**Practical 1**

**Configure Cisco Routers for Syslog, NTP, and SSH Operations**

**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.



### 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.

### SYSLOG server

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 tolog different types of events.

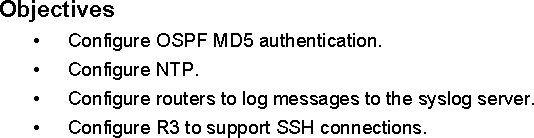
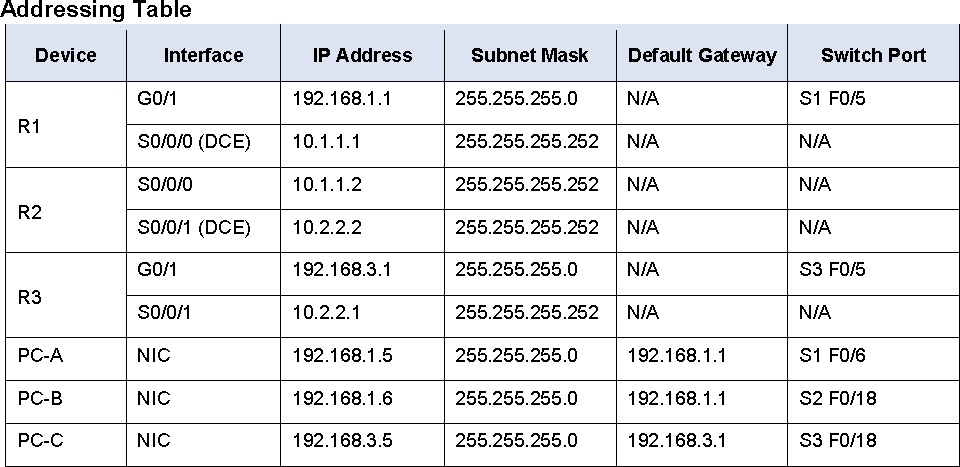
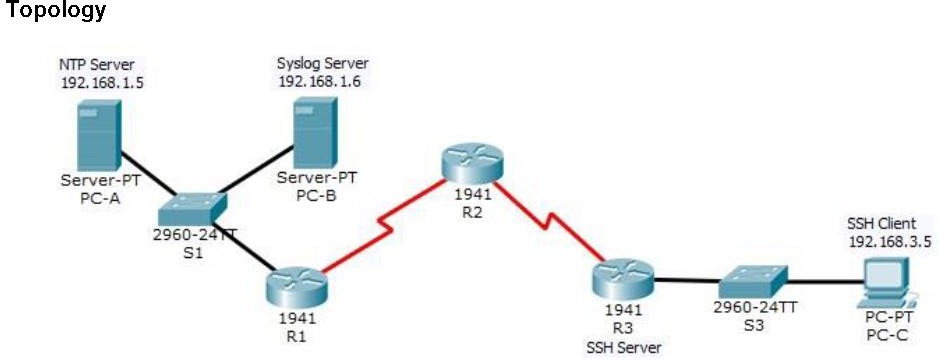
For example, a router might send messages about users logging on to console sessions, while a web-server might log access-denied events.

### SSH

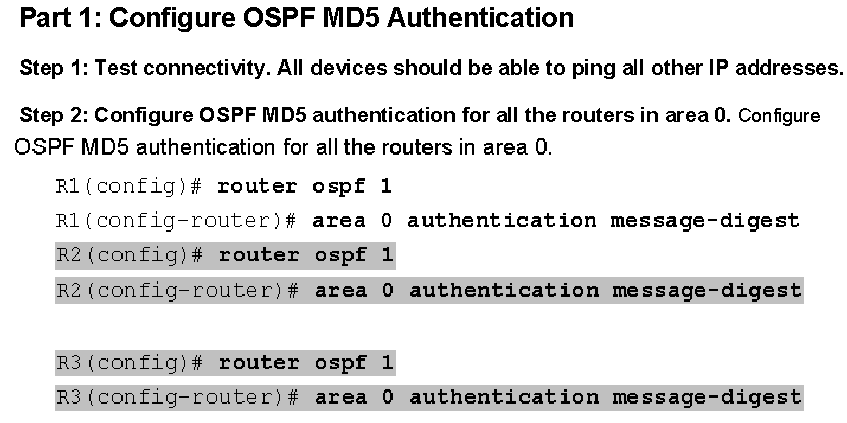
An SSH server is a software program which uses the secure shell protocol to accept connections from remote computers.

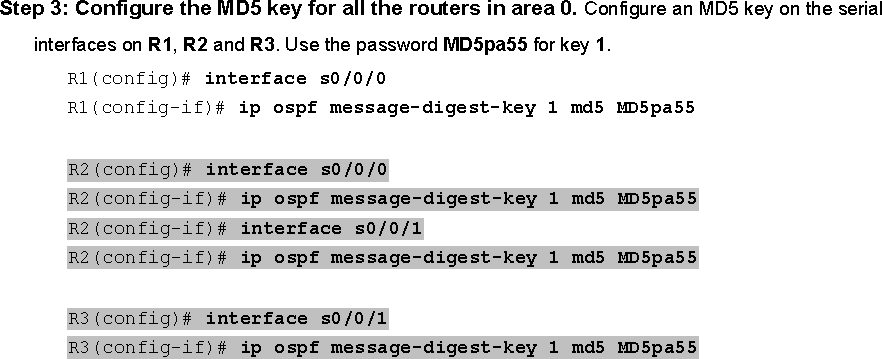
The way SSH works is by making use of a client-server model to allow for authentication of two remote systems and encryption of the data that passes between them.

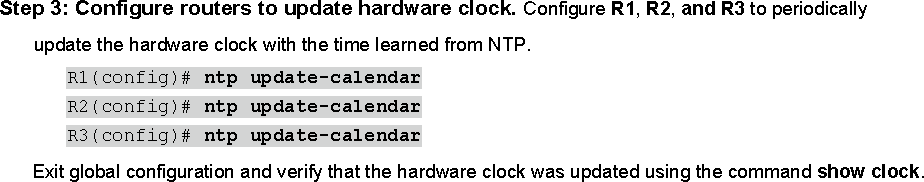
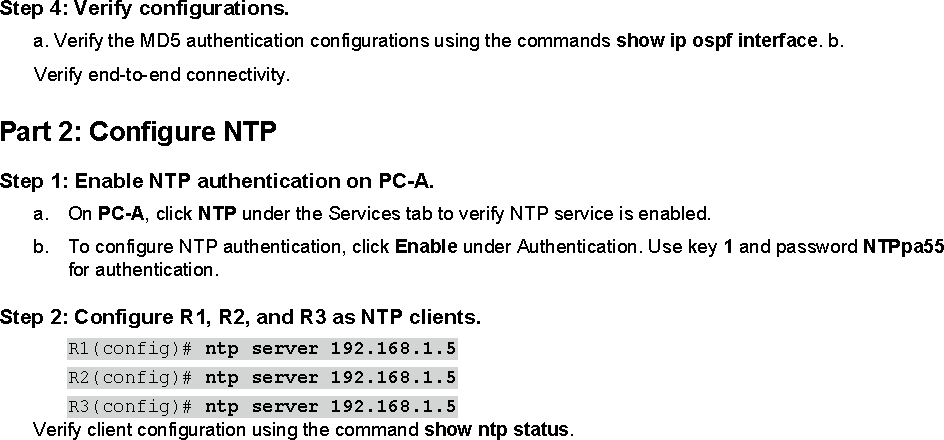
It organizes the secure connection by authenticating the client and opening the correct shell environment if the verification is successful.

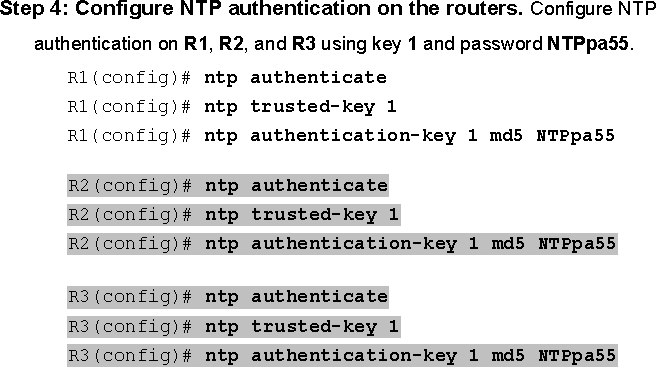


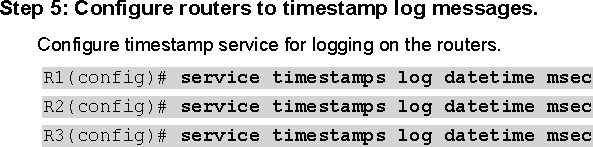


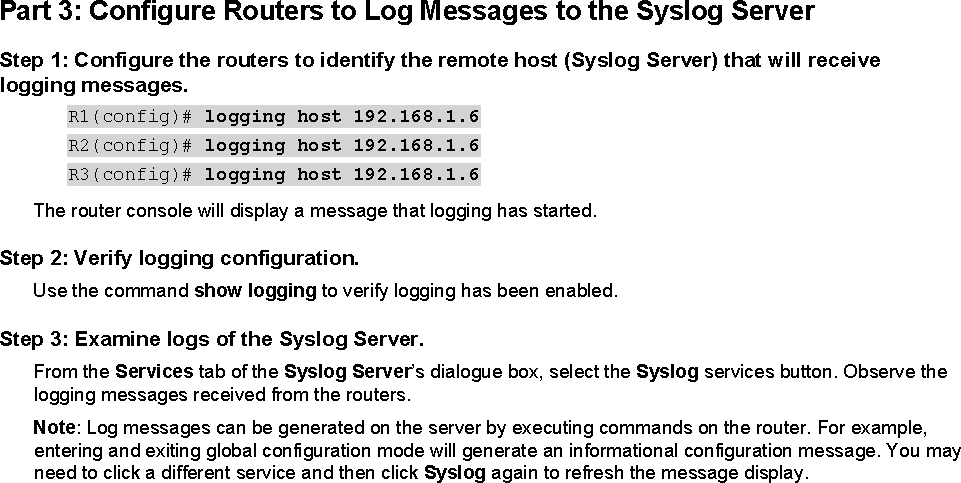


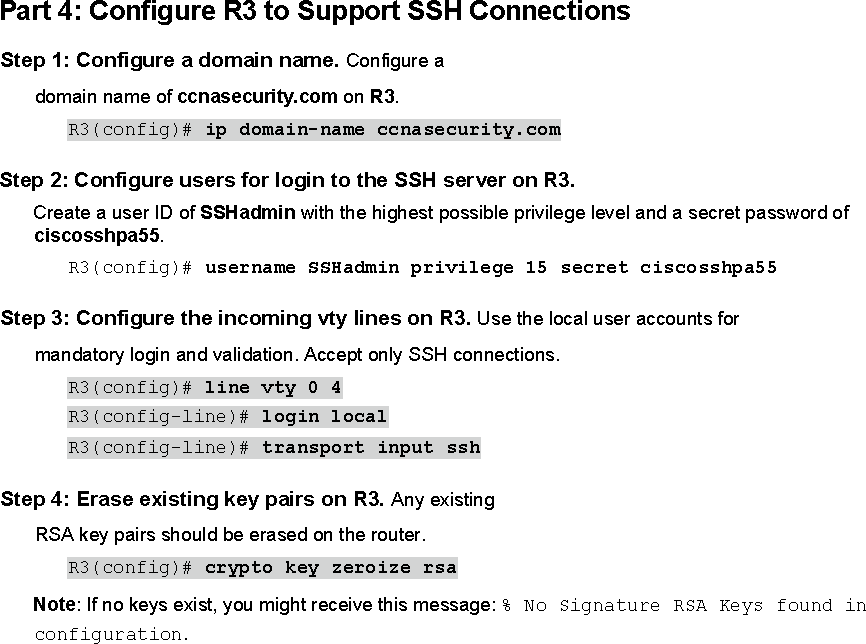


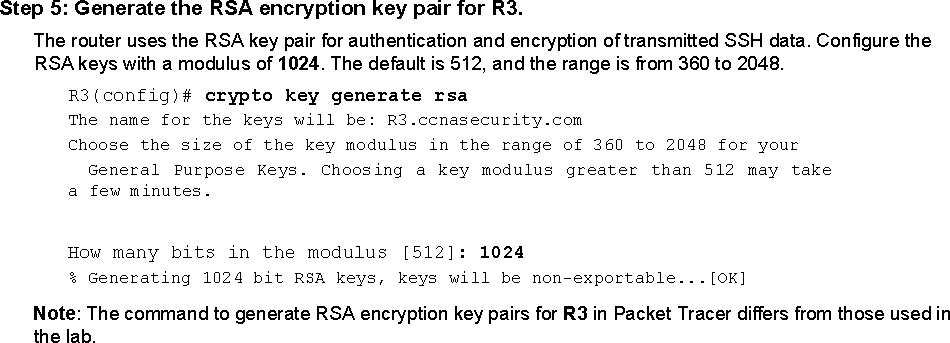


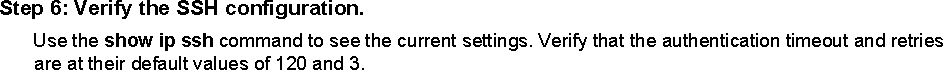


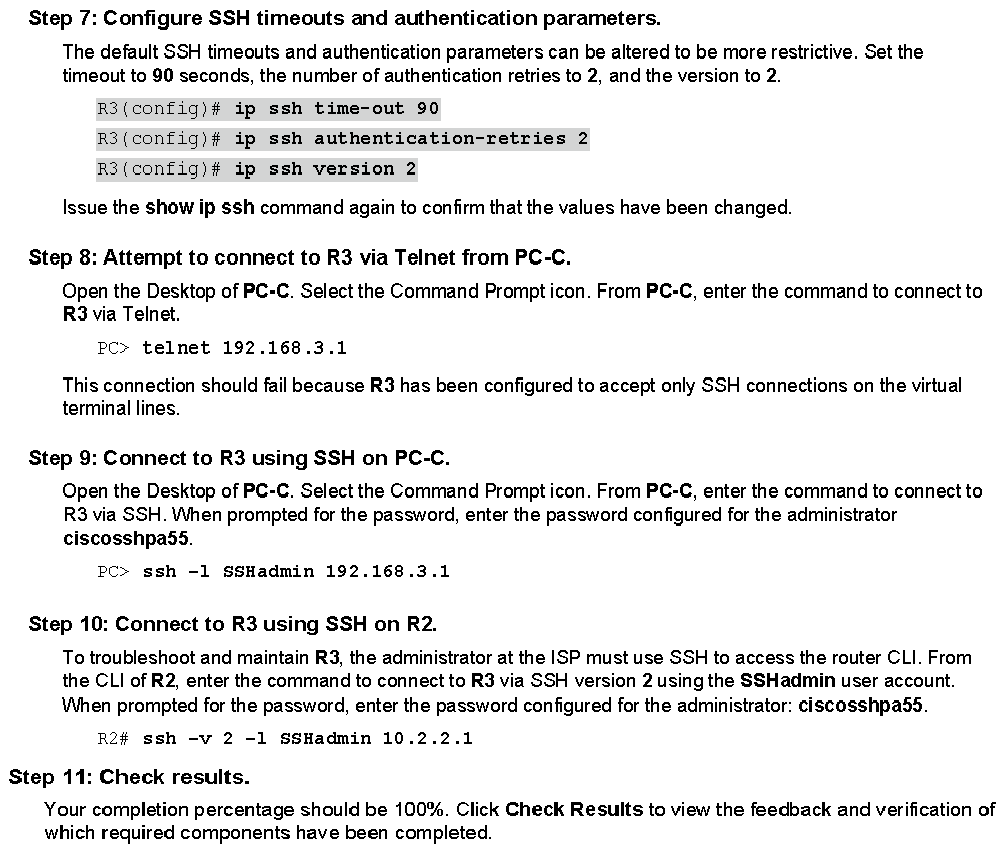












**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical 1**

**Configure Cisco Routers for Syslog, NTP, and SSH Operations**

**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.

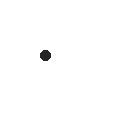


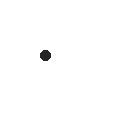
If nothing in the message changes, the hash value of the receiver should match the hash value of the sender which is transmitted with the message.

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.

### 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.

### SYSLOG server

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 tolog different types of events.

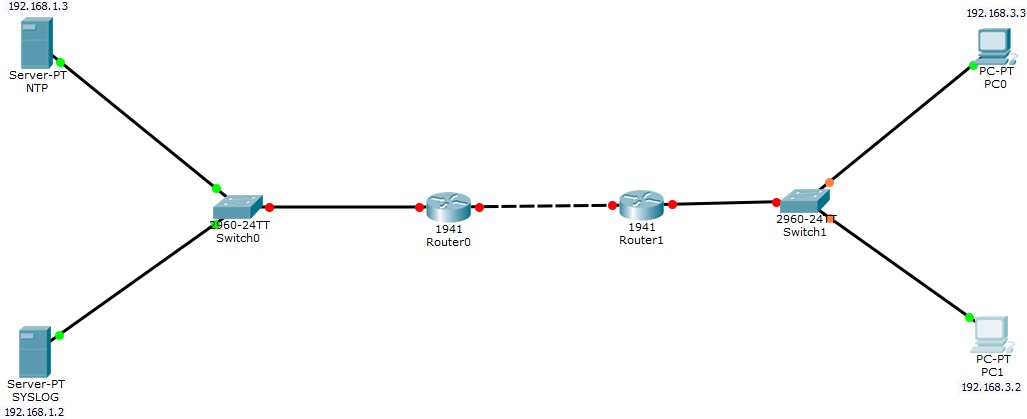
For example, a router might send messages about users logging on to console sessions, while a web-server might log access-denied events.

### SSH

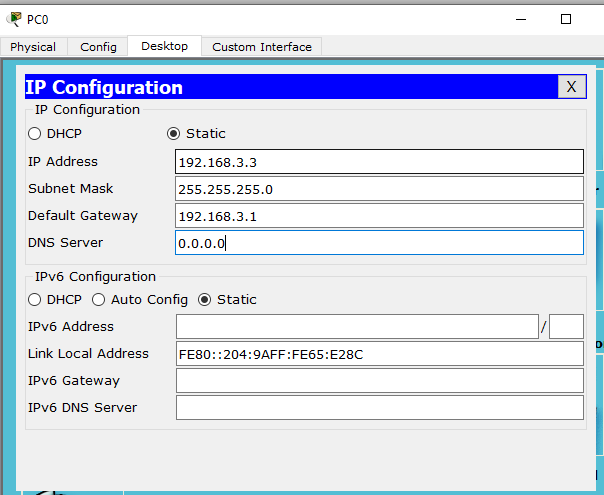
An SSH server is a software program which uses the secure shell protocol to accept connections from remote computers.

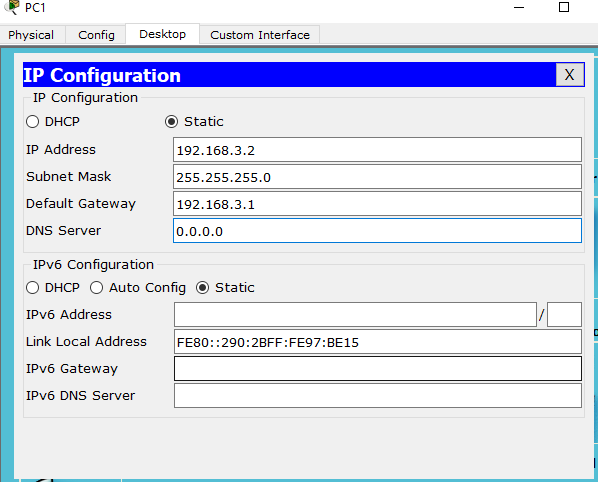
The way SSH works is by making use of a client-server model to allow for authentication of two remote systems and encryption of the data that passes between them.

It organizes the secure connection by authenticating the client and opening the correct shell environment if the verification is successful.

