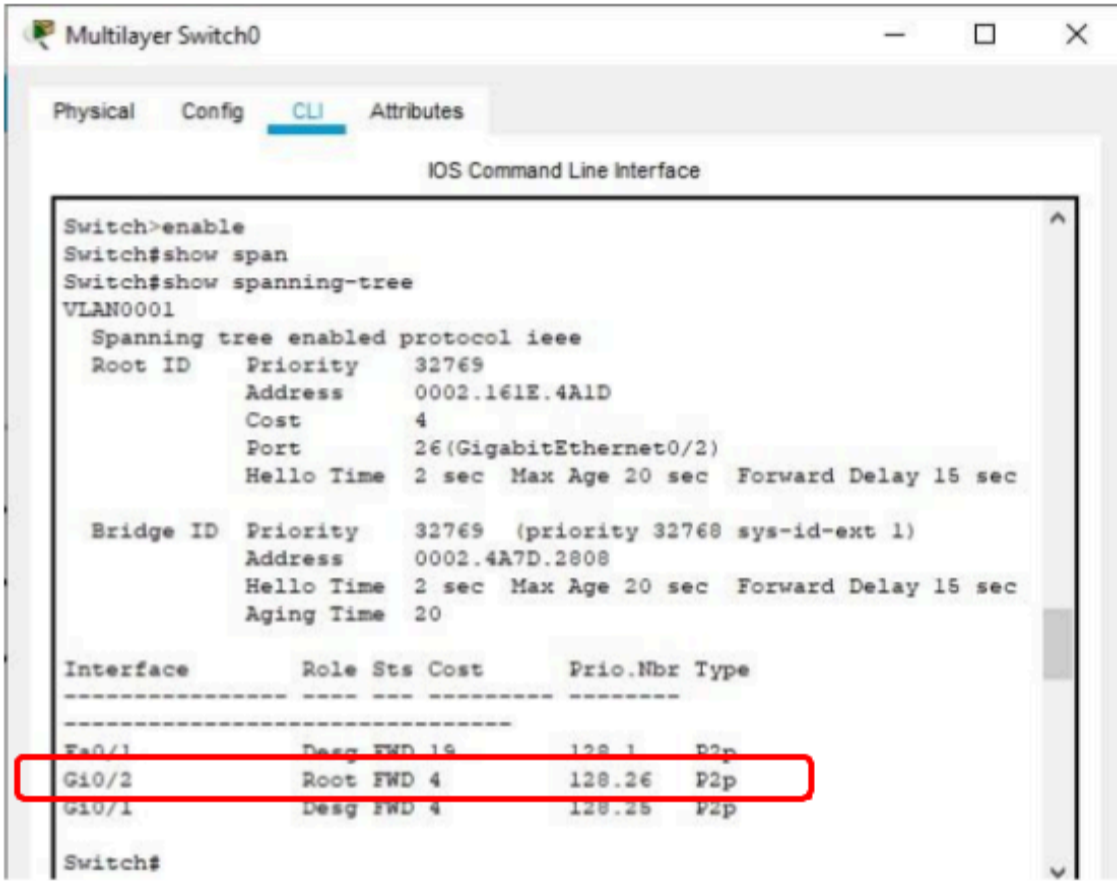
Device	Interface	Switch Port
PC 0	FastEthernet0	Switcha F0/1
PC 1	FastEthernet0	Switcha F0/2
PC 2	FastEthernet0	Switchb F0/1
PC 3	FastEthernet0	Switchb F0/2
Switch a	F0/23	Switch1 F0/23

	F0/24	Switch2 F0/1
Switch b	F0/23	Switch2 F0/23
	F0/24	Switch1 F0/1
Switch 1	F0/24	Switch2 F0/24
	GE 0/1	Multilayer Switch0 GE 0/1
Switch 2	GE 0/1	Multilayer Switch0 GE 0/2
Router 0	GE 0/1	Multilayer Switch0 F0/1
	S0/1/0	Cloud0 S4

Part 1: Configure Root Bridge

Type the following command in CLI mode of Multilayer Switch0, to check which is the Root bridge