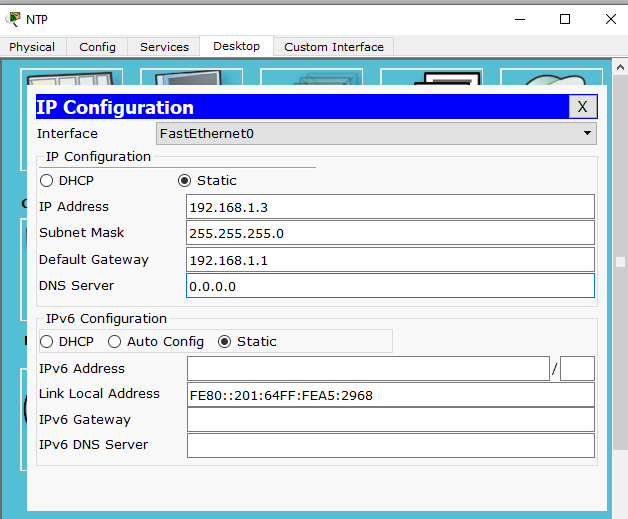


### Configuring PC0

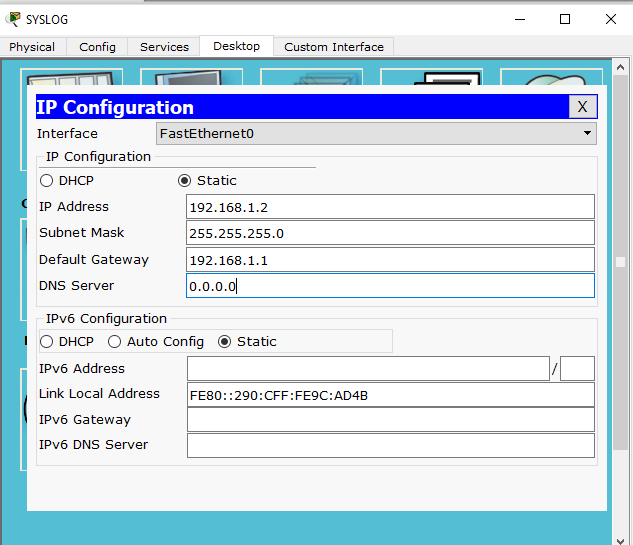


**Configuring PC1**

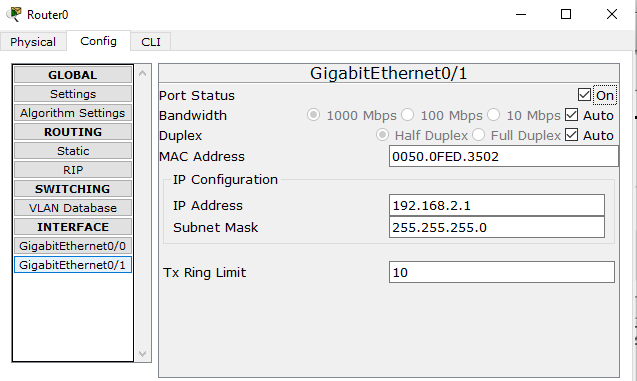
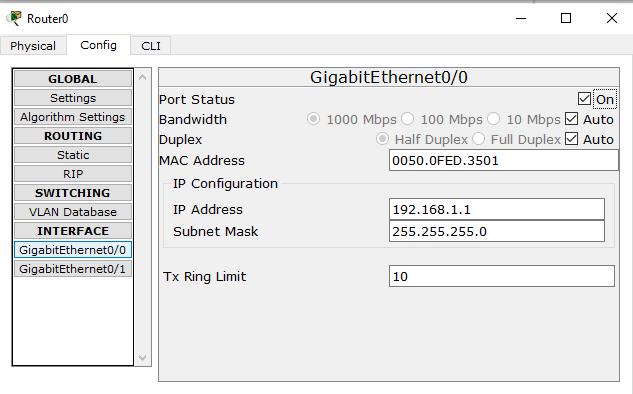
### Configuring NTP Server



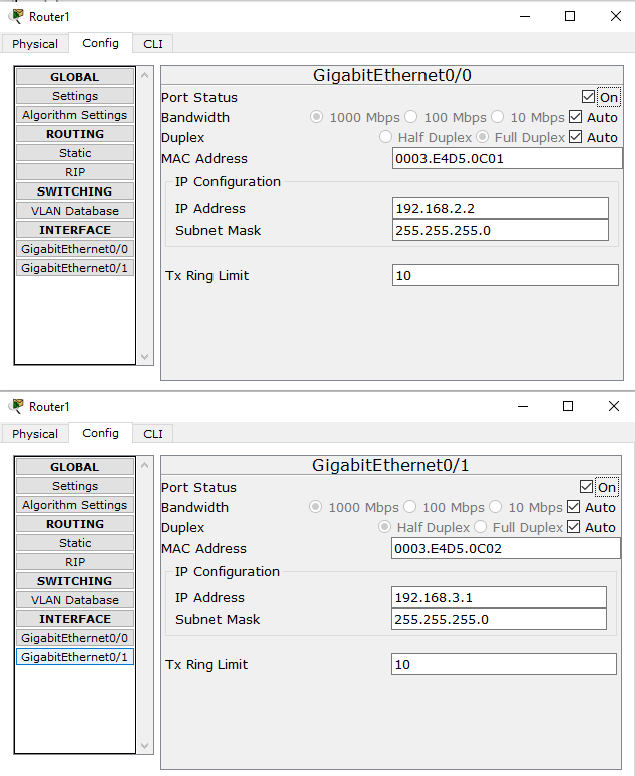
**Configuring SYSLOG Server**



### Configuring Router0



**Configuring Router1**



## Part 1: Configure OSPF MD5 Authentication

### ROUTER 0: Type the following command in the CLI mode

Router>en Router#conf t

Router(config)#router ospf 1

Router(config-router)#network 192.168.1.0 0.255.255.255 area 1

Router(config-router)#network 192.168.2.0 0.255.255.255 area 1 Router(config-router)#exit

Router(config)#exit Router#

**ROUTER1: Type the following command in the CLI mode**

Router>en Router#conf t

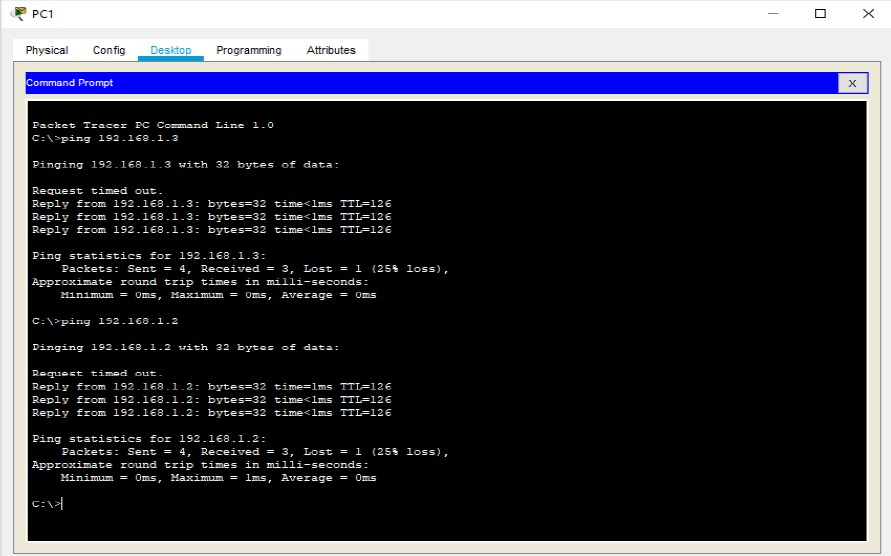
Router(config)#router ospf 1

Router(config-router)#network 192.168.3.0 0.255.255.255 area 1

Router(config-router)#network 192.168.2.0 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

**ROUTER0: Type the following command in the CLI mode**

Router>enable Router# conf t

Router(config)#int g0/0

Router(config-if)#ip ospf authentication message-digest Router(config-if)#ip ospf message-digest-key 1 md5 dalmia Router(config-if)#exit

Router(config)#int g0/1

Router(config-if)#ip ospf authentication message-digest Router(config-if)#ip ospf message-digest-key 1 md5 dalmia Router(config)#exit

### ROUTER1: Type the following command in the CLI mode

Router>enable Router# conf t

Router(config)#int g0/0

Router(config-if)#ip ospf authentication message-digest Router(config-if)#ip ospf message-digest-key 1 md5 dalmia Router(config-if)#exit

Router(config)#int g0/1

Router(config-if)#ip ospf authentication message-digest Router(config-if)#ip ospf message-digest-key 1 md5 dalmia Router(config)#exit

**Verify the MD5 Authentication using the following command in the CLI mode of Router0**

### 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

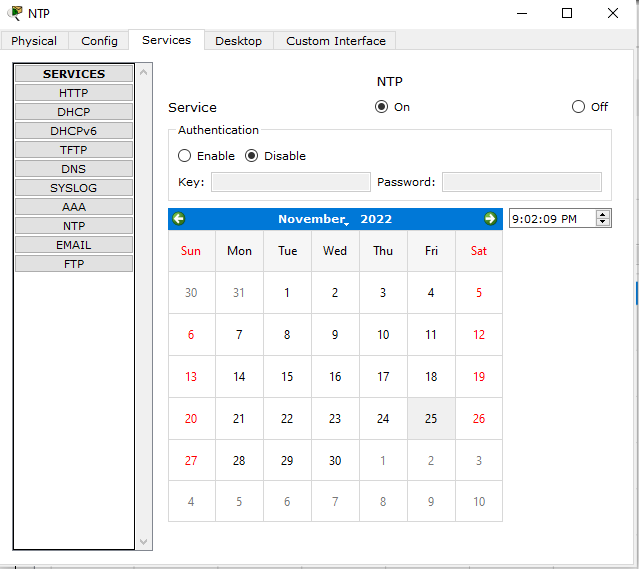
Last flood scan time is 0 msec, maximum is 0 msec Neighbor Count is 1, Adjacent neighbor count is 1 Adjacent with neighbor 192.168.3.1 (Designated Router) Suppress hello for 0 neighbor(s)

### Message digest authentication enabled

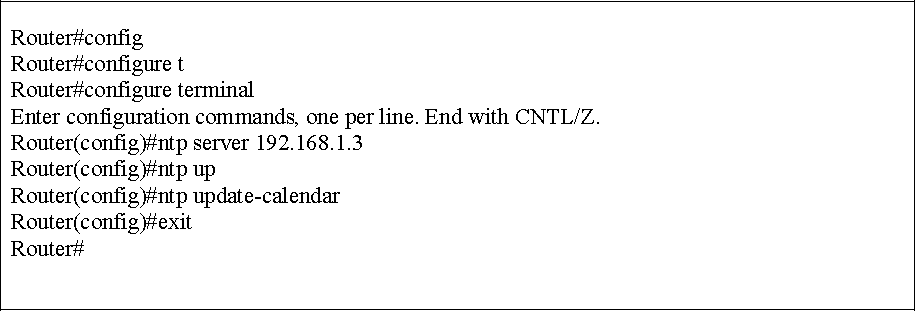
Youngest key id is 1

**MD5 Authentication has been verified**

# Part 2: Configure NTP Server and enable the NTP service



### We must disable the NTP service on other servers’ else output won’t be obtained

Now Go to CLI Mode of both the routers and type the following commands:-

### To verify the Output, we use the following command

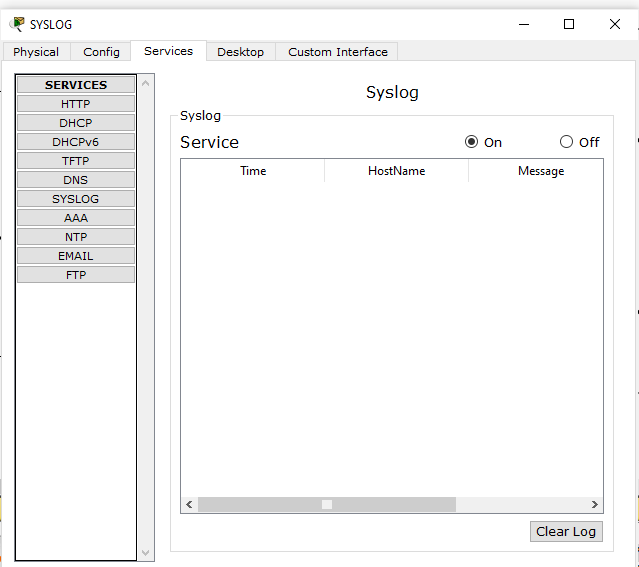
Router#show clock

\*21:7:3.987 UTC Fri Nov 25 2022

Router#

# Part 3: Configure SYSLOG Server and enable the service

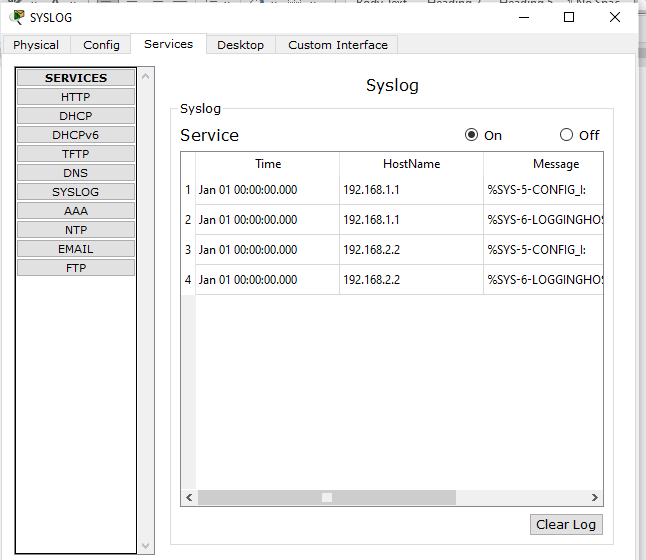
### Turn ON the SYSLOG service on the server



**And Turn OFF on all other Servers**

### Now Go to CLI Mode of both the Routers and type the following commands: -

**Output:**



# Part 4: Configure SSH on Router1

### Go to CLI Mode of Router1 and type the following commands: -

Router#conf t

Router(config)#ip domain-name dalmia.com Router(config)#hostname R1

R1(config)#

R1(config)#crypto key generate rsa

The name for the keys will be: R0.dalmia.com

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

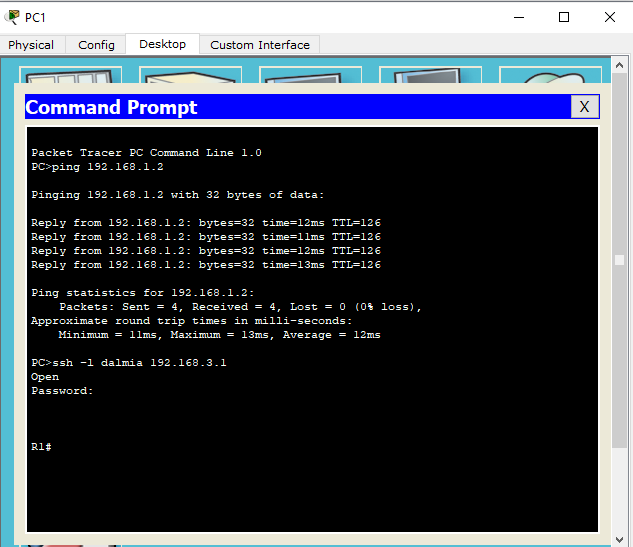
How many bits in the modulus [512]: 1024

% Generating 1024 bit RSA keys, keys will be non-exportable...[OK] R1(config)#line vty 0 4

\*Nov 25 21:19:48.169: %SSH-5-ENABLED: SSH 1.99 has been enabled R1(config-line)#transport input ssh

R1(config-line)#login local R1(config-line)#exit

R1(config)#username dalmia privilege 15 password cisco R1(config)#

**Output: Go to cmd of PC1 and type the command ssh –l dalmia 192.168.3.1 and type the password cisco**

### Hence SSH is also verified

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical 2**

**Configure AAA Authentication on Cisco Routers**

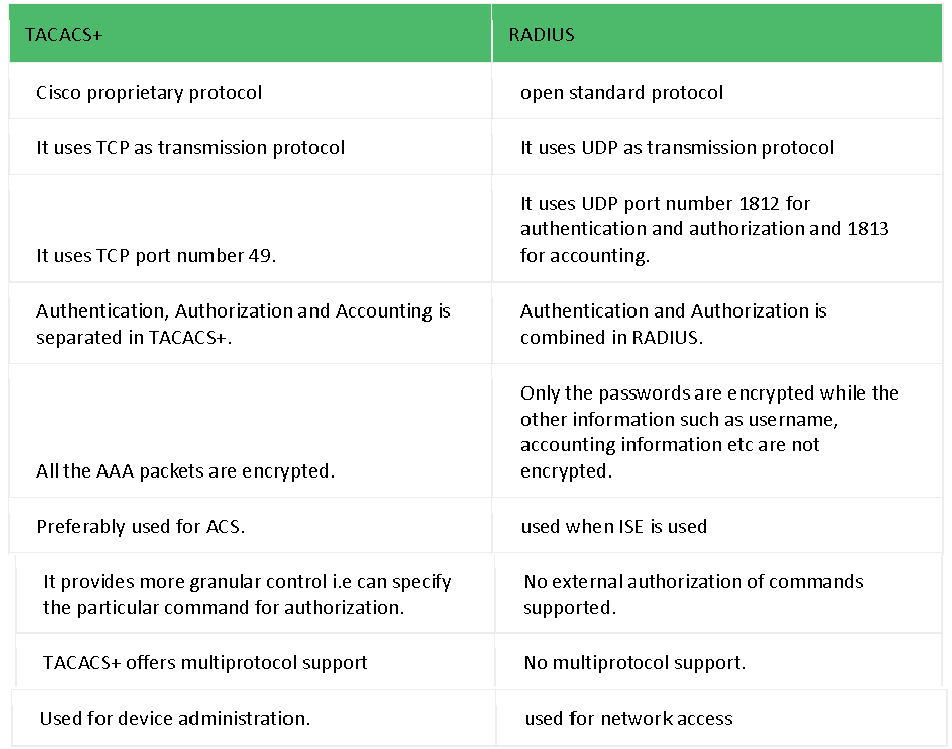
To provide a centralized management system for the authentication, authorization and accounting (AAA framework), Access Control Server (ACS) is used. For the communication between the client and the ACS server, two protocols are used namely TACACS+ and RADIUS.

### 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.



### Similarities

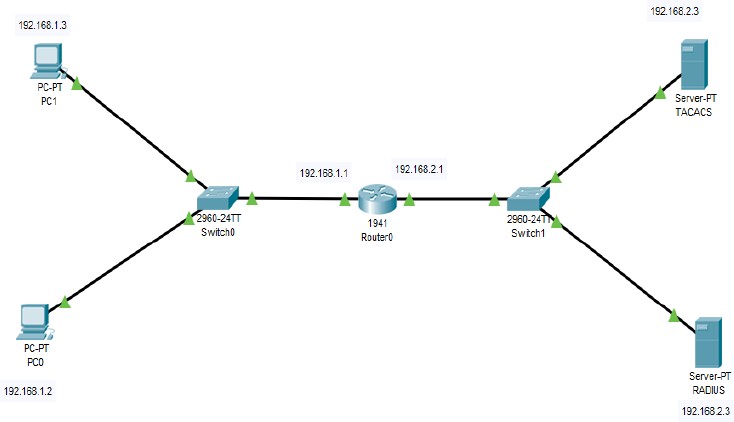
The process is start by Network Access Device (NAD – client of TACACS+ or RADIUS). NAD contact the TACACS+ or RADIUS server and transmit the request for authentication (username and password) to the server. First, NAD obtain username prompt and transmit the username to the server and then again, the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

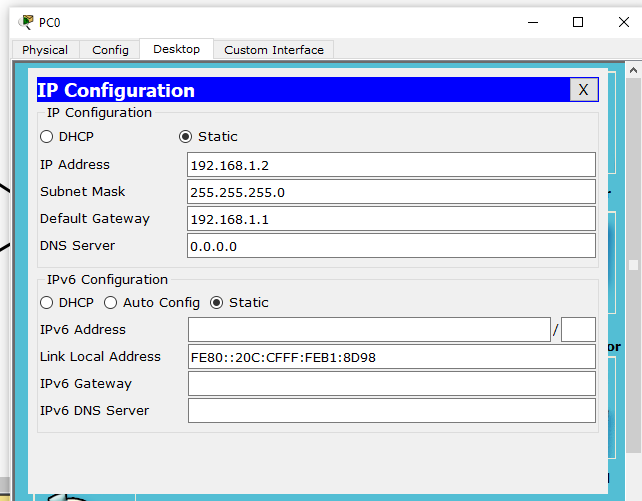
1. As TACACS+ uses TCP therefore more reliable than RADIUS.
2. TACACS+ provides more control over the authorization of commands while in RADIUS, no external authorization of commands is supported.
3. All the AAA packets are encrypted in TACACS+ while only the passwords are encrypted in RADIUS i.e more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

### Advantages (RADIUS over TACACS+)

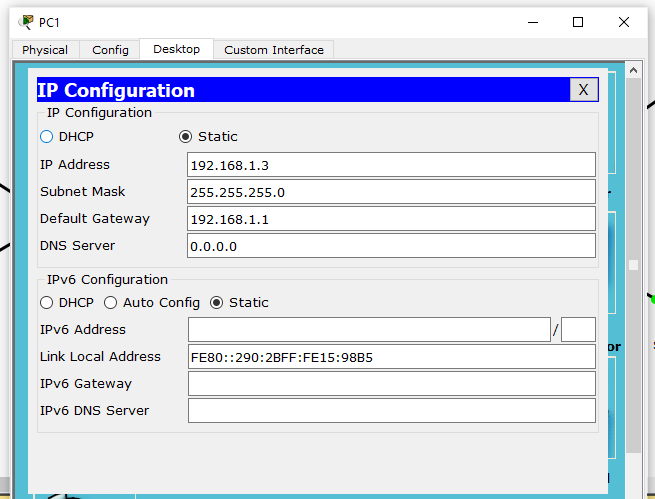
1. As it is open standard therefore RADIUS can be used with other vendors device while because TACACS+ is Cisco proprietary, it can be used with Cisco devices only.
2. It has more extensive accounting support than TACACS+.



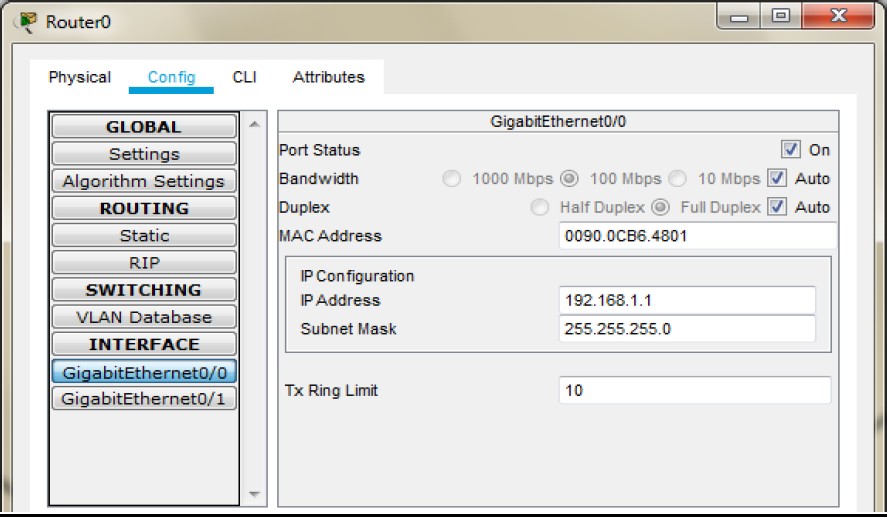
### Configuring PC0

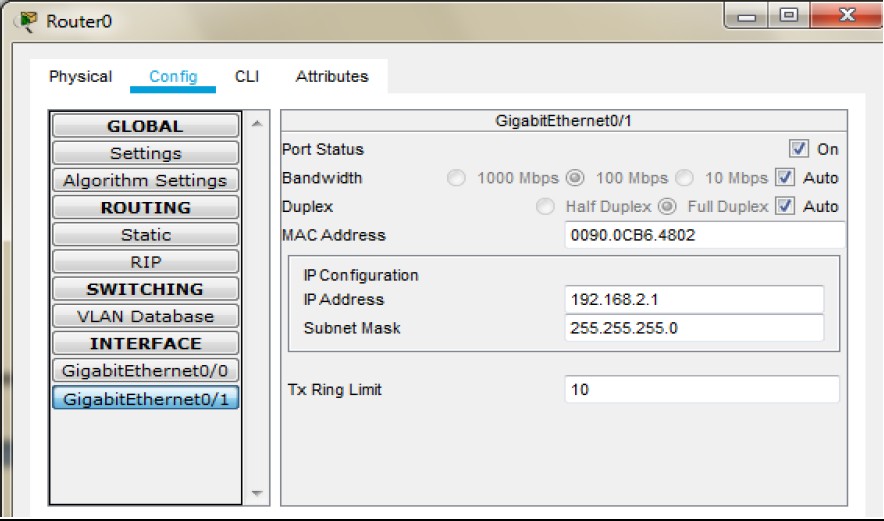


**Configuring PC1**



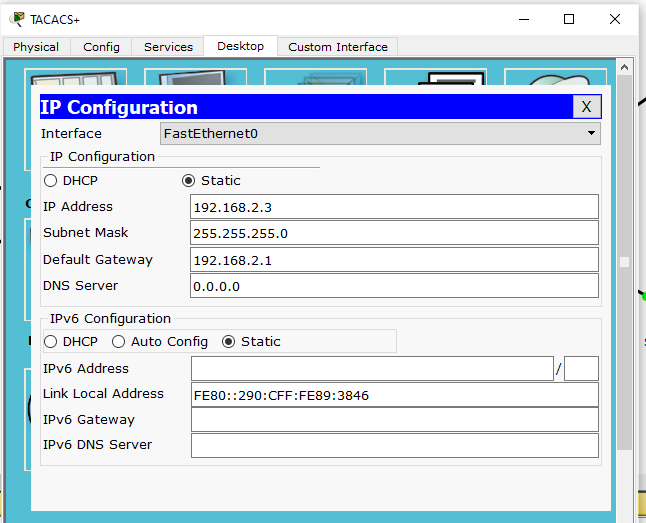
### Configuring Router0

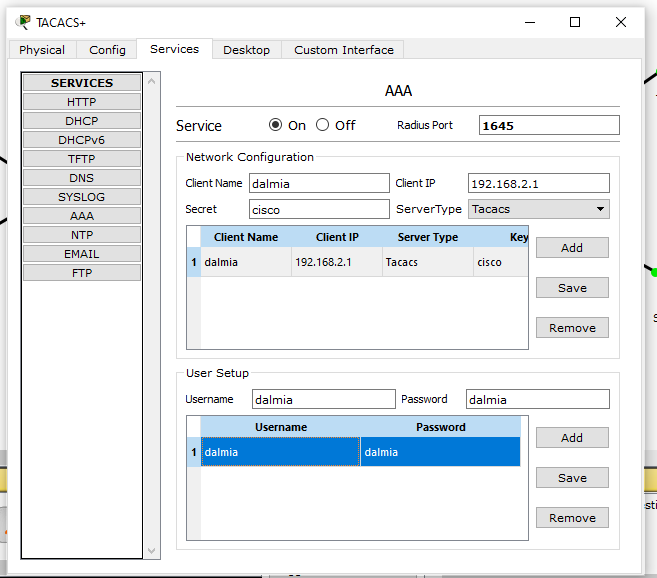




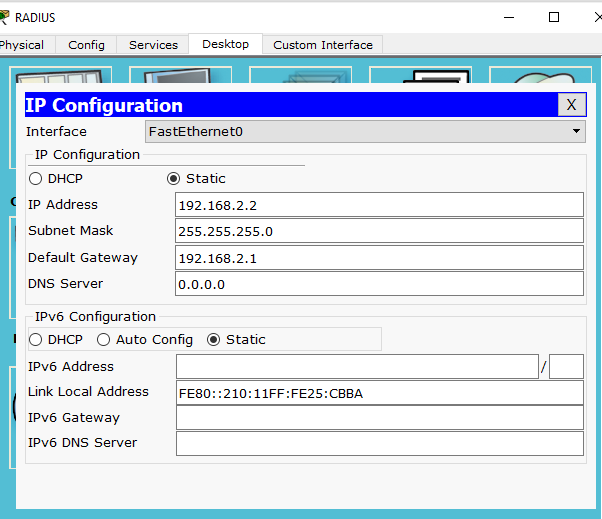
**Configuring Server0(As TACACS)**

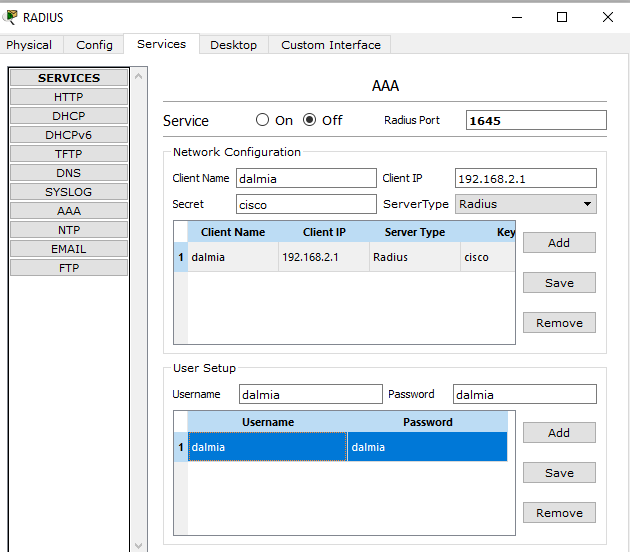
While configuring the TACACS/RADIUS server the Client IP address must be the Router IP.





### Configuring Server1(As RADIUS)





**Click on Router 0 go to CLI tab and press enter and enter the following commands: -**

Router>en Router#conf t

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 dalmia group tacacs+ group radius local Router(config)#line vty 0 4

Router(config-line)#login authentication dalmia Router(config-line)#exit

Router(config)#

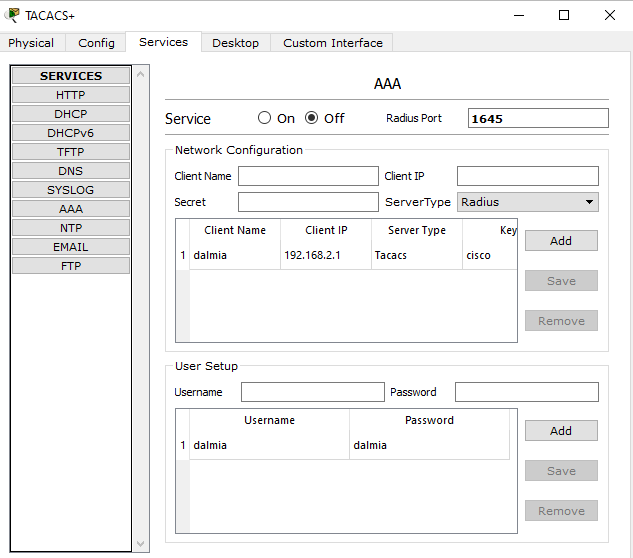
The Authentication can be done by typing the command **telnet 192.168.2.1** (the Router IP) in any of the PCs

We get a prompt to type the username and password, the username and password set in TACACS are entered

Username: dalmia Password: dalmia We get the following

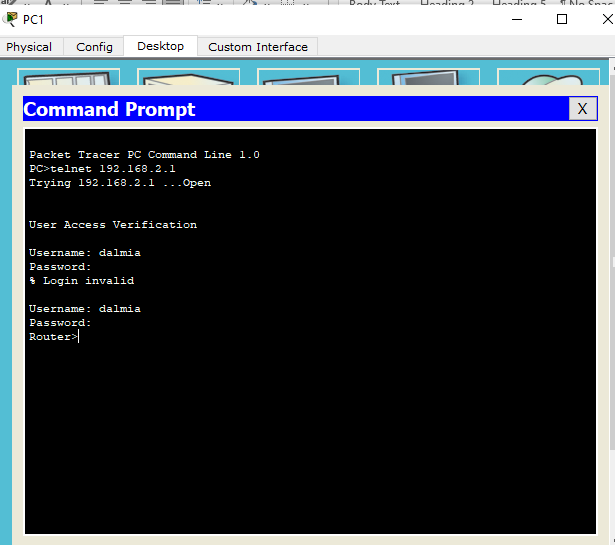


In order to authenticate the RADIUS server, we need to turn OFF the TACACS service



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

We get the following



The local login can also be verified by turning OFF both TACACS and RADIUS service. Hence the authentication through both TACACS and RADIUS.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical 3**

**Configuring Extended ACLs: Configure, Apply and Verify an Extended Numbered ACL**

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: Standard Access Lists, and



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**

# 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:

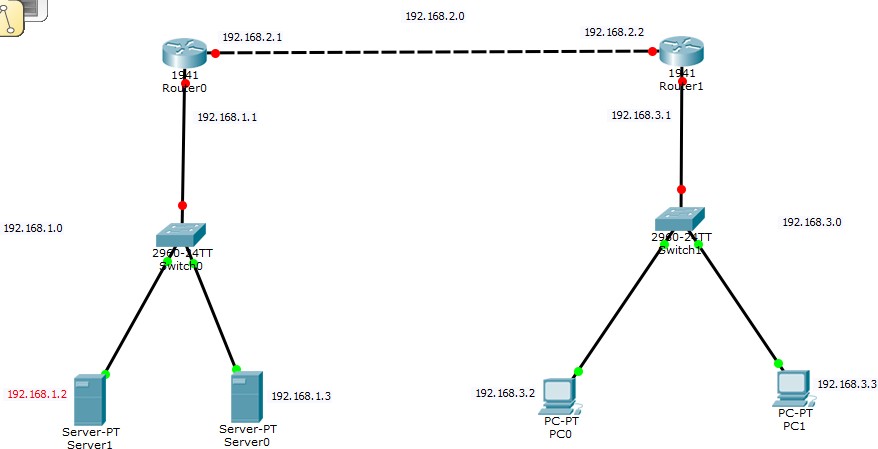
interface <interface> ip access-group {number|name} {in|out}

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:

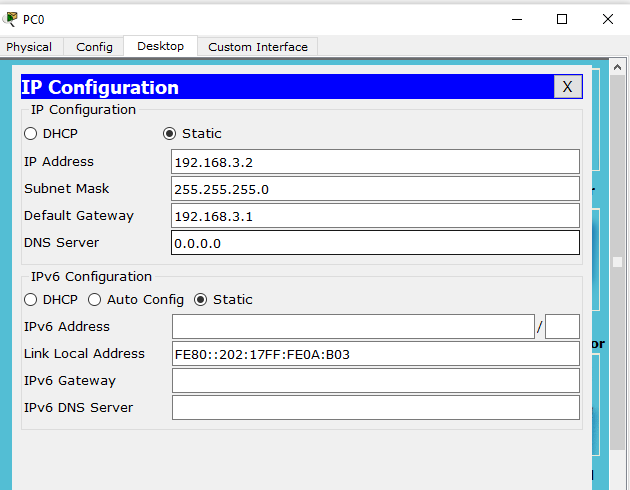
**Rouer(config)#interface serial0 Rouer(config-if)#ip access-group 10 out**

**Consider the following topology**

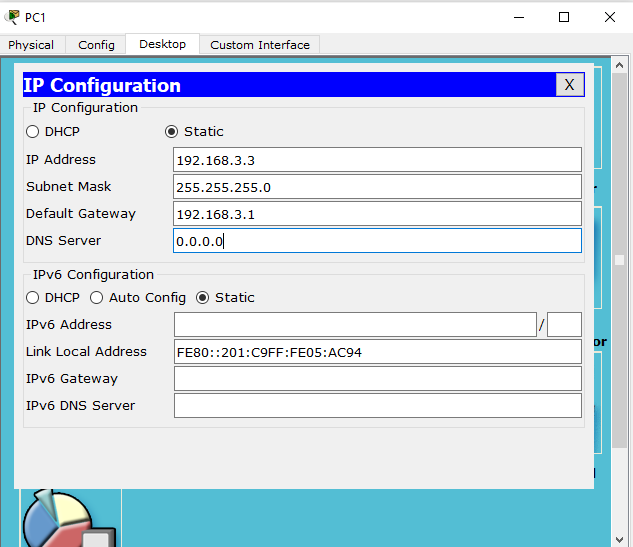


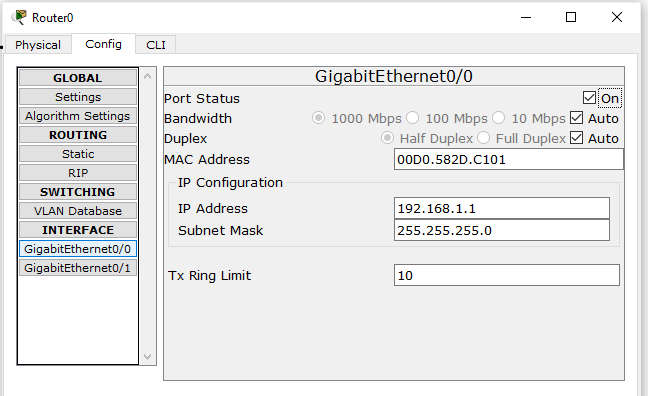
**Topology Configuration**

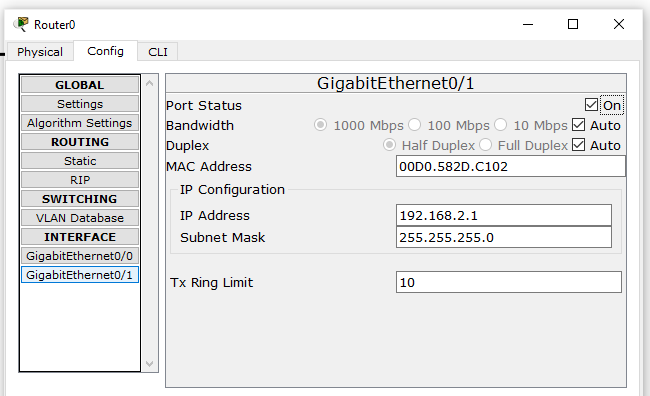
### Configuring PC0



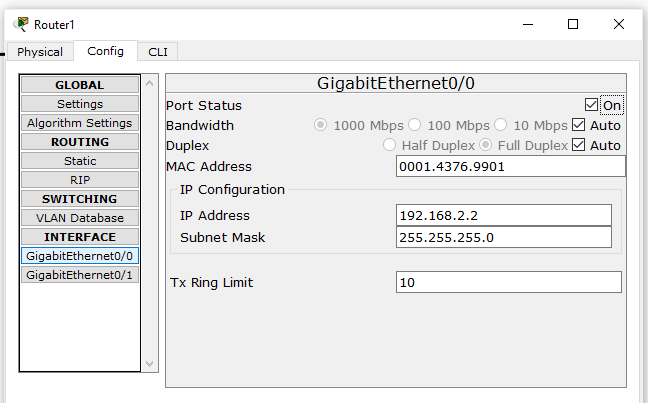
**Configuring PC1**

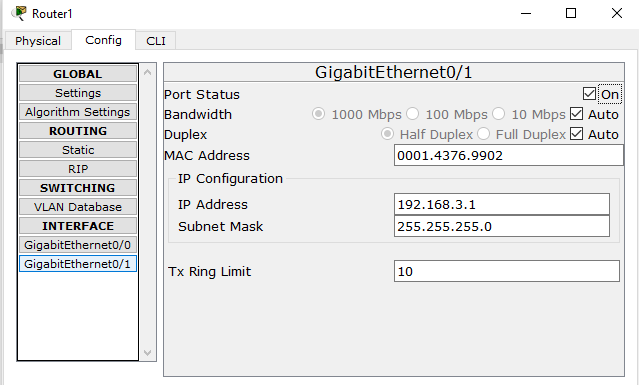




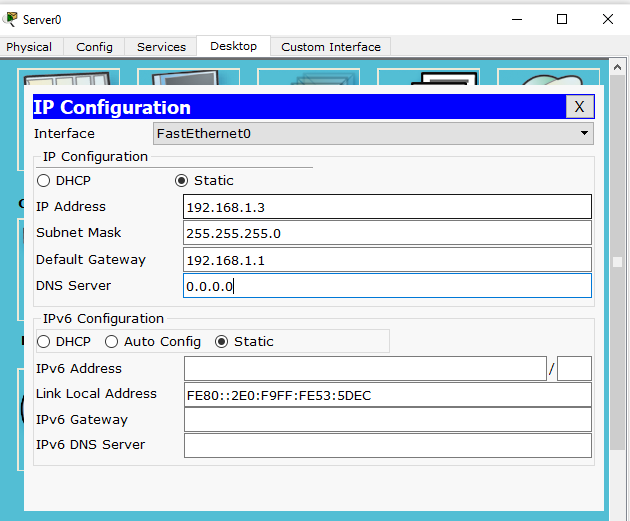


### Configuring Router1

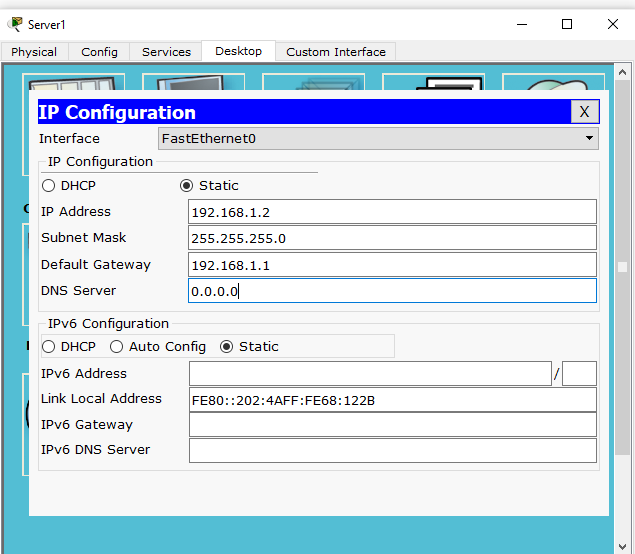




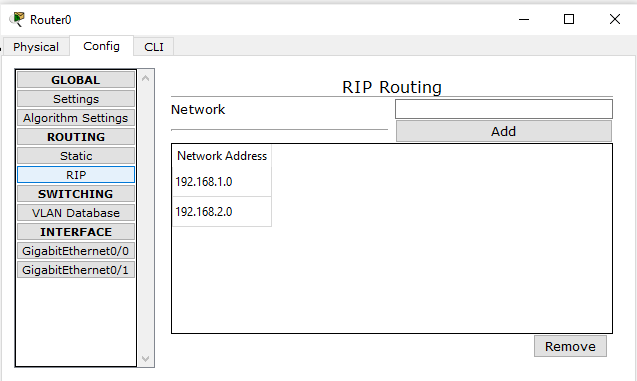
**Configuring Server0**

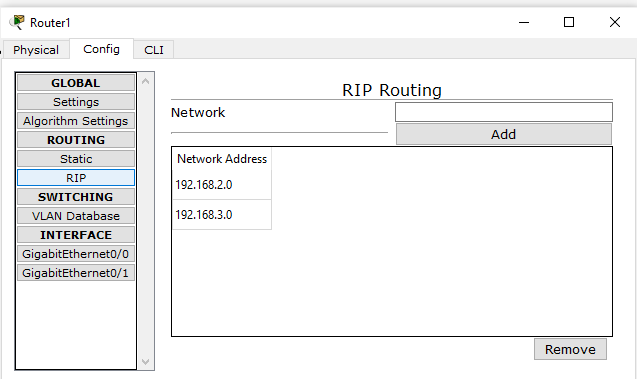


### Configuring Server1



**Set the RIP protocol on both the Routers as follows**





**Check the connectivity by using the ping command**

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

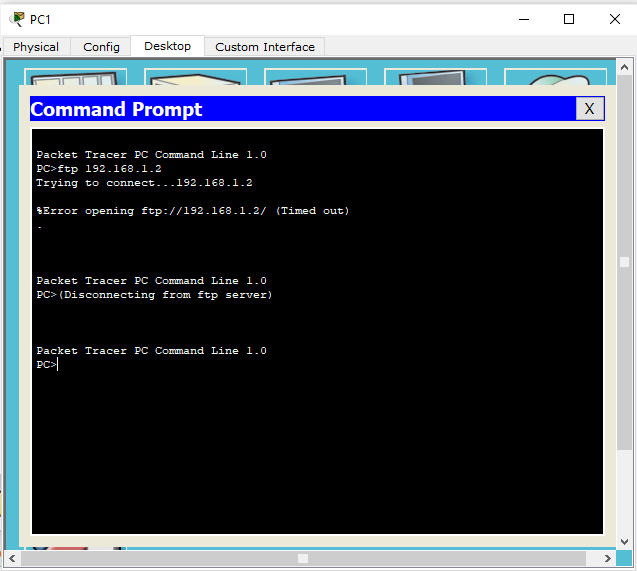
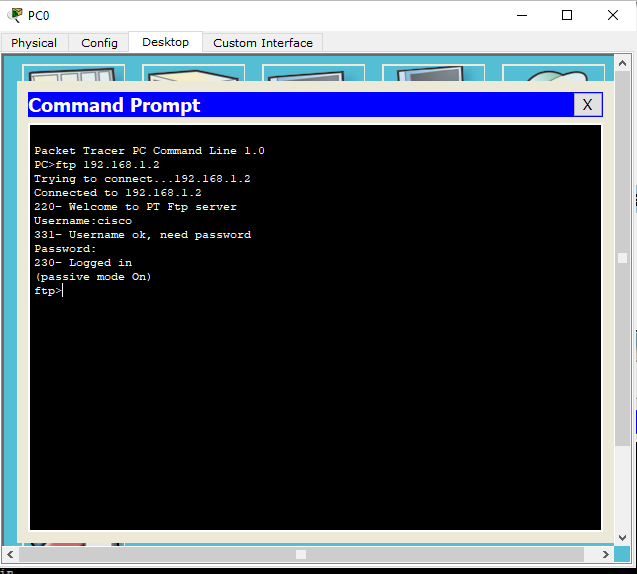
### Click on Router1 go to CLI tab and press enter and enter the following commands: -