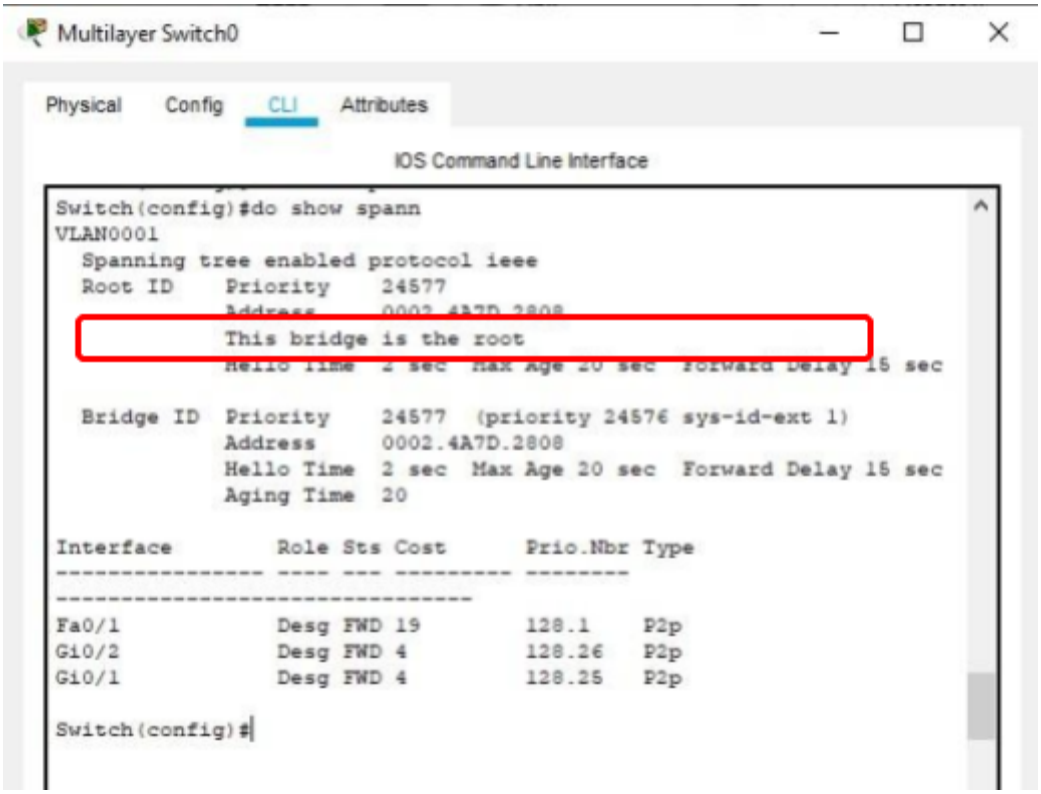
Switch>enable
Switch#show spanning-tree



The output shows that the bridge connected to GigabitEthernet 0/2 is the Root Bridge, i.e., Switch 2 is the Root Bridge in the above topology.

Now we need to make Multilayer Switch0 as the Root Bridge. Type the following commands in the CLI mode of Multilayer Switch0.

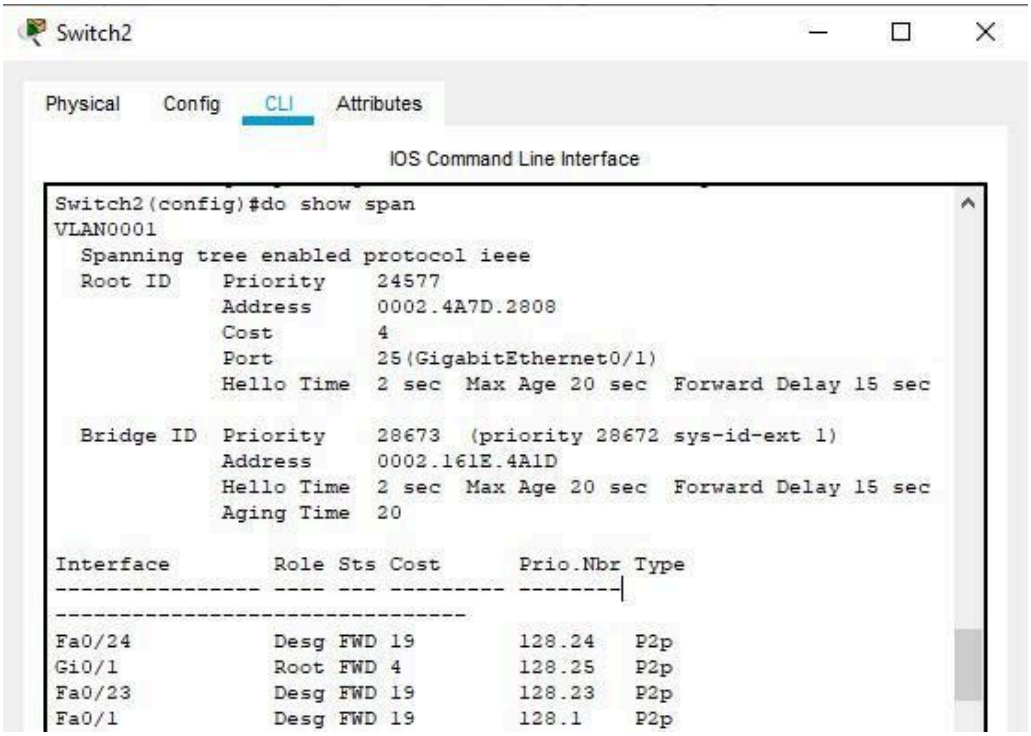
Switch#
Switch#configure terminal
Switch(config)#spanning-tree vlan 1 root primary
Switch(config)#do show spann