Router>en Router#conf t

Router(config)# access-list 100 permit tcp host 192.168.3.2 host 192.168.1.2 eq ftp Router(config)# interface GigabitEthernet0/0

Router(config)# ip access-group 100 out Router(config-line)#exit Router(config)#

**Now verify the ftp (ftp 192.168.1.2) 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

### Click on Router1 go to CLI tab and press enter and enter the following commands: -

Router>en Router#conf t

Router(config)# ip access-list extended DALMIA

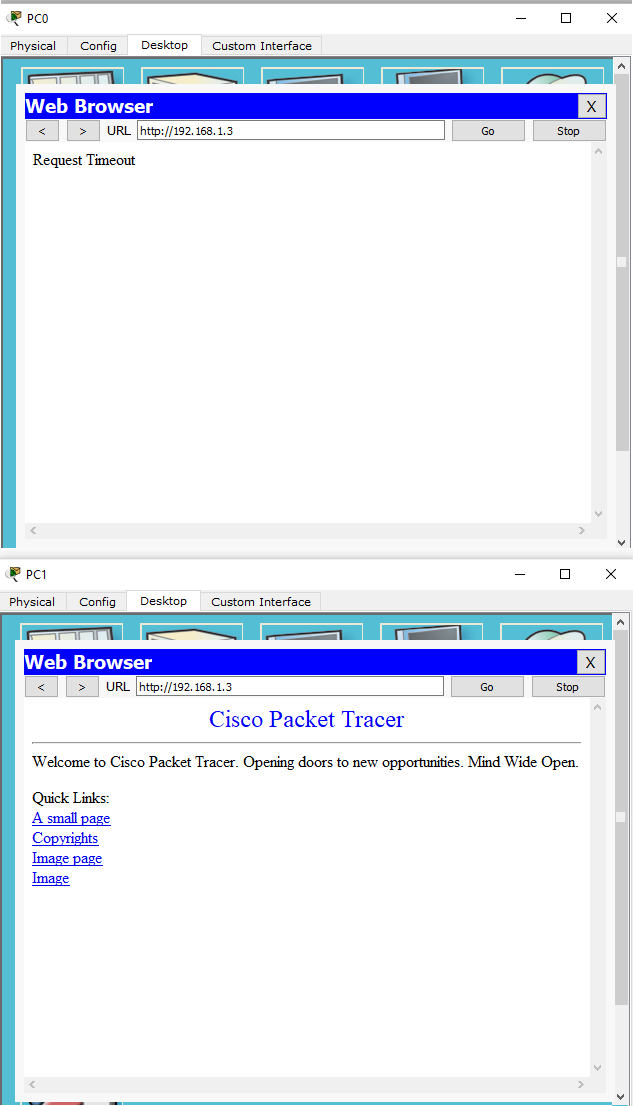
Router(config-ext-nacl)# permit tcp host 192.168.3.3 host 192.168.1.3 eq www Router(config-ext-nacl)#exit

Router(config)#

Router(config)#interface GigabitEthernet0/0 Router(config-if)#ip access-group DALMIA 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 4**

**Configure IP ACLs to Mitigate Attacks**

### 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:
5. Per-protocol (IP, IPX, AppleTalk)
6. Per direction (in or out)
7. Per port (interface) basis.
8. ACLs control traffic in one direction at a time on an interface.
9. A separate ACL would need to be created for each direction, one for inbound and one for outbound traffic.
10. 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

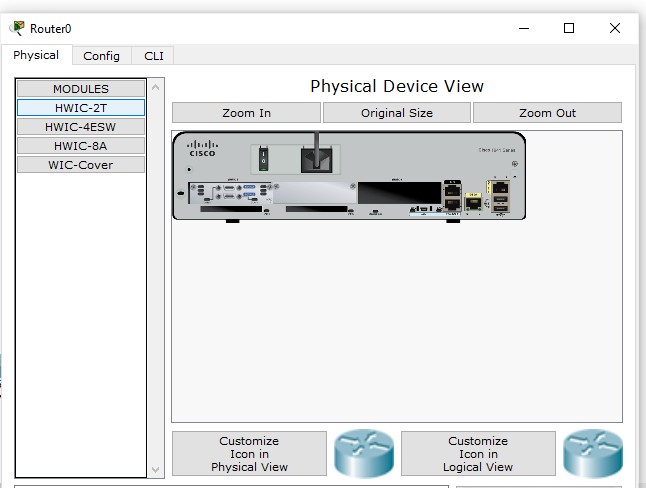
An access list is a sequential series of commands or filters. These lists tell the router what types of packets to: accept or deny Acceptance and denial can be based on specified conditions. ACLs applied on the router's interfaces.

## Consider the following topology

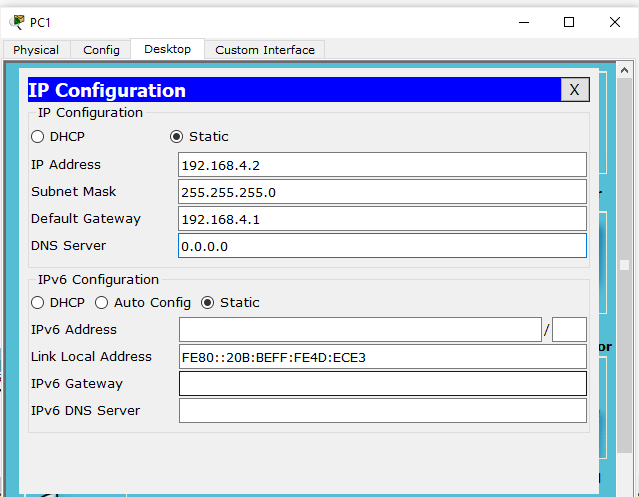


**Topology Configuration**

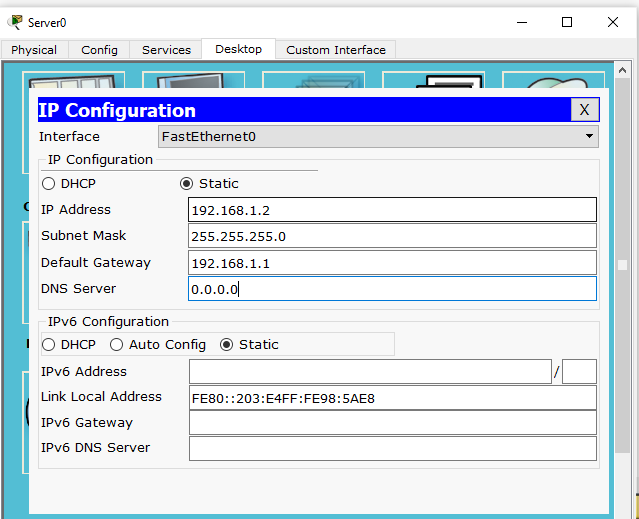
### The serial interface in each Router is added as follows

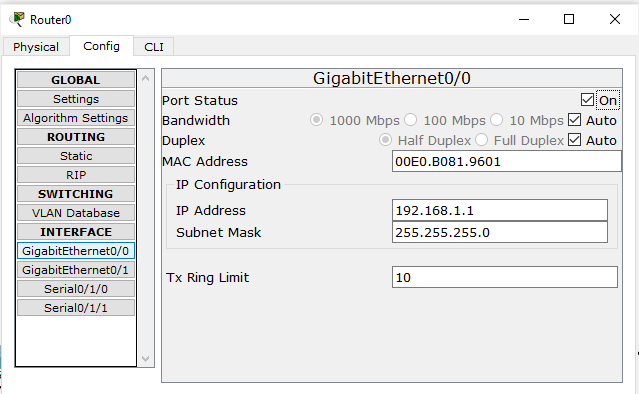


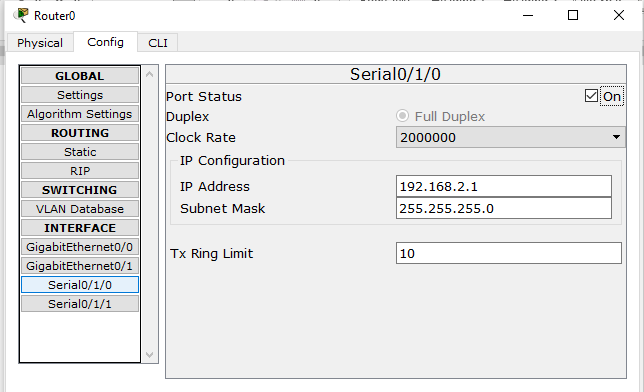
**Configuring PC1**

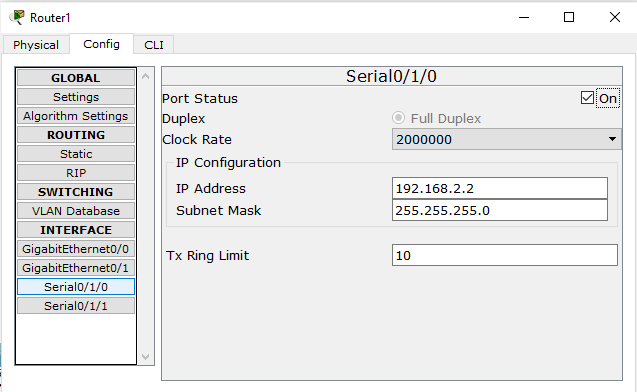


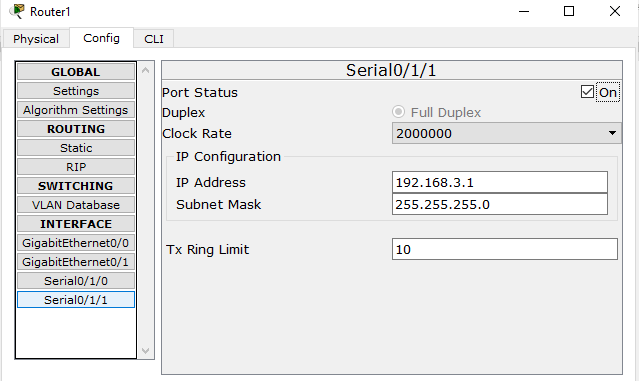
### Configuring Server0



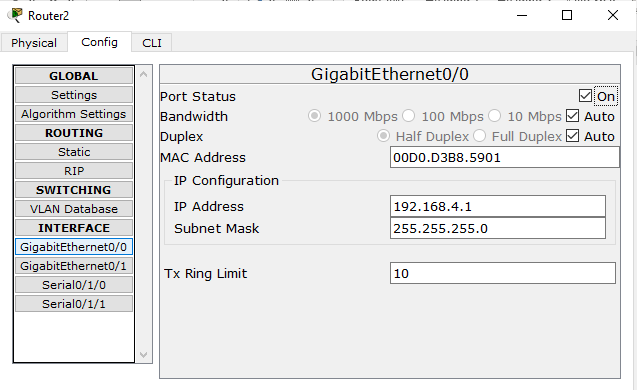
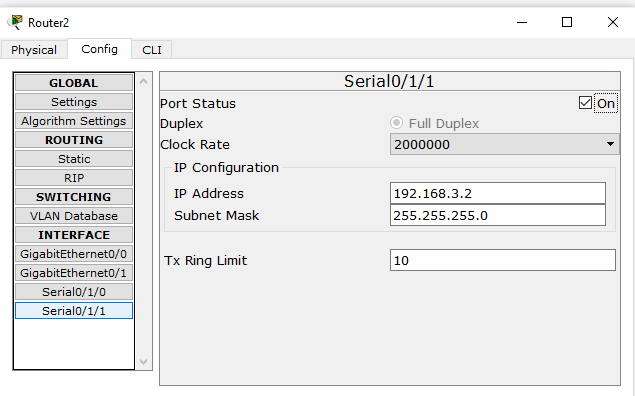




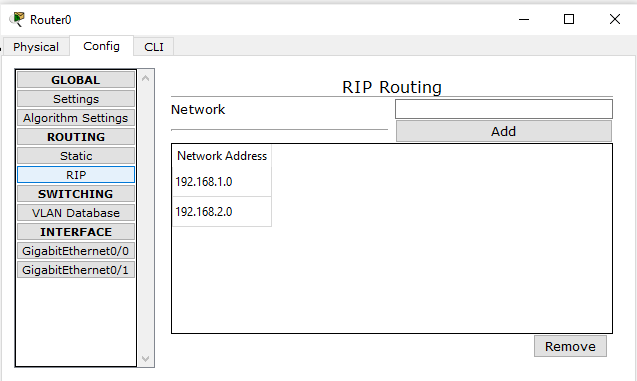


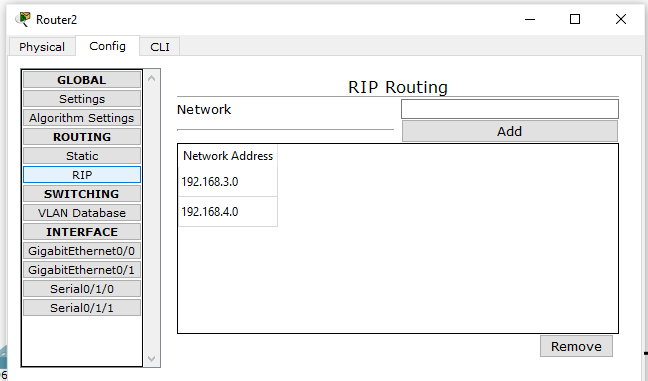
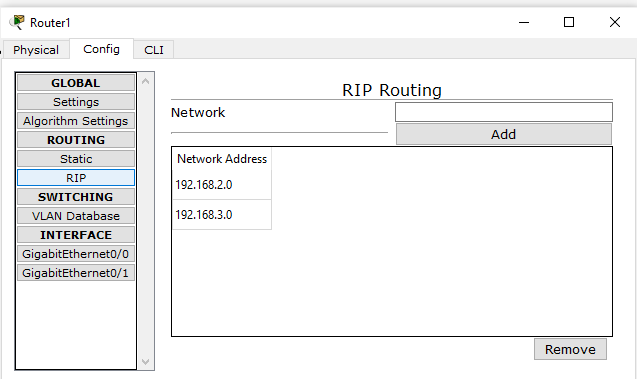


**Configuring Router2**



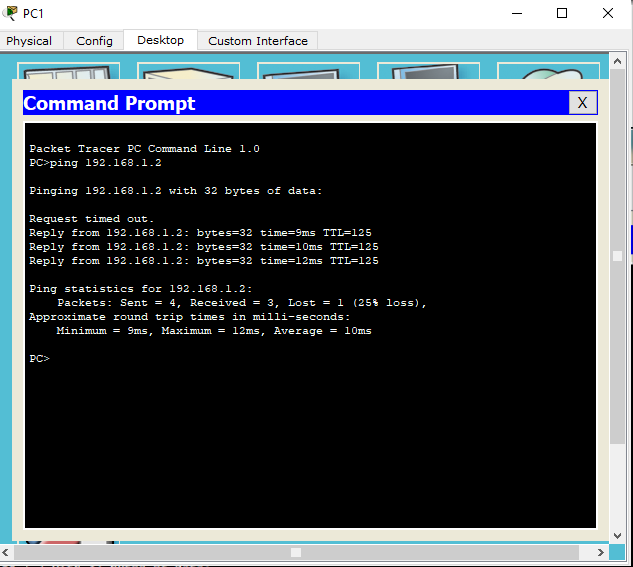
**Set the RIP protocol on all the Routers as follows**





## Part 1: Verify Basic Connectivity

### 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.

### Part a) Set up the SSH protocol

**Enter the following commands in CLI mode of all Routers.**

Router>en Router#conf t

Router(config)# ip domain-name dalmia.com Router(config)# hostname R0

R0(config)# crypto key generate rsa R0(config)# line vty 0 4

R0(config-line)# transport input ssh R0(config-line)# login local R0(config-line)# exit

R0(config)# username SSHadmin privilege 15 password dalmia R0(config)#exit

### Part b) Create an ACL 10 to permit remote access to PC only

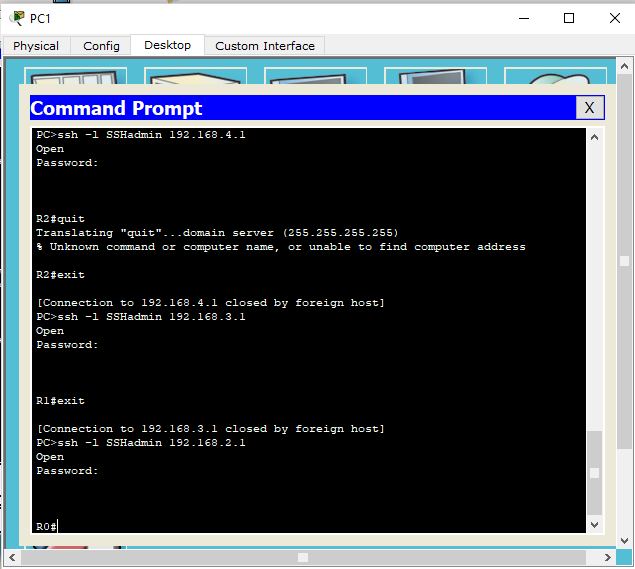
Enter the following commands in CLI mode of all Routers

Router>en Router#conf t

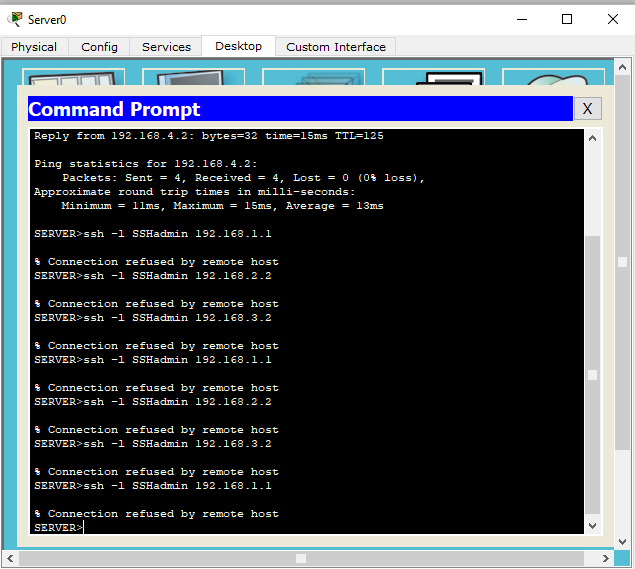
Router(config)# access-list 10 permit host 192.168.4.2

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**



## 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 **PC1 to** access **R1** via SSH. (done in previous part)

### Enter the following commands in the CLI mode of Router1

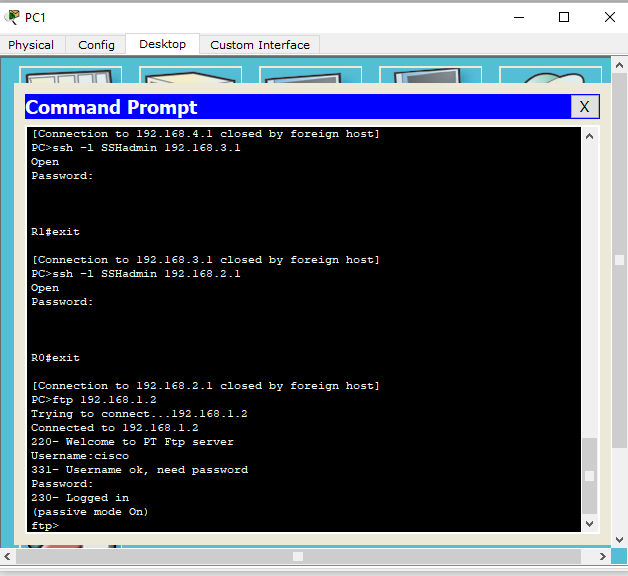
R1>enable R1#

R1#configure terminal

R1(config)#access-list 120 permit udp any host 192.168.1.2 eq domain R1(config)#access-list 120 permit tcp any host 192.168.1.2 eq smtp R1(config)#access-list 120 permit tcp any host 192.168.1.2 eq ftp R1(config)#access-list 120 deny tcp any host 192.168.1.2 eq 443 R1(config)#exit

R1#configure terminal R1(config)#interface Serial0/1/1 R1(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.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical 5**

**Configure IPv6 ACLs**

### Access Control Lists (ACLs) 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

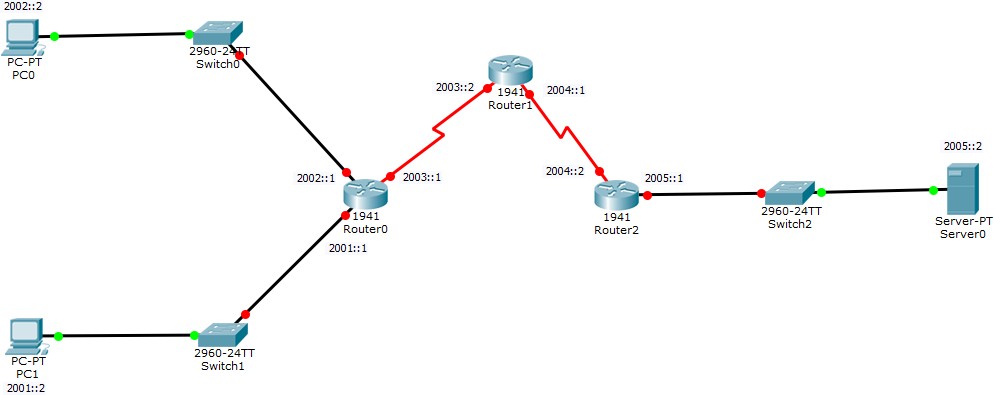
The following header fields are used for IPv6 inspection: traffic class, flow label, payload length, next header, hop limit, and source or destination IP address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

### 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**

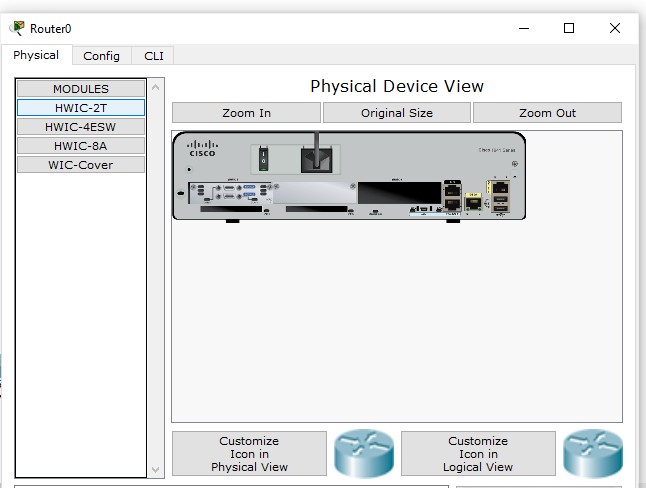
**access-class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against 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.