Now, we have made the Multilayer Switch0 as the Root Bridge.

But we also need to remove the Switch2 from Root Bridge. For that open the CLI mode of Switch2 and type the following code.

Switch2#configure terminal
Switch2(config)#spanning-tree vlan 1 root secondary
Switch2(config)#do show span



Thus, we have successfully made the central (Multilayer Switch0) as the Root Bridge.

Part 2: Protect Against STP Attacks

Open CLI mode of Switch a and type the following command

```
Switcha>enable
Switcha#configure termial
Switcha(config)#interface range fastEthernet 0/1-2
Switcha(config-if-range)#switchport mode access
Switcha(config-if-range)#spanning-tree portfast
Switcha(config-if-range)#spanning-tree bpduguard enable
```

Now minimize the Switch a window and open the Switch b CLI mode and type the same command

```
Switchb>enable
Switchb#configure termial
Switchb(config)#interface range fastEthernet 0/1-2
Switchb(config-if-range)#switchport mode access
Switchb(config-if-range)#spanning-tree portfast
Switchb(config-if-range)#spanning-tree bpduguard enable
```

Now minimize the Switch b window and open the Switch 1 CLI mode and type the following command

```
Switch1>enable
Switch1#configure terminal
Switch1(config)#interface range fastEthernet 0/23-24
Switch1(config-if-range)#spanning-tree guard root
```

Now minimize the Switch 1 window and open the Switch 2 CLI mode and type the same command

```
Switch2>enable
Switch2#configure terminal
Switch2(config)#interface range fastEthernet 0/23-24
Switch2(config-if-range)#spanning-tree guard root
```

Thus, we have Protected all the switch against STP Attacks.

Part 3: Configure Port Security and Disable unsed ports

Open CLI mode of Switch a and type the following command

Switcha(config-if-range)#switchport port-security
Switcha(config-if-range)#switchport port-security maximum 2
Switcha(config-if-range)#switchport port-security mac-address sticky
Switcha(config-if-range)#switchport port-security violation shutdown

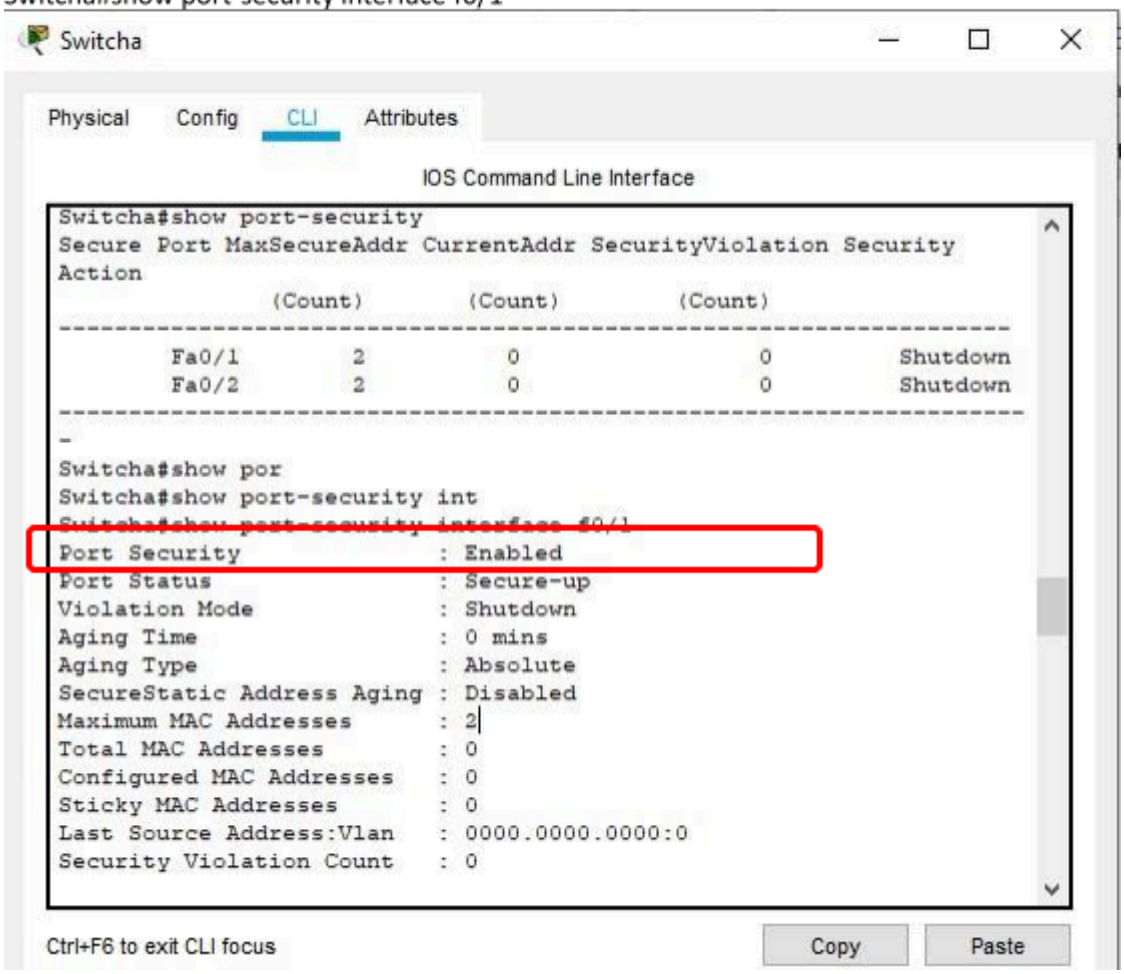
Now minimize the Switch a window and open the Switch b CLI mode and type the same command

Switchb(config-if-range)#switchport port-security
Switchb(config-if-range)#switchport port-security maximum 2
Switchb(config-if-range)#switchport port-security mac-address sticky
Switchb(config-if-range)#switchport port-security violation shutdown

Now let us check if the security is enabled or not. Open CLI mode of Switch a and type the following

Switcha(config-if-range)# CTRL Z

Switcha#show port-security interface f0/1



Let us now disable all the unused ports in switch a and switch b.

Open the CLI mode of Switch a and type the following command

Switcha#enable

Switcha#configure terminal

Switcha(config)#interface range fastEthernet 0/3-22 Switcha(config-if-range)#shutdown