## Consider the following topology

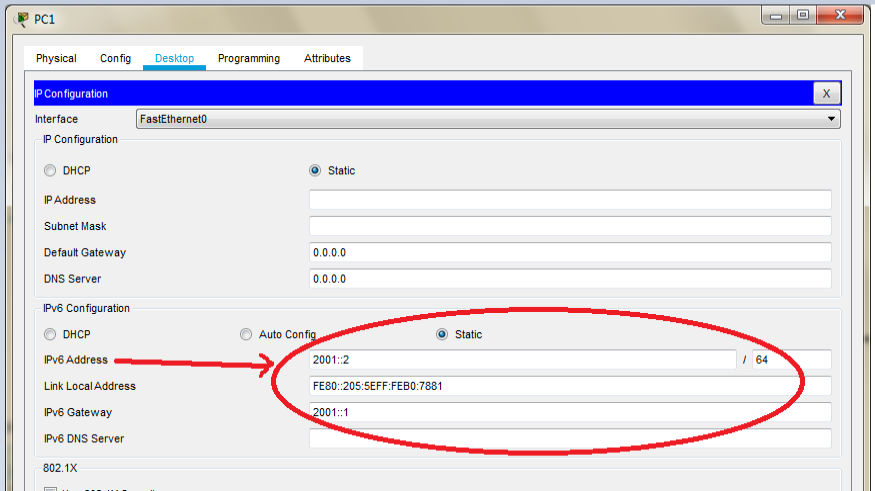


**Topology Configuration**

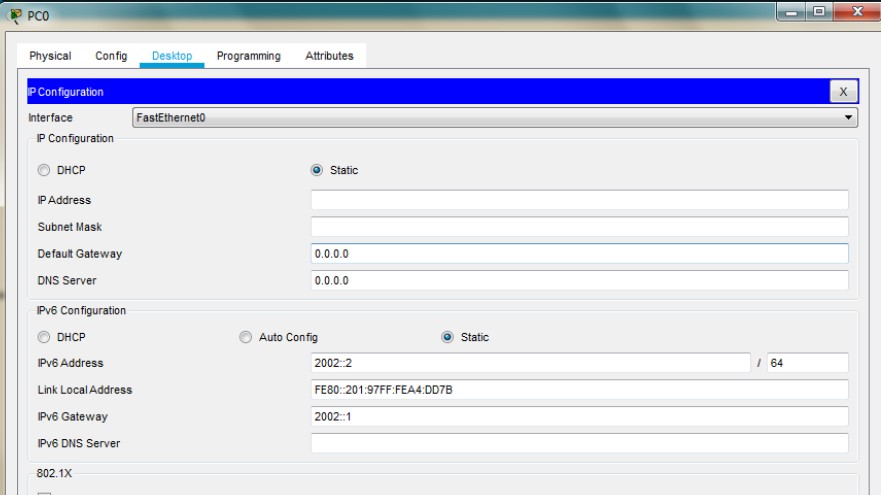
### The serial interface in each Router is added as follows



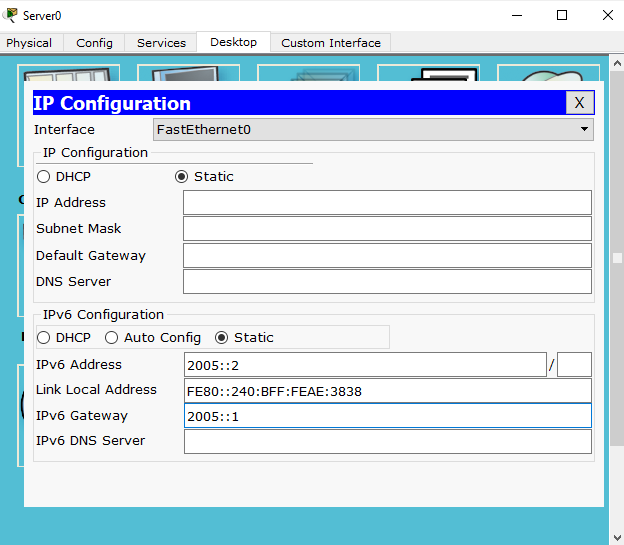
**Configuring PC1**



### Configuring PC0



**Configuring Server0**



64

### For setting the ipv6 addresses we need to use the CLI mode for each Router as follows Configuring Router0

Router> Router>en Router# Router#conf t

Router(config)#ipv6 unicast-routing Router(config)#int G0/0

Router(config-if)#ipv6 address 2002::1/64 Router(config-if)#ipv6 rip a enable Router(config-if)#no shut

Router(config-if)#exit Router(config)# Router(config)#int G0/1

Router(config-if)#ipv6 address 2001::1/64 Router(config-if)#ipv6 rip a enable Router(config-if)#no shut

Router(config-if)#exit Router(config)# Router(config)#int Se0/1/0

Router(config-if)#ipv6 address 2003::1/64 Router(config-if)#ipv6 rip a enable Router(config-if)#no shut

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

**Configuring Router1**

Router>en Router#conf t

Router(config)#ipv6 unicast-routing Router(config)#

Router(config)#int Se0/1/0

Router(config-if)#ipv6 address 2003::2/64 Router(config-if)#ipv6 rip a enable Router(config-if)#no shut

Router(config-if)# Router(config-if)#exit Router(config)# Router(config)#int Se0/1/1

Router(config-if)#ipv6 address 2004::1/64 Router(config-if)#ipv6 rip a enable Router(config-if)#no shut

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

**Configuring Router2**

Router>en Router#conf t

Router(config)#ipv6 unicast-routing Router(config)#

Router(config)#int Se0/1/1

Router(config-if)#ipv6 address 2004::2/64 Router(config-if)#ipv6 rip a enable Router(config-if)#no shut

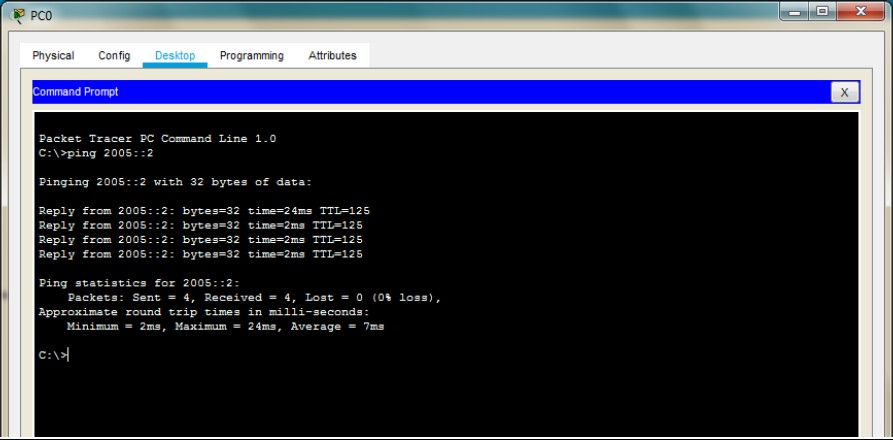
Router(config-if)#exit Router(config)#int G0/0

Router(config-if)#ipv6 address 2005::1/64 Router(config-if)#ipv6 rip a enable Router(config-if)#no shut

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

**Verify Connectivity**

### We can now verify the connectivity by pinging Server from PCs





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 apply it at the proper interface

Router> Router>enable Router#conf t

Router(config)#ipv6 access-list dalmia

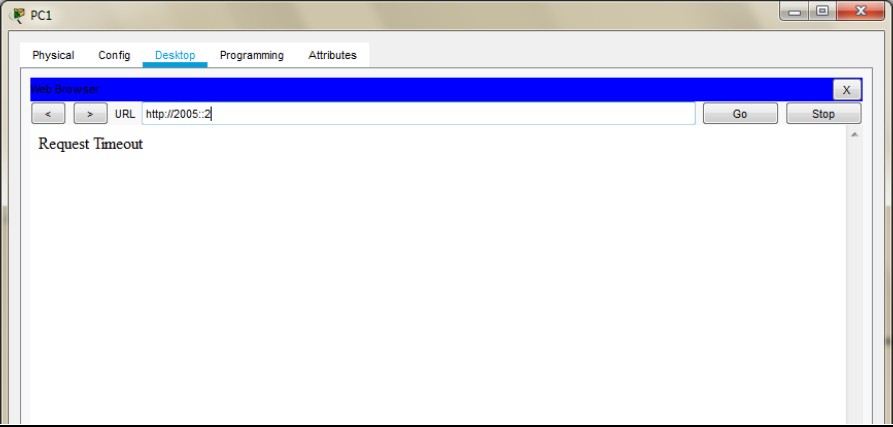
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq www Router(config-ipv6-acl)#deny tcp any host 2005::2 eq 443 Router(config-ipv6-acl)#permit ipv6 any any

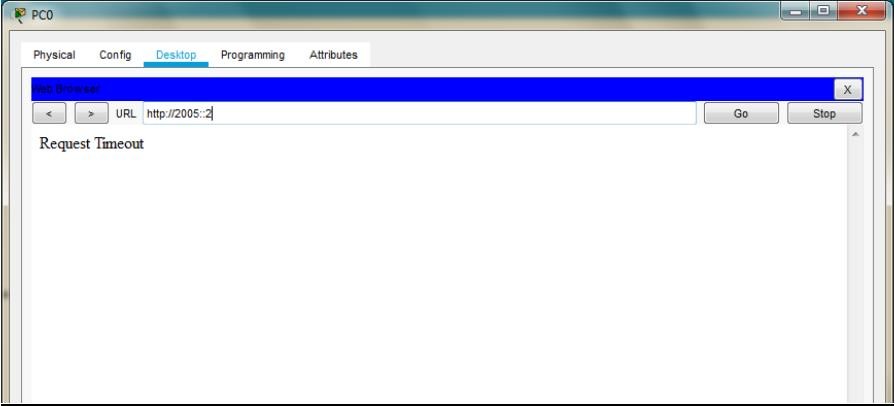
Router(config-ipv6-acl)# Router(config-ipv6-acl)#exit Router(config)# Router(config)#int Se0/1/0

Router(config-if)#ipv6 traffic-filter dalmia in Router(config-if)#exit

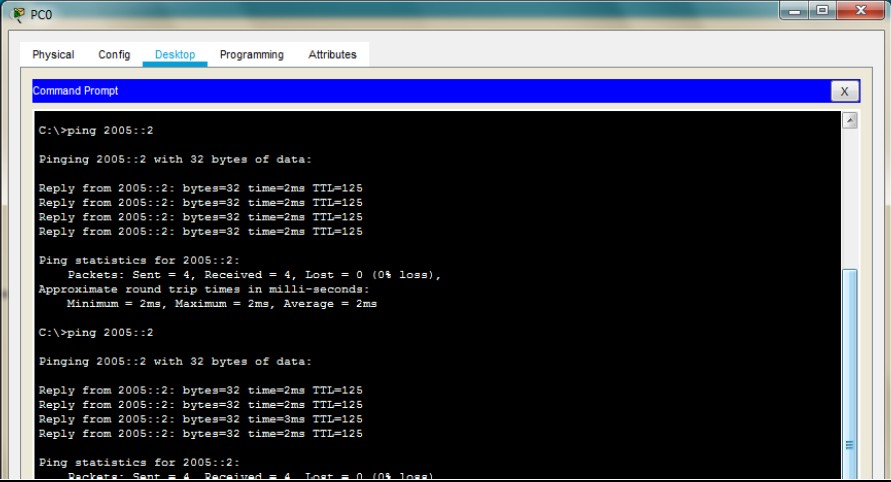
Router(config)#

**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 6**

**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:

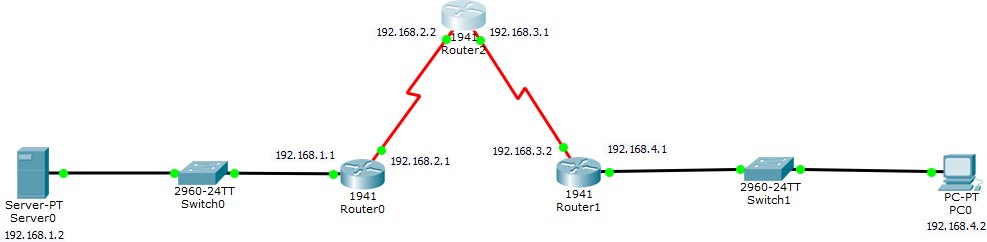
1. HTTP
2. Post Office Protocol (POP3),
3. Internet Mail Access Protocol (IMAP),
4. Simple Mail Transfer Protocol/Enhanced Simple Mail Transfer Protocol (SMTP/ESMTP)
5. Sun Remote Procedure Call (RPC)
6. Instant Messaging (IM) applications:
   1. Microsoft Messenger
   2. Yahoo! Messenger
   3. AOL Instant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

1. Authentication proxy
2. Stateful firewall failover
3. Unified firewall MIB
4. IPv6 stateful inspection
5. TCP out-of-order support

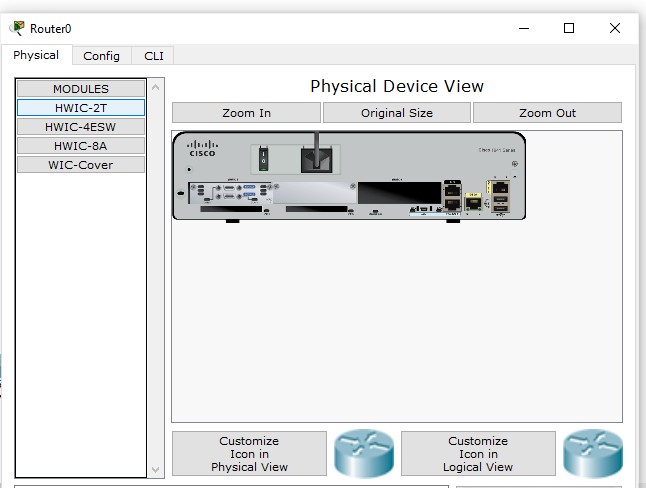
ZFW generally improves Cisco IOS performance for most firewall inspection activities. Neither Cisco IOS ZFW or Classic Firewall include stateful inspection support for multicast traffic.

## Consider the following topology

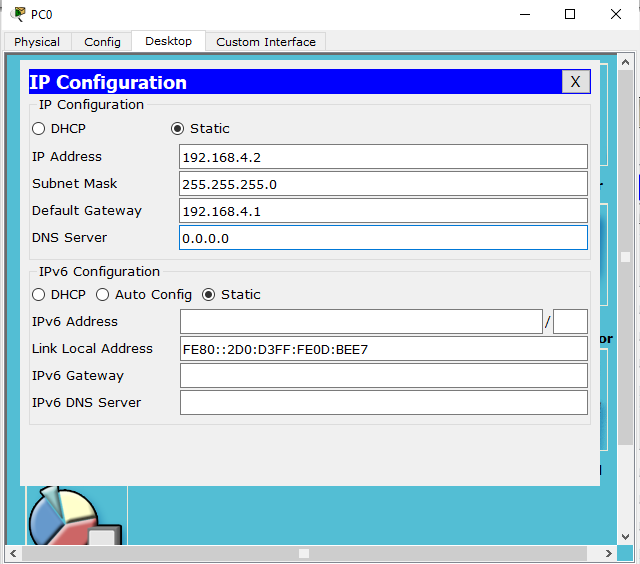


**Topology Configuration**

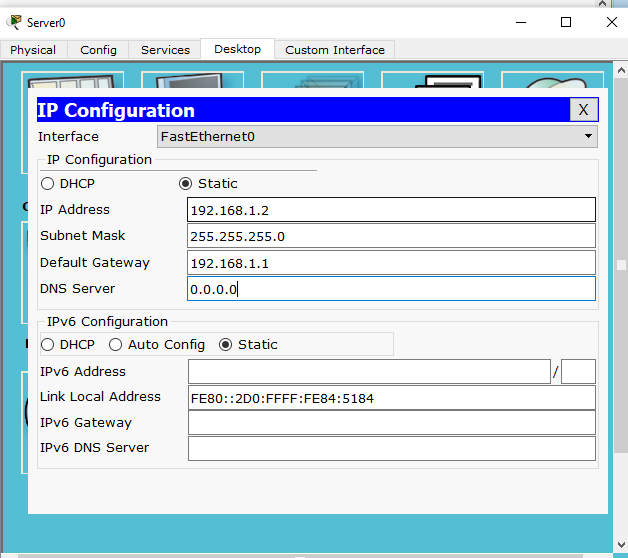
### Serial Interface must be added in each Router before configuring it The serial interface in each Router is added as follows