Open the CLI mode of Switch b and type the following command

Switchb#enable

Switchb#configure terminal

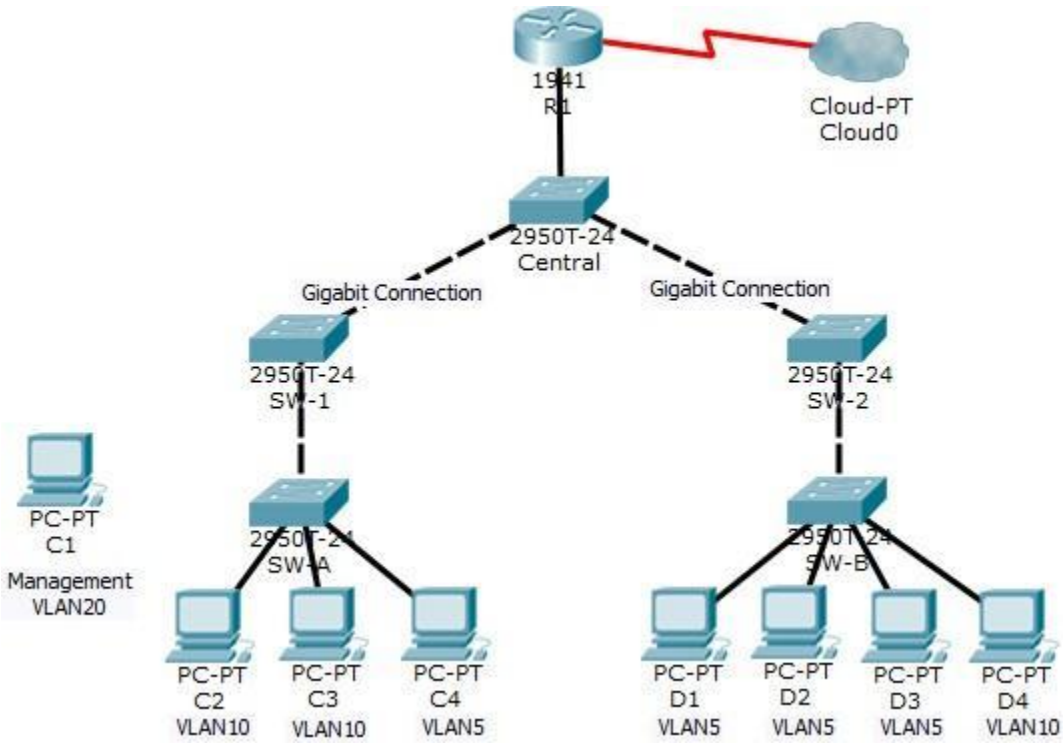
Switchb(config)#interface range fastEthernet 0/3-22 Switchb(config-if-range)#shutdown

Thus, Port Security is enabled and all the unsed ports are disabled.

PRACTICAL - 8

Packet Tracer - Layer 2 VLAN Security

Topology



Objectives

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

Background / Scenario

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

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

All devices have been preconfigured with: **o**

Enable secret password: **ciscoenpa55 o**

Console password: **ciscoconpa55**

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

Part 1: Verify Connectivity

Step 1: Verify connectivity between C2 (VLAN 10) and C3 (VLAN 10). Step 2: Verify connectivity between C2 (VLAN 10) and D1 (VLAN 5).

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

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

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

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

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

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

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

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

Part 3: Enable VLAN 20 as a Management VLAN

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
SW-B(config)# interface vlan 20
SW-B(config-if)# ip address 192.168.20.2 255.255.255.0
SW-1(config)# interface vlan 20
SW-1(config-if)# ip address 192.168.20.3 255.255.255.0
SW-2(config)# interface vlan 20
SW-2(config-if)# ip address 192.168.20.4 255.255.255.0
Central(config)# interface vlan 20
Central(config-if)# ip address 192.168.20.5 255.255.255.0
```

Step 3: Connect and configure the management PC.

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

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

Interface F0/1 must be part of VLAN 20.

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

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

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

Part 4: Enable the Management PC to Access Router R1

Step 1: Enable a new subinterface on router R1.

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

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

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

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

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

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

Step 3: Enable security.

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

a. Create an ACL that allows only the Management PC to access the router.

Example: (may vary from student configuration)

```
R1(config)# access-list 101 deny ip any 192.168.20.0 0.0.0.255
R1(config)# access-list 101 permit ip any any
R1(config)# access-list 102 permit ip host 192.168.20.50 any
```

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

Example: (may vary from student configuration)

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

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

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

Step 4: Verify security.

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

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

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

_____ _____ _____

_____ _____ _____

_____ _____

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

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

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

Step 5: Check results.

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

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

!!! Script for SW-1

```
conf t interface f0/23
switchport mode trunk
switchport trunk native vlan 15
switchport nonegotiate  no
shutdown vlan 20  exit interface
vlan 20
 ip address 192.168.20.3 255.255.255.0
```

!!! Script for SW-2

```
conf t interface f0/23
switchport mode trunk
switchport trunk native vlan 15
switchport nonegotiate  no
shutdown vlan 20  exit interface
vlan 20
 ip address 192.168.20.4 255.255.255.0
```

!!! Script for SW-A

```
conf t vlan 20  exit interface vlan 20
ip address 192.168.20.1 255.255.255.0
interface f0/1  switchport access
vlan 20  no shutdown
```

!!! Script for SW-B

```
conf t vlan 20
exit interface
vlan 20
```

```
 ip address 192.168.20.2 255.255.255.0
```

!!! Script for Central

```
conf t vlan 20
exit interface
vlan 20
 ip address 192.168.20.5 255.255.255.0
```

!!! Script for R1

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