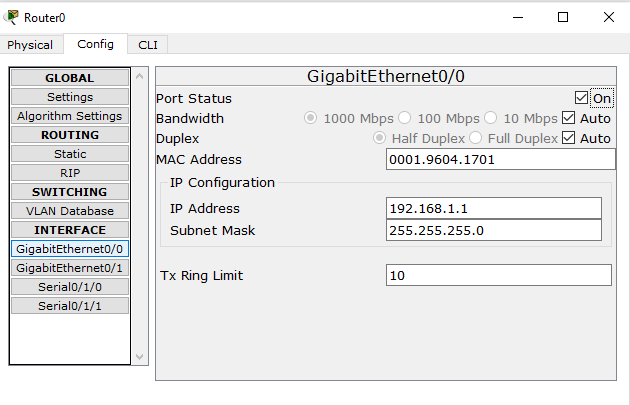


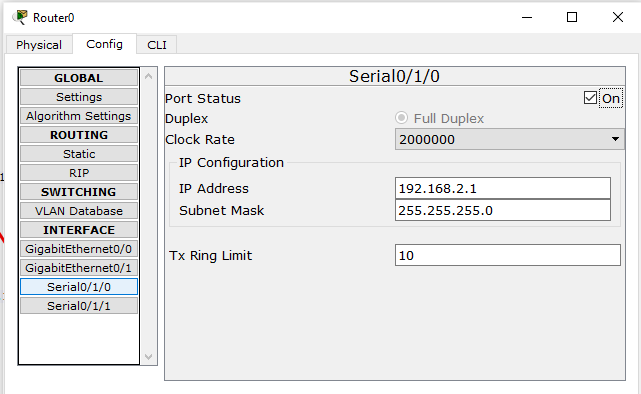
**Configuring PC0**

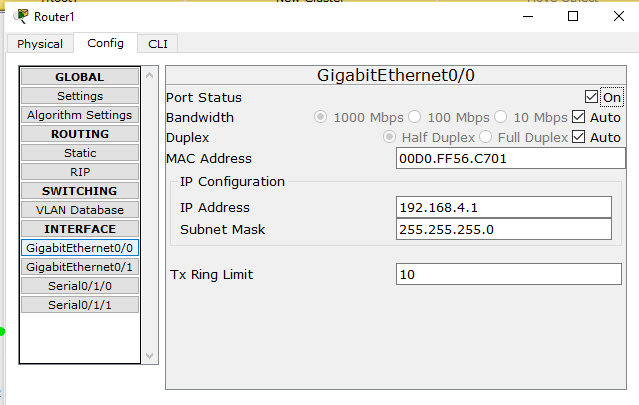


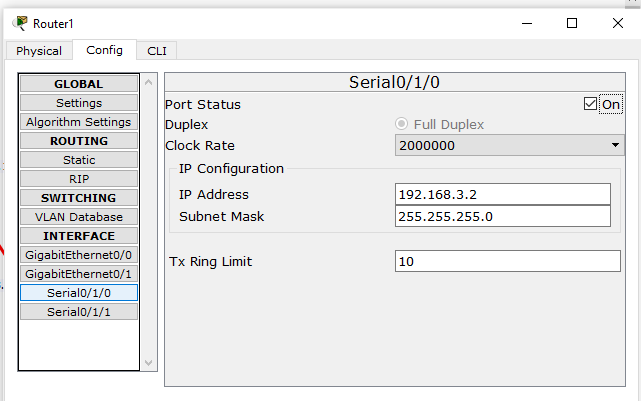
### Configuring Server0



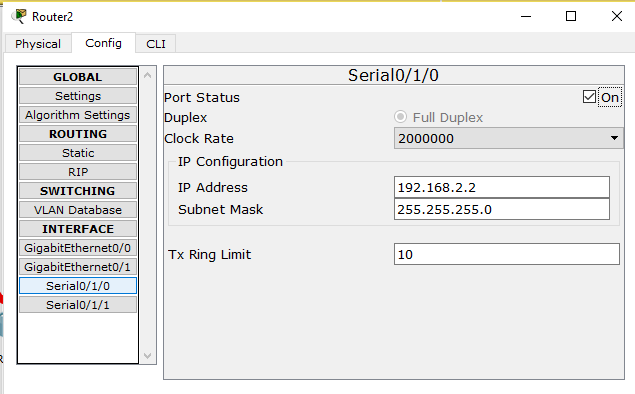








**Configuring Router2**

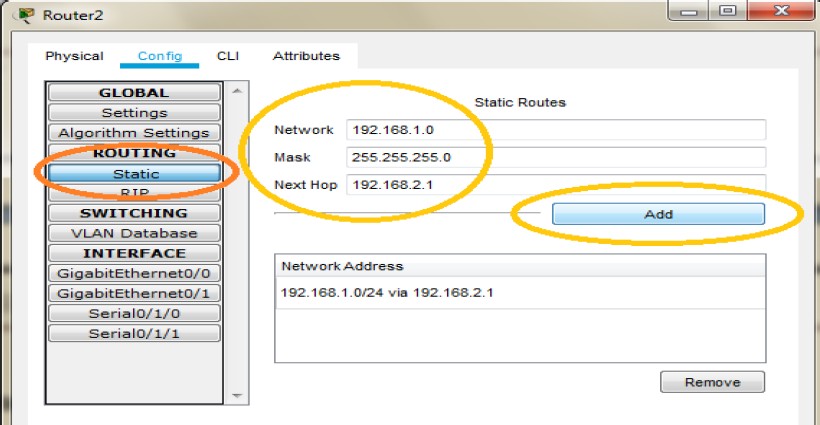


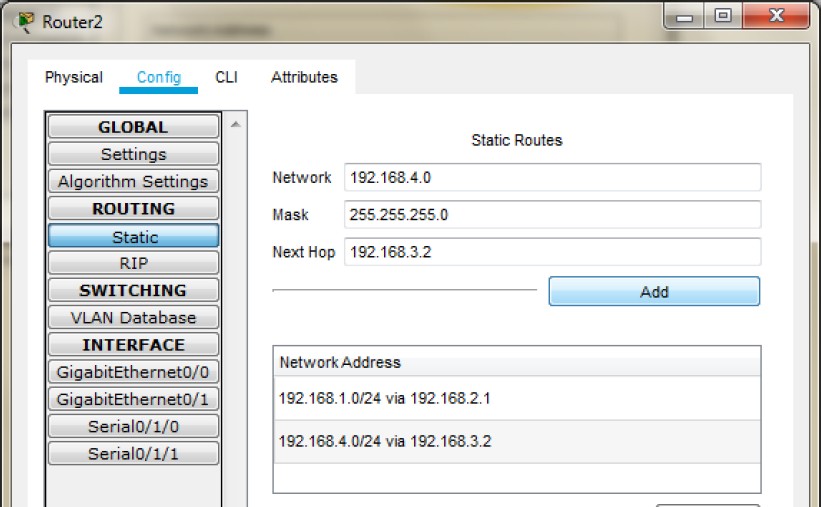


## Part 1: Static Routing

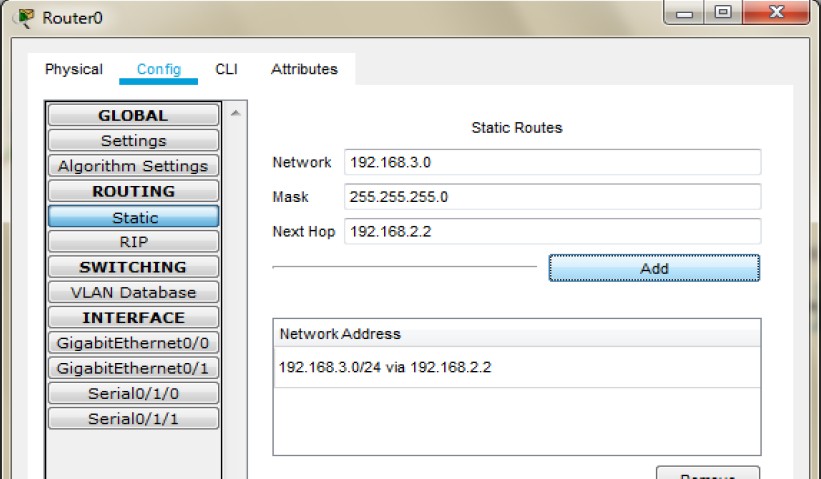
Static Routing is done using the following procedure for each Router

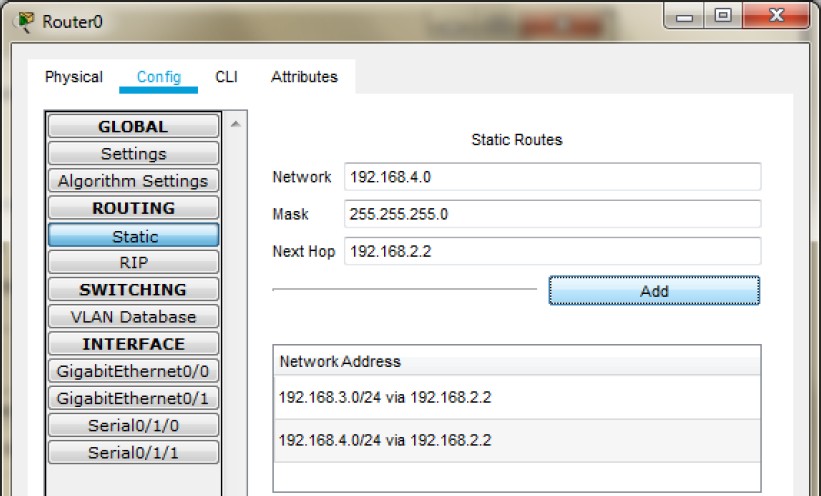
### Router 2: Add the following Routes in the Static mode





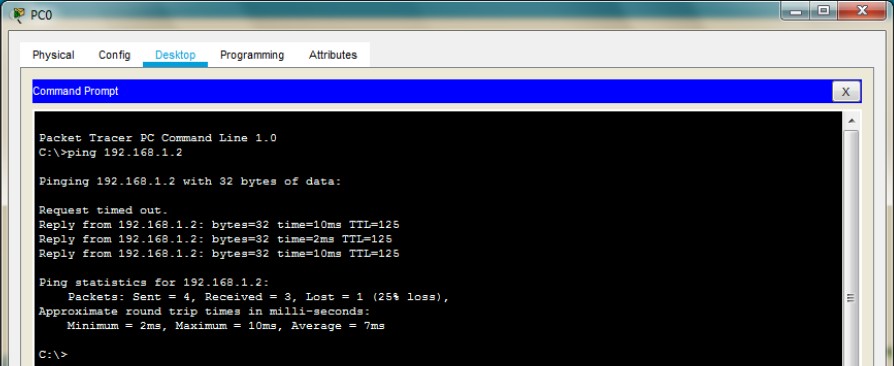
**Router 0: Add the following Routes in the Static mode**





### Router 1: Add the following Routes in the Static mode

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



## Part 2: Configuring SSH on Router 2

Type the following commands in the CLI mode of Router2

Router>en Router#conf t

Router(config)#ip domain-name dalmia.com Router(config)#hostname R2 R2(config)#crypto key generate rsa

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

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

How many bits in the modulus [512]: 1024

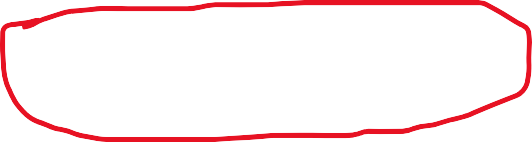
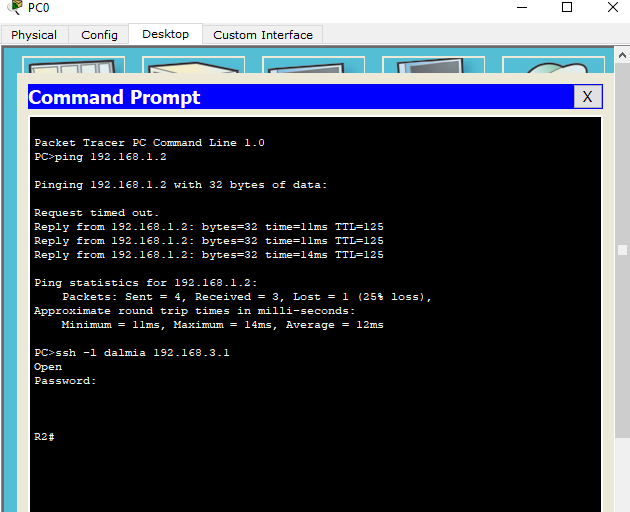
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK] R2(config)#line vty 0 4

\*Mar 2 0:52:50.777: %SSH-5-ENABLED: SSH 1.99 has been enabled R2(config-line)#transport input ssh

R2(config-line)#login local R2(config-line)#exit

R2(config)#username dalmia privilege 15 password cisco

Now we verify the SSH using PC as follows



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>en Router#conf t

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

Router(config)#exit Router#copy run start Press enter when prompted Router#reload

Continue with configuration dialog? [yes/no]: n Router>en

Router#conf t

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)#int G0/0

Router(config-if)#zone-member security in-zone Router(config-if)#exit

Router(config)#

Router(config)#int Se0/1/1

Router(config-if)#zone-member security out-zone Router(config-if)#exit

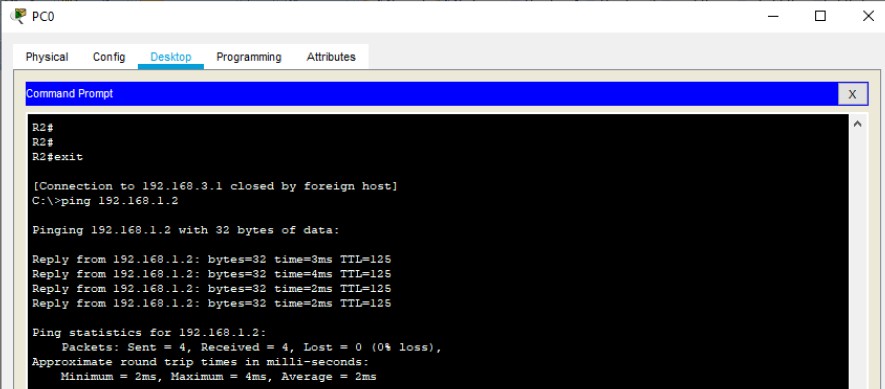
Router(config)#exit

Router#copy running-config startup-config Destination filename [startup-config]?

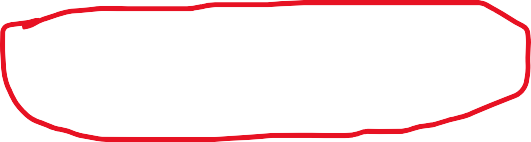
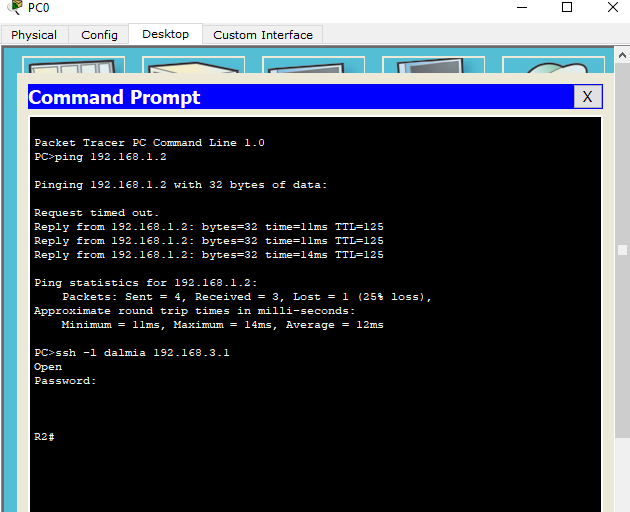
Building configuration... [OK]

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

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

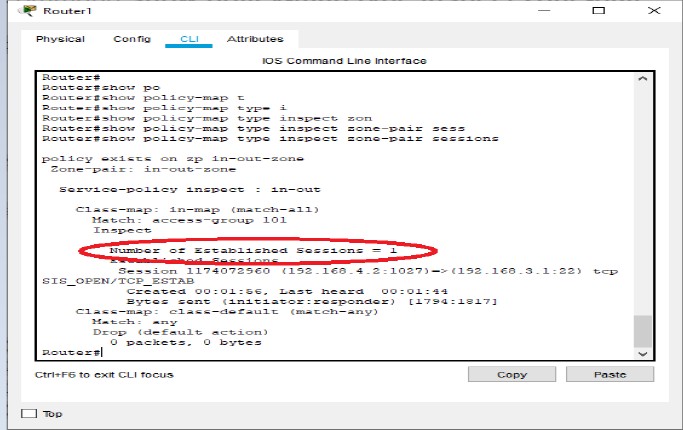


**Step 2: Start an SSH session from PC to Router 2 (ip 192.168.1.2)**



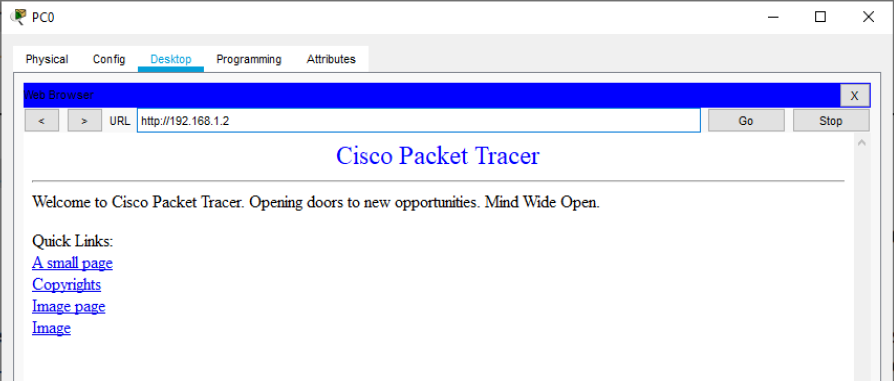
As seen above the session becomes active and we get access to Router2 (Do not exit and the session and continue to Step 3)

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



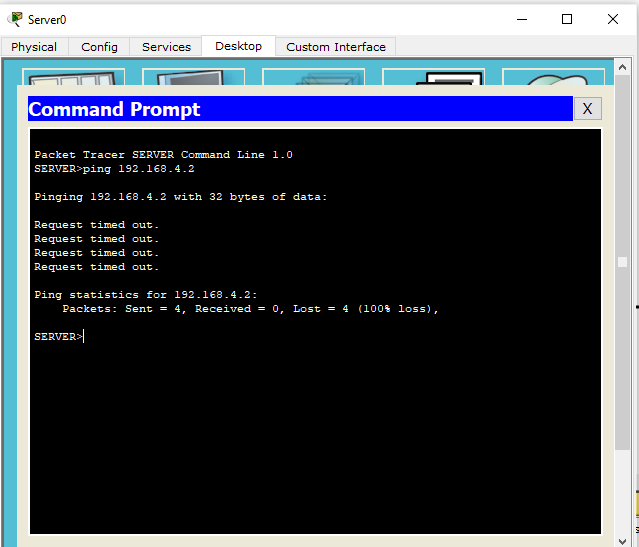
Router#show policy-map type inspect zone-pair sessions

**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 step: -

### Ping PC0 from the SERVER (it will result in Failure)



Hence the Firewall functionality has been verified.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical 7**

**Configuring IOS Intrusion Prevention System (IPS) Using the CLI: a) Enable IOS IPS.**

**b) Modify an IPS signature.**

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.

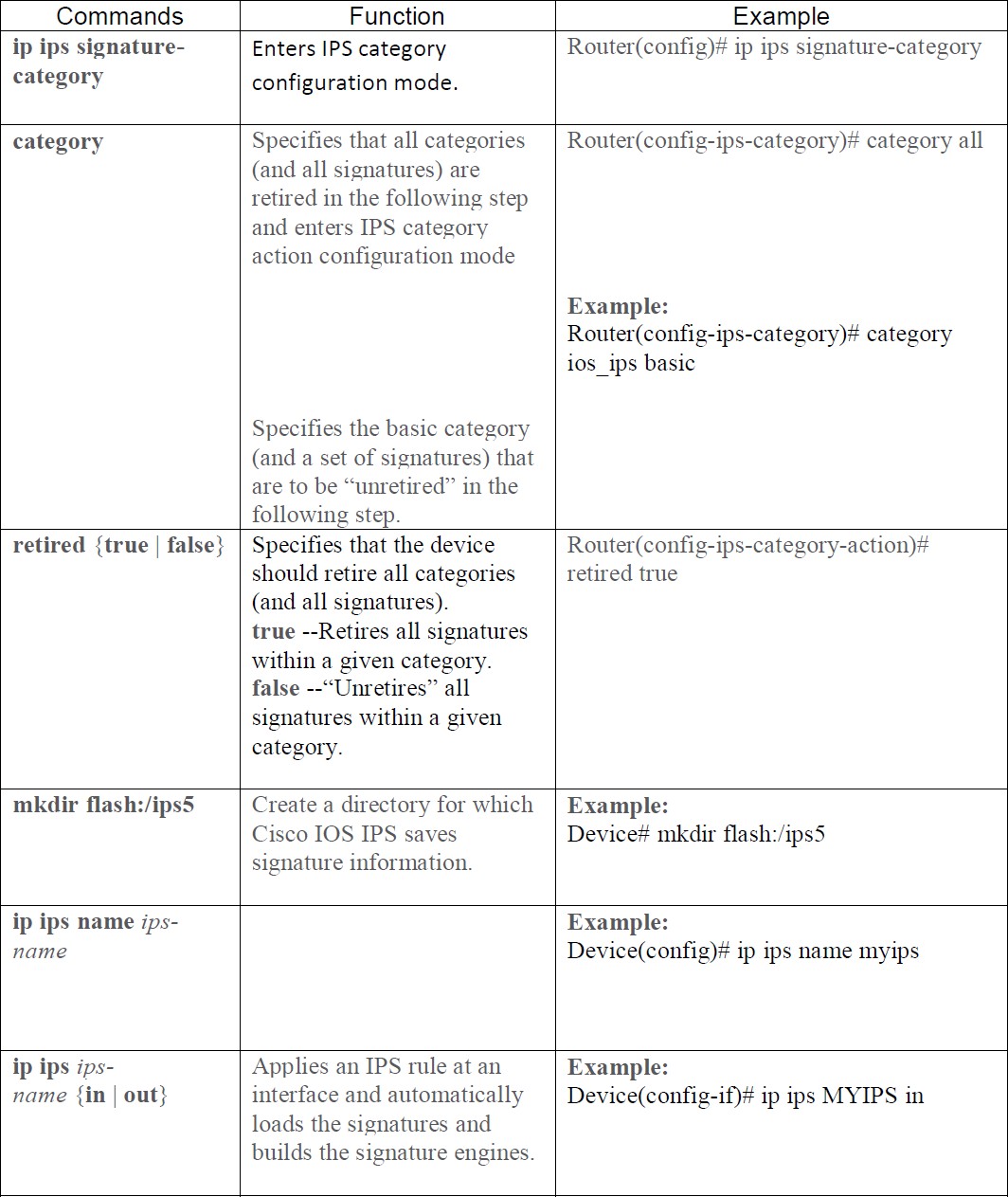
Cisco developed its Cisco IOS software-based intrusion-prevention capabilities and Cisco IOS Firewall with flexibility in mind, so that individual signatures could be disabled in case of false positives. Generally, it is preferable to enable both the firewall and Cisco IOS IPS to support network security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

### Signatures:

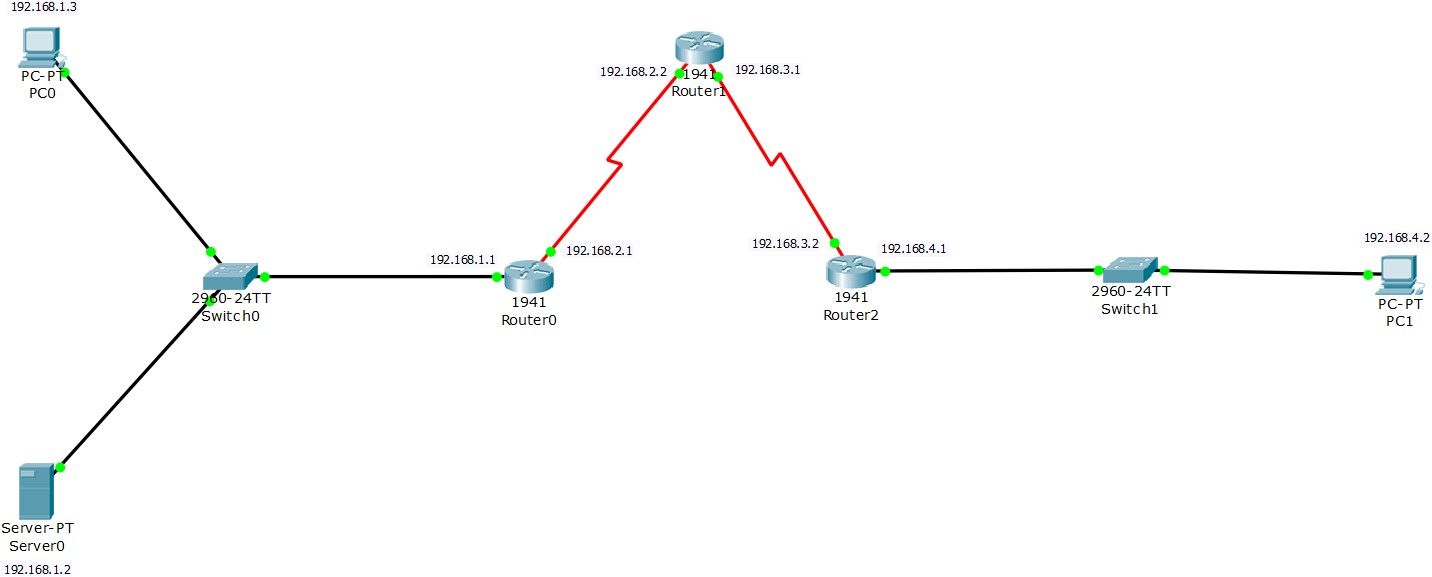
A signature is a set of rules that an IDS and an IPS use to detect typical intrusive activity, such as DoS attacks. We can easily install signatures using IDS and IPS management software such as Cisco IDM. Sensors enables us to modify existing signatures and define new ones.

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 define some of the commands which will be used while configuring the Router for IPS.

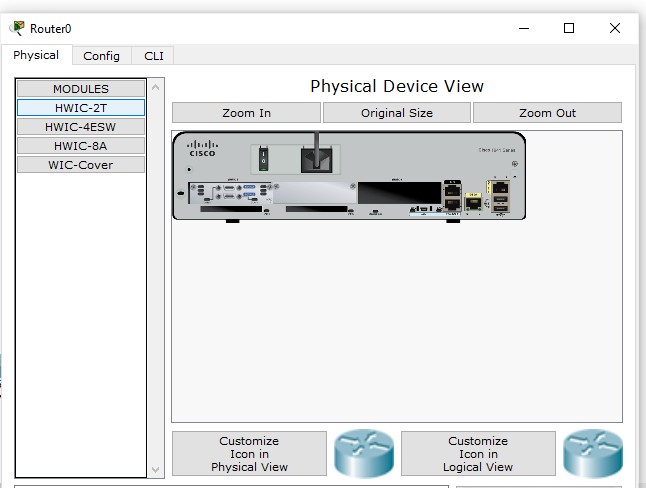


## Consider the following topology

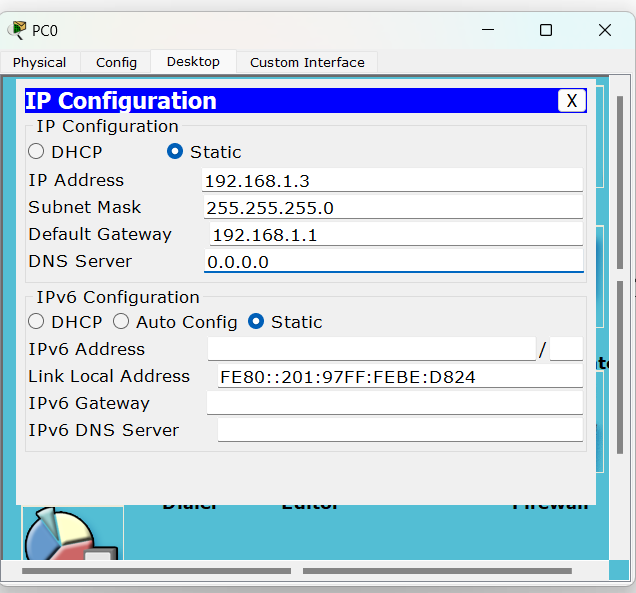


**Topology Configuration**

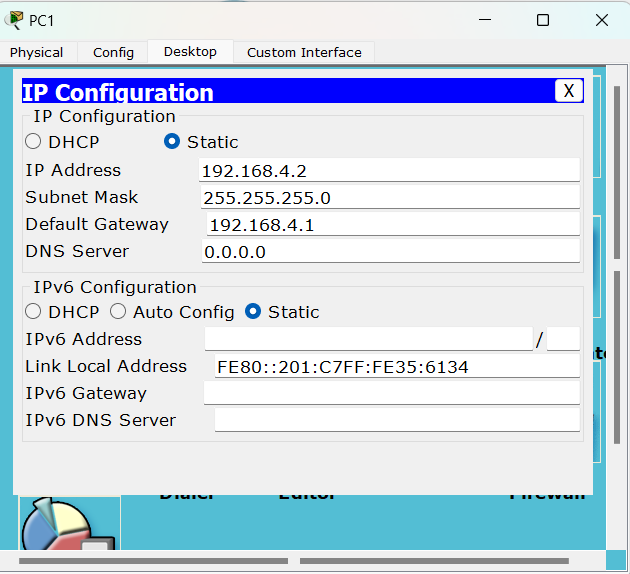
### Serial Interface must be added in each Router before configuring it The serial interface in each Router is added as follows



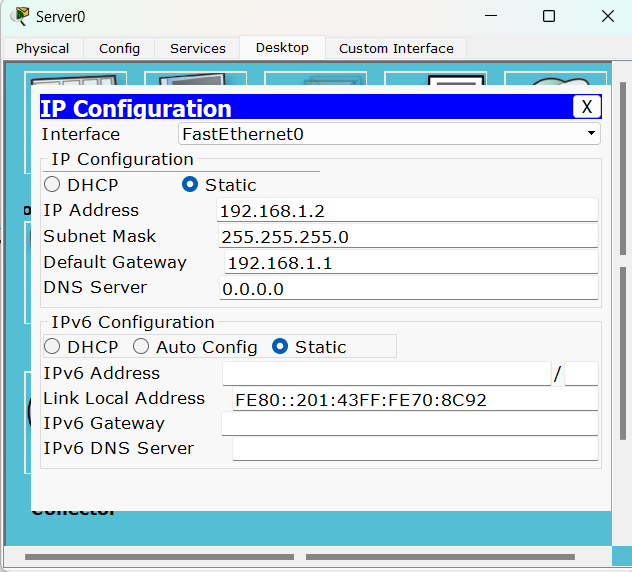
**Configuring PC0**



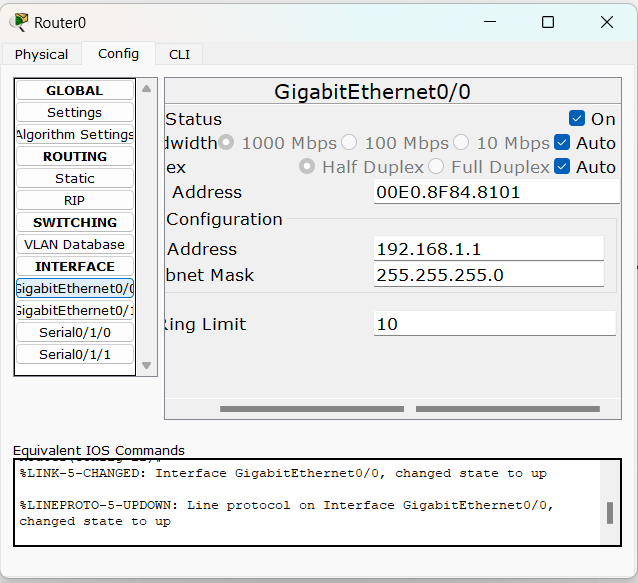
### Configuring PC1

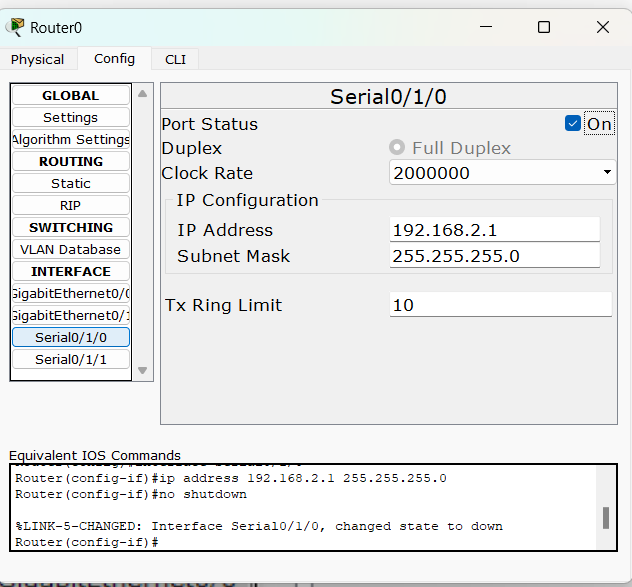


**Configuring Server0**

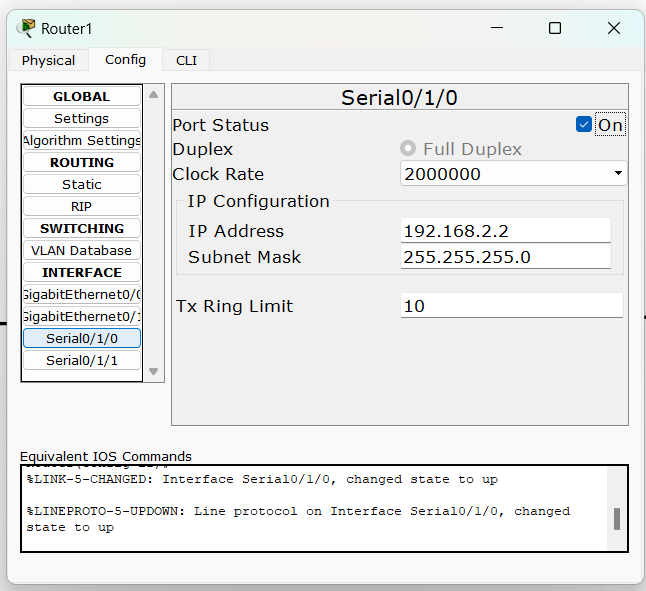


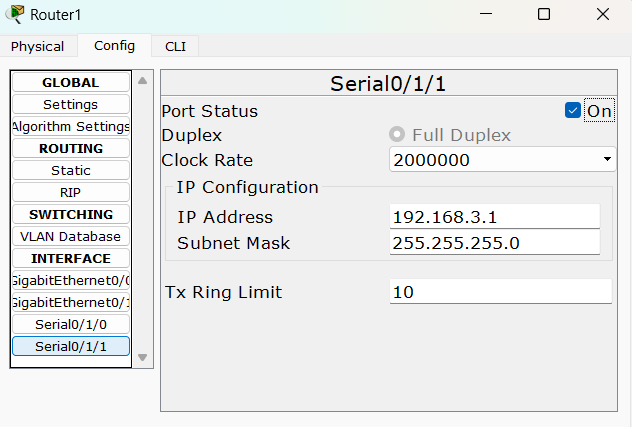
### Configuring Router0



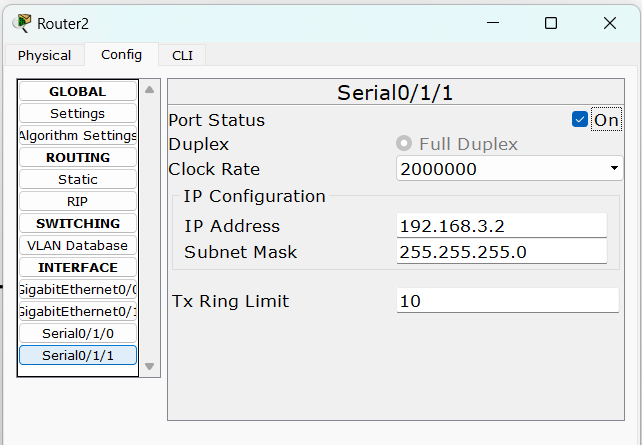


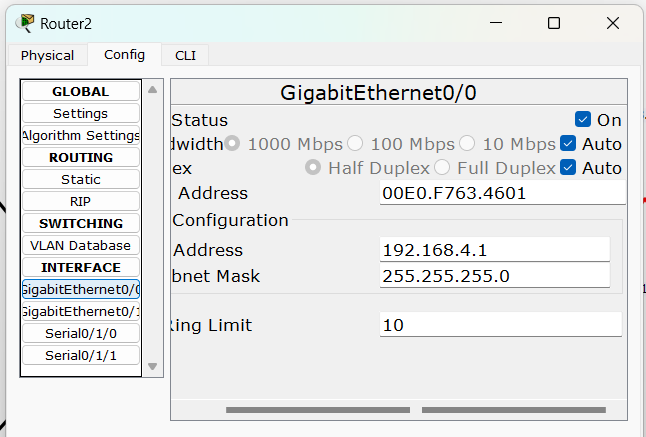
**Configuring Router1**



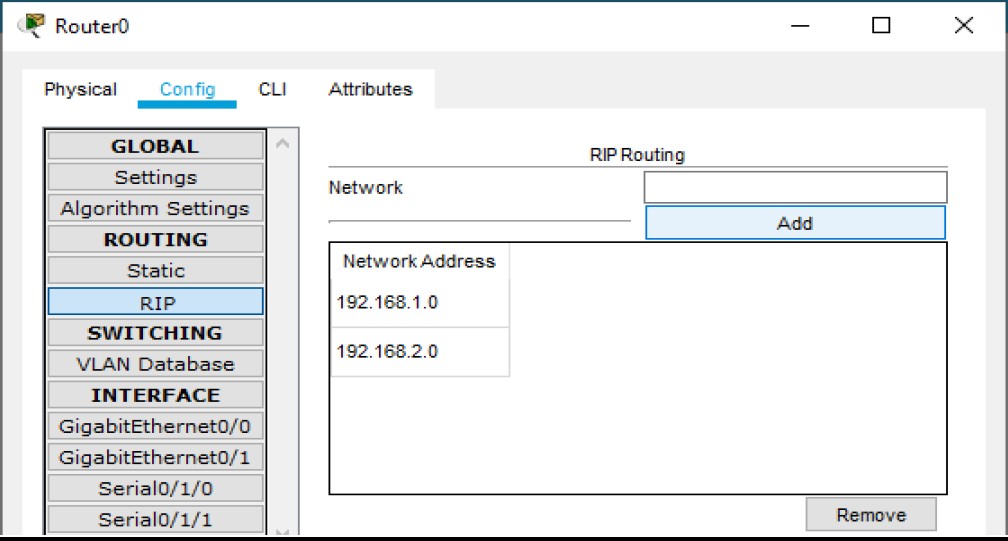


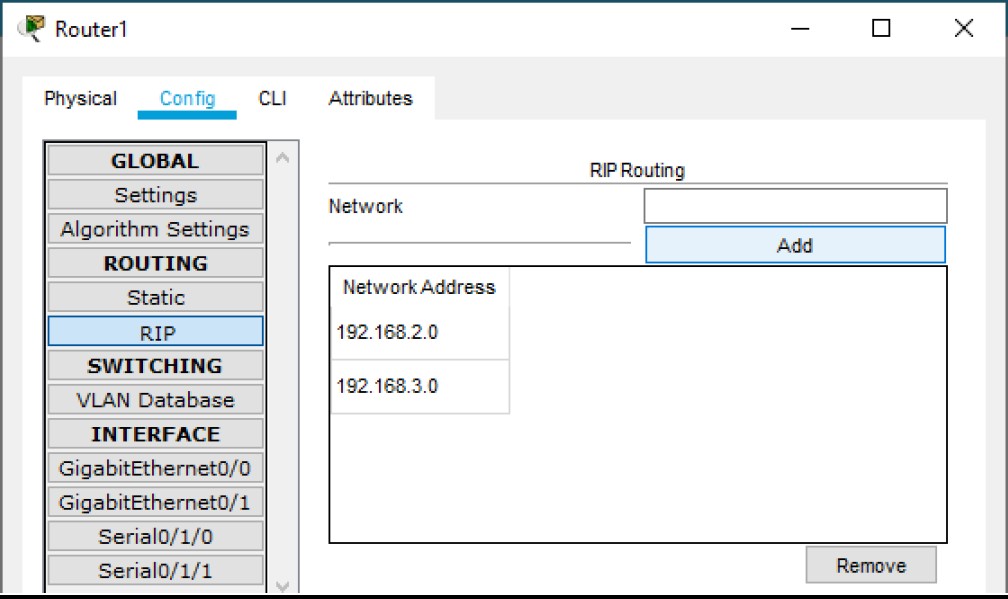
**Configuring Router2**



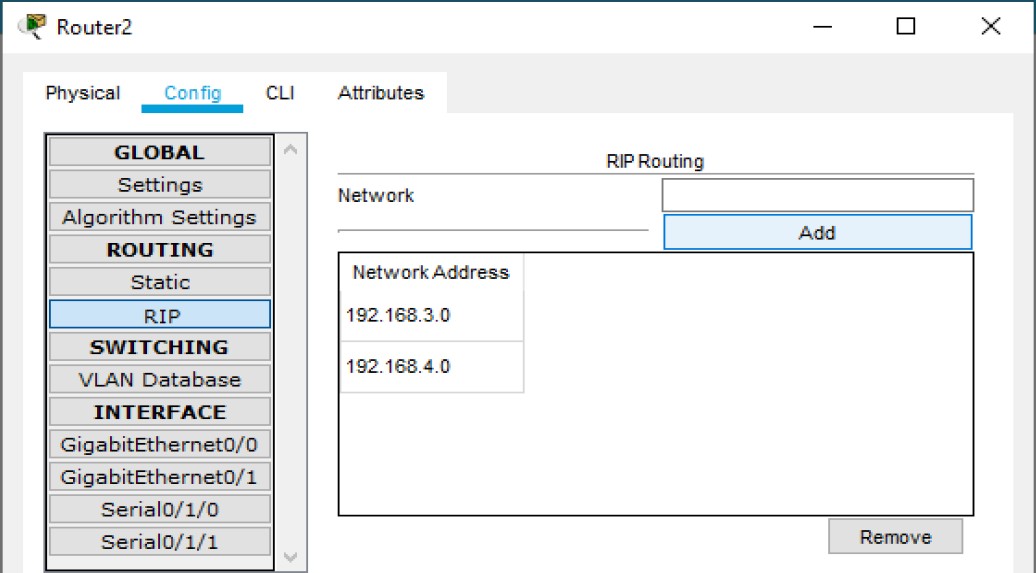


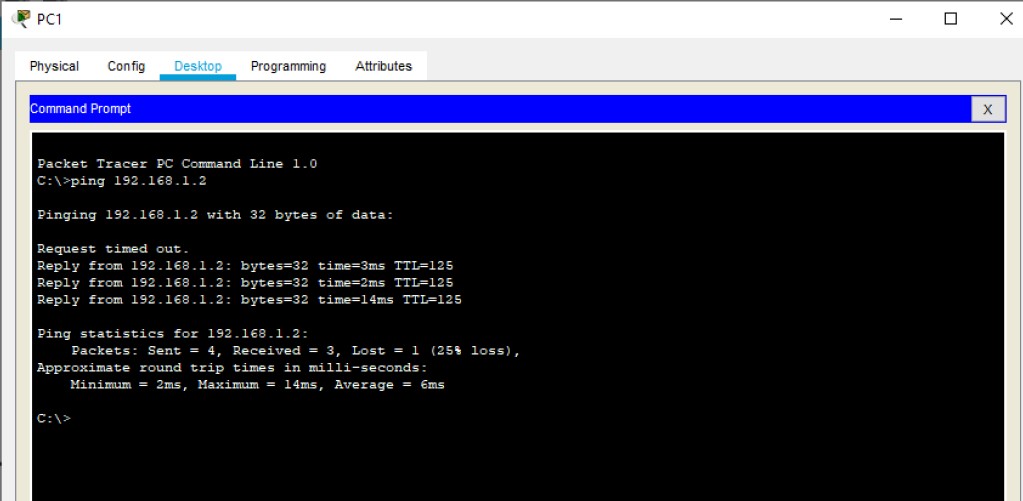
## We need to set the Routing table in all the Routers so that each node could send and receive packets from others (RIP is set in all the Routers as follows)

**Router0**

**Router1**

**Router2**



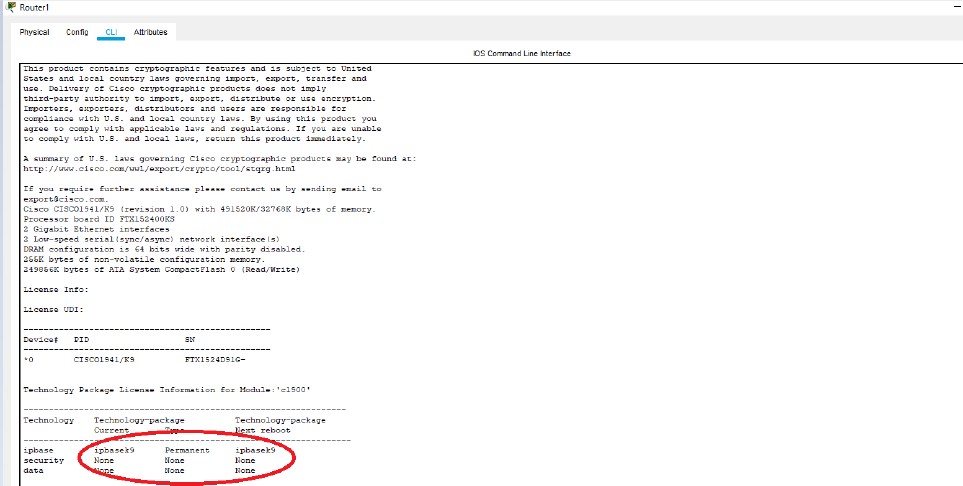
Now we can check the connectivity by sending ping commands from any node to any other node

**So, we conclude that the connectivity has been established**

## Part 1: 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

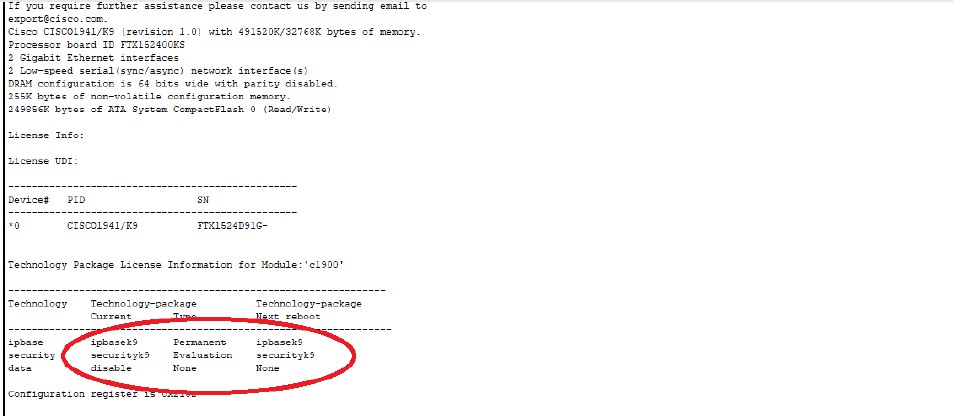
### As seen above the security package is not enabled, to enable the security feature, type the following command in Router1

Router>en Router#conf t

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

Router(config)#exit Router#copy run start Press enter when prompted Router#reload

Continue with configuration dialog? [yes/no]: n 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>en Router#mkdir dalmia

Create directory filename [dalmia]? Created dir flash:dalmia Router#conf t

Router(config)#ip ips config location flash:dalmia 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] Applying Category configuration to signatures ...

%IPS-6-ENGINE\_BUILDING: atomic-ip - 288 signatures - 6 of 13 engines

%IPS-6-ENGINE\_READY: atomic-ip - build time 30 ms - packets for this engine will be scanned

Router(config)#int Se0/1/0 Router(config-if)#ip ips iosips out

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

## Part 2: Modify the Signature

### Type the following commands in the CLI mode of Router1

Router#conf t

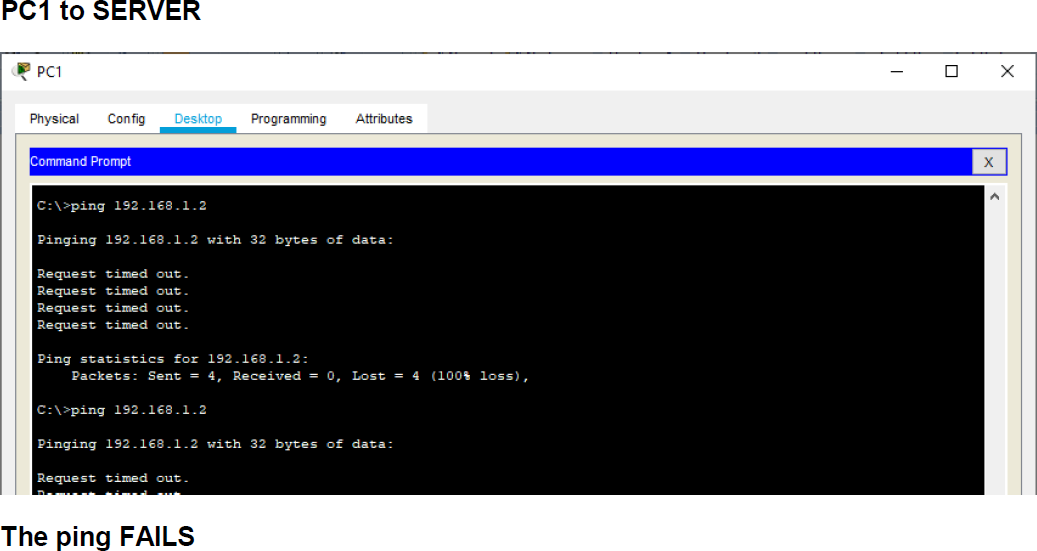
Router(config)#ip ips signature-definition Router(config-sigdef)#signature 2004 0 Router(config-sigdef-sig)#status Router(config-sigdef-sig-status)#retired false Router(config-sigdef-sig-status)#enabled true Router(config-sigdef-sig-status)#exit Router(config-sigdef-sig)#engine

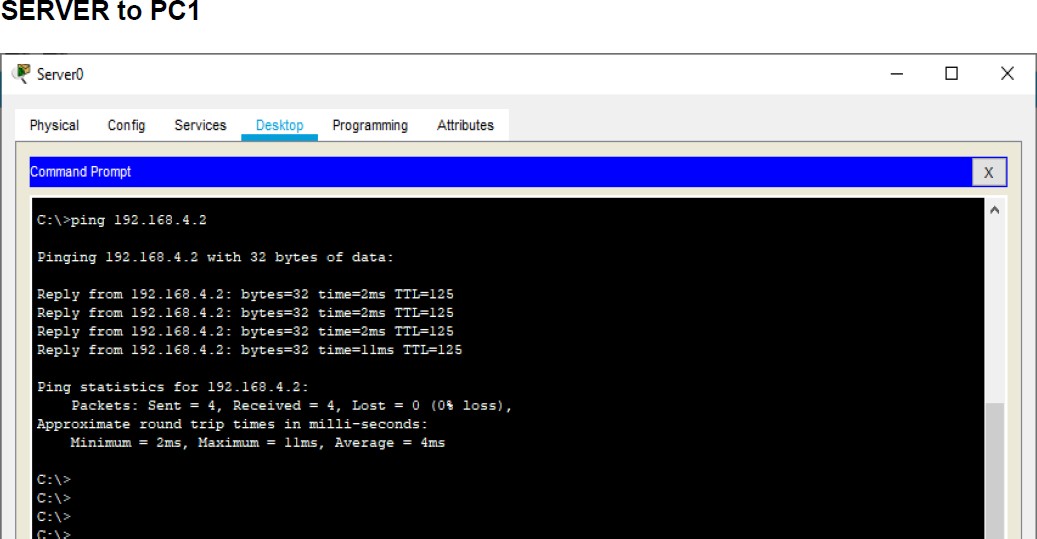
Router(config-sigdef-sig-engine)#event-action produce-alert Router(config-sigdef-sig-engine)#event-action deny-packet-inline Router(config-sigdef-sig-engine)#exit

Router(config-sigdef-sig)#exit Router(config-sigdef)#exit

Do you want to accept these changes? [confirm] <Enter> 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**





### Also, we can observe the Syslog service in the SERVER to check the log activities

Use show commands to verify IPS on Router1

Router#show ip ips all

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical 8**

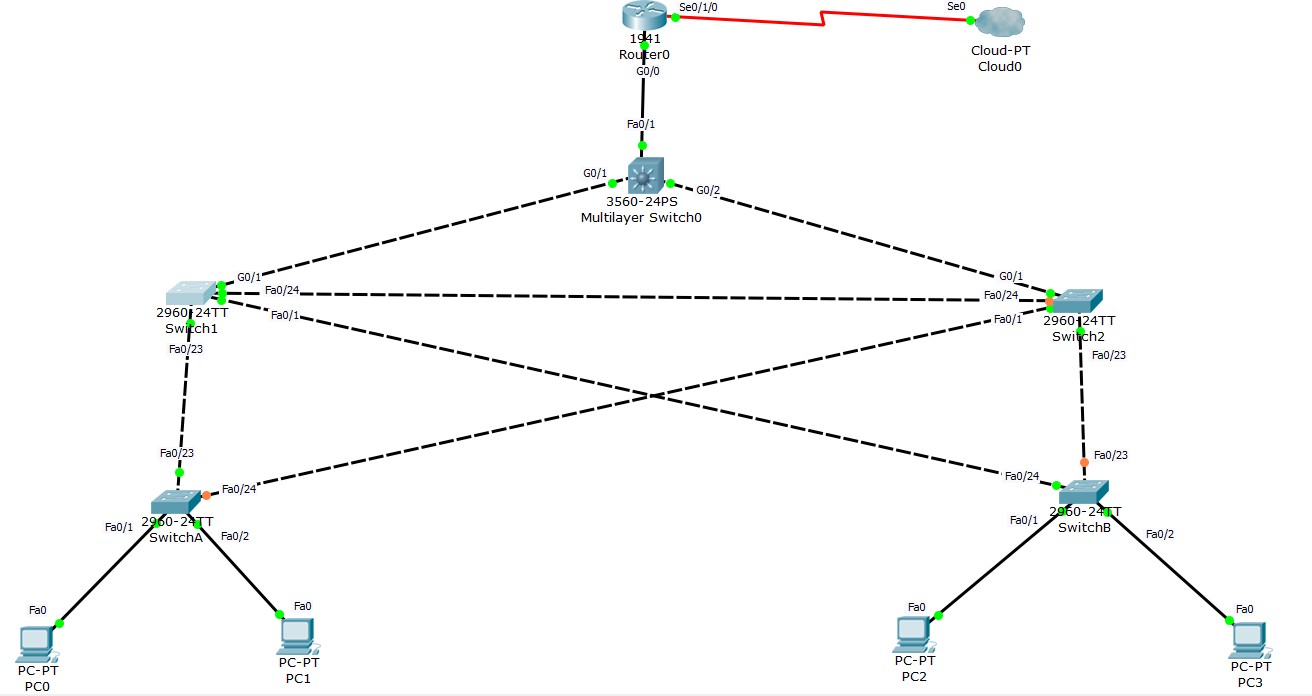
**Layer 2 Security a) Assign the Central switch as the root bridge.**

**b) Secure spanning-tree parameters to prevent STP manipulation attacks. c) Enable port security and disable unused ports.**

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

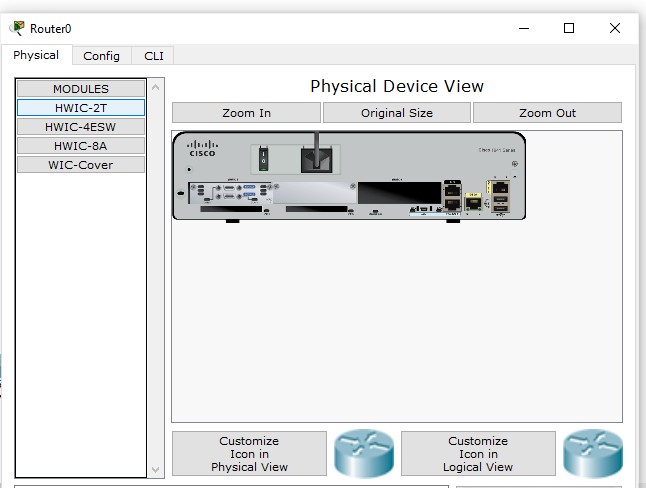
For optimum performance and security, the administrator would like to ensure that the root bridge is the 3560 multi-layer switch. To prevent spanning-tree manipulation attacks, the administrator wants to ensure that the STP parameters are secure. To prevent against CAM (Content Addressable Memory) table 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.

## Consider the following topology

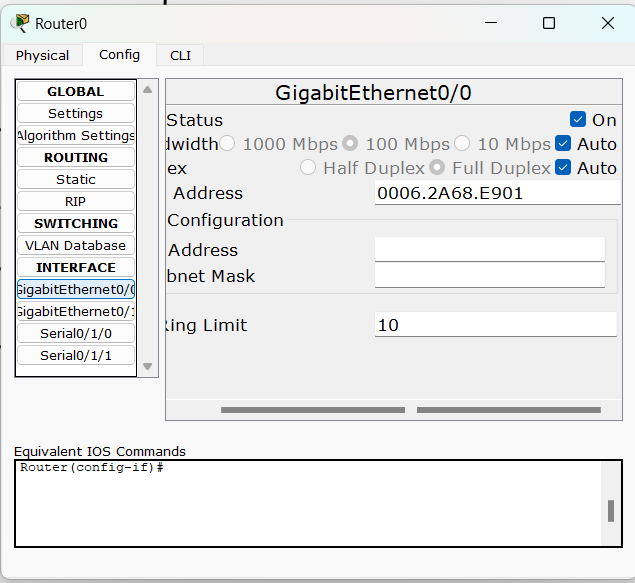


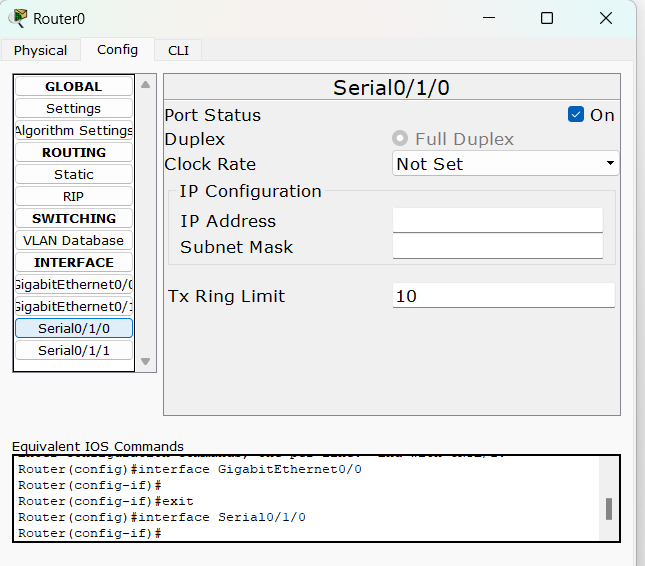
**Topology Configuration**

### Serial Interface must be added in the Router0 before configuring it The serial interface in Router0 is added as follows



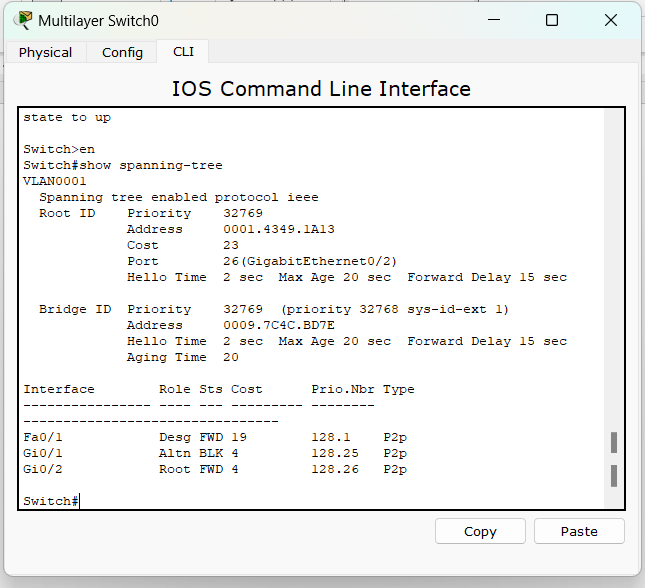
**Configuring Router0**





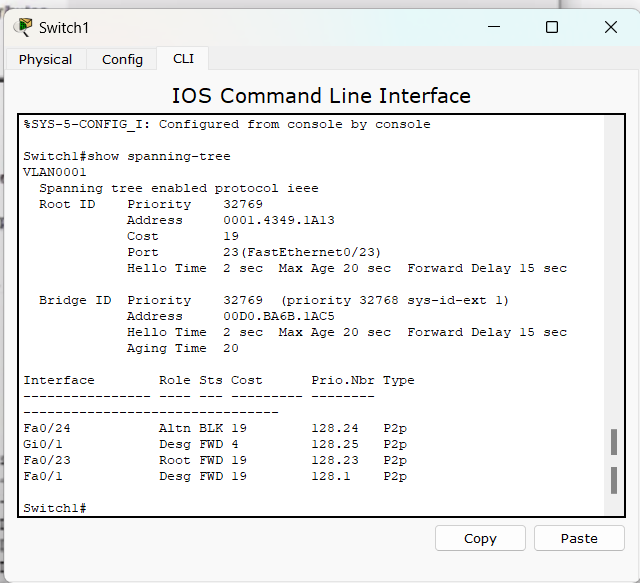
## Part 1: Configure Root Bridge

### Step 1: Determine the current root bridge.

From **Multilayer Switch0**, issue the **show spanning-tree** command to determine the current root bridge, to see the ports in use, and to see their status.

switch>en

switch# show spanning-tree



**Step 2: Assign Multilayer Switch0 as the primary root bridge.** Using the **spanning-tree vlan 1 root primary** command, and assign **Multilayer switch0** as the root bridge.

switch#conf t

switch(config)#spanning-tree vlan 1 root primary switch(config)#do show span

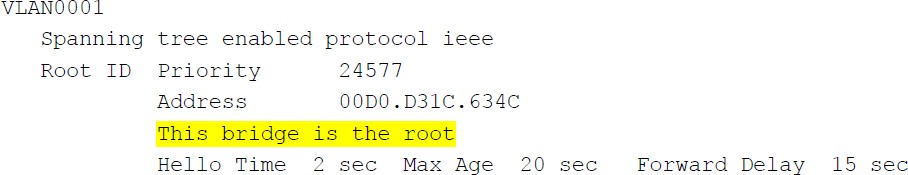
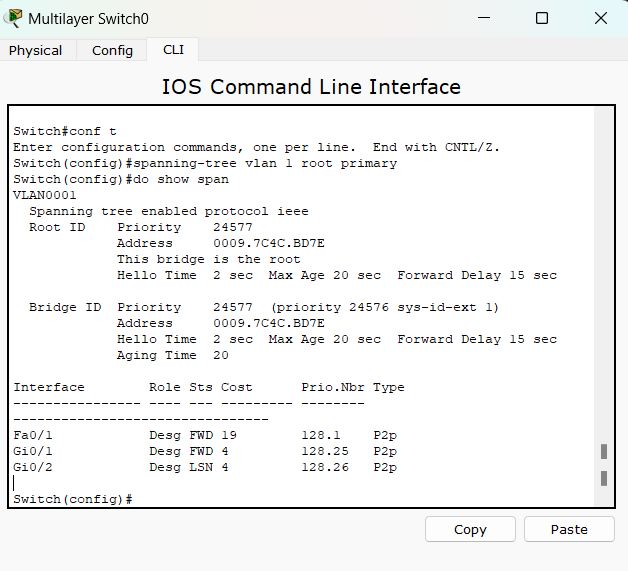
### Step 3: Assign Switch1 as a secondary root bridge. Assign SW-1 as the secondary root bridge using the spanning-tree vlan 1 root secondary command.

switch#conf t

switch(config)#spanning-tree vlan 1 root secondary

**Step 4: Verify the spanning-tree configuration. Issue the show spanning-tree command to verify that Multi-layer Switch0 is the root bridge.**

switch# show spanning-tree



## Part 2: Protect Against STP Attacks

### Secure the STP parameters to prevent STP manipulation attacks.

**Step 1: Enable PortFast on all access ports.**

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

SwitchA>en SwitchA#conf t

SwitchA(config)#int range f0/1-2 SwitchA(config-if-range)#spanning-tree portfast

SwitchB>en SwitchB#conf t

SwitchB(config)#int range f0/1-2 SwitchB(config-if-range)#spanning-tree portfast

### Step 2: Enable BPDU guard on all access ports.

BPDU guard is a feature that can help prevent rogue switches and spoofing on access ports. Enable BPDU guard on SwitchA and SwitchB access ports.

SwitchA(config)#int range f0/1-2

SwitchA(config-if-range)#spanning-tree bpduguard enable

SwitchB(config)#int range f0/1-2

SwitchB(config-if-range)#spanning-tree bpduguard enable

### Step 3: Enable root guard.

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

On Switch1, enable root guard on ports F0/23 and F0/24. On Switch2, enable root guard on ports F0/23 and F0/24.

Switch1>en Switch1#conf t

Switch1(config)#int range f0/23-24 Switch1(config-if-range)#spanning-tree guard root

Switch2>en Switch2#conf t

Switch2(config)#int range f0/23-24 Switch2(config-if-range)#spanning-tree guard root

## Part 3: Configure Port Security and Disable Unused Ports

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

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

SwitchA>en SwitchA#conf t

SwitchA(config)#int range f0/1-2

SwitchA(config-if-range)#switchport mode access SwitchA(config-if-range)#switchport port-security SwitchA(config-if-range)#switchport port-security maximum 2

SwitchA(config-if-range)#switchport port-security violation shutdown SwitchA(config-if-range)#switchport port-security mac-address sticky

SwitchB>en SwitchB#conf t

SwitchB(config)#int range f0/1-2

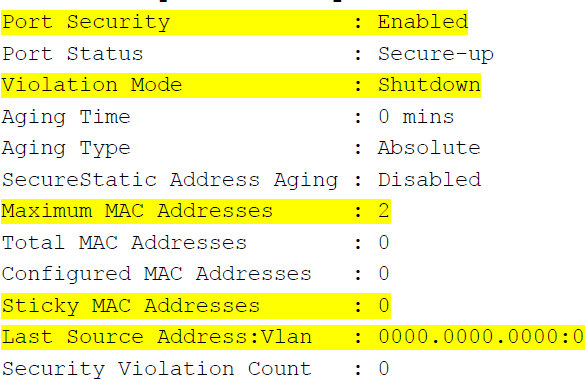
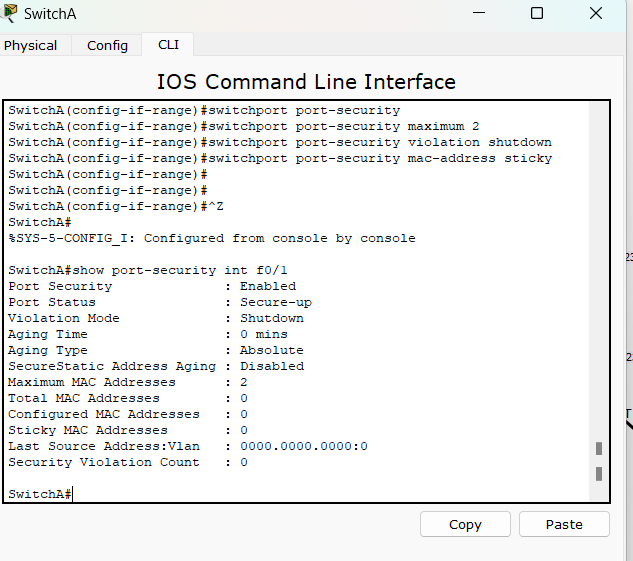
SwitchB(config-if-range)#switchport mode access SwitchB(config-if-range)#switchport port-security SwitchB(config-if-range)#switchport port-security maximum 2

SwitchB(config-if-range)#switchport port-security violation shutdown SwitchB(config-if-range)#switchport port-security mac-address sticky

## Step 2: Verify port security.

### On SwitchA, issue the command show port-security int f0/1 to verify that port security has been configured.

SwitchA#show port-security int f0/1



**Step 3: Disable unused ports.**

Disable all ports that are currently unused.

SwitchA(config)#int range f0/3-22 SwitchA(config-if-range)#shutdown

SwitchB(config)#int range f0/3-22 SwitchB(config-if-range)#shutdown

Hence the Port security has been enabled.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***