# Security in Computing

## Journal 2022-23

**Name: _____**

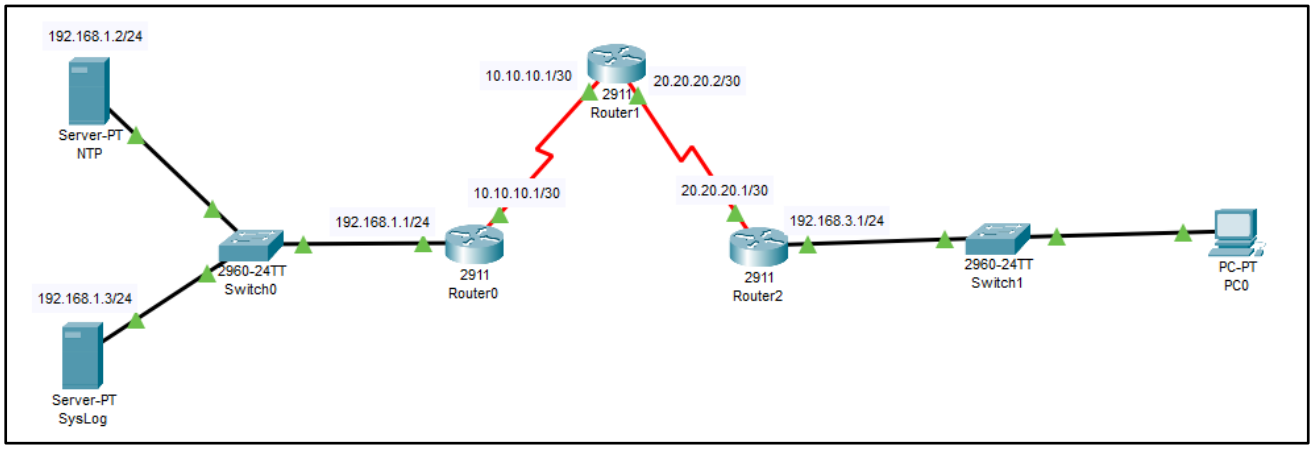
**Roll No: _____**

**Class: TY-BSc.IT**

# Index

# Practical 1: Configure Cisco Routers for OSPF, Syslog, NTP, and SSH Operations

**Topology:**




**Addressing Table:**

| Device | Interface | IP-Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| NTP | | 192.168.1.2 | 255.225.255.0 | 192.168.1.1 |
| SYSLOG | | 192.168.1.3 | 255.255.255.0 | 192.168.1.1 |
| R0 | Gig0/0 | 192.168.1.1 | 255.255.255.0 | N/A |
| | s0/0/0 | 10.10.10.1 | 255.255.255.252 | |
| R1 | s0/0/0 | 10.10.10.2 | 255.255.255.252 | |
| | s0/0/1 | 20.20.20.2 | 255.255.255.252 | |
| R2 | s0/0/1 | 20.20.20.1 | 255.255.255.252 | |
| | Gig0/0 | 192.168.3.1 | 255.255.255.0 | |
| PC0 | FastE0 | 192.168.3.5 | 255.255.255.0 | 192.168.3.1 |

## A. Configure OSPF MD5 Authentication

**Step 1:** Configure OSPF for Router0, Router1, Router2 on Each Interface.

**On Router0:**

```
Router>en
Router#conf t
Router(config)#router ospf 1
Router(config-router)#network 192.168.1.0  0.0.0.255   area 0
Router(config-router)#network 10.10.10.0 0.0.0.3 area 0
```

Network ID

Wild Subnet Mask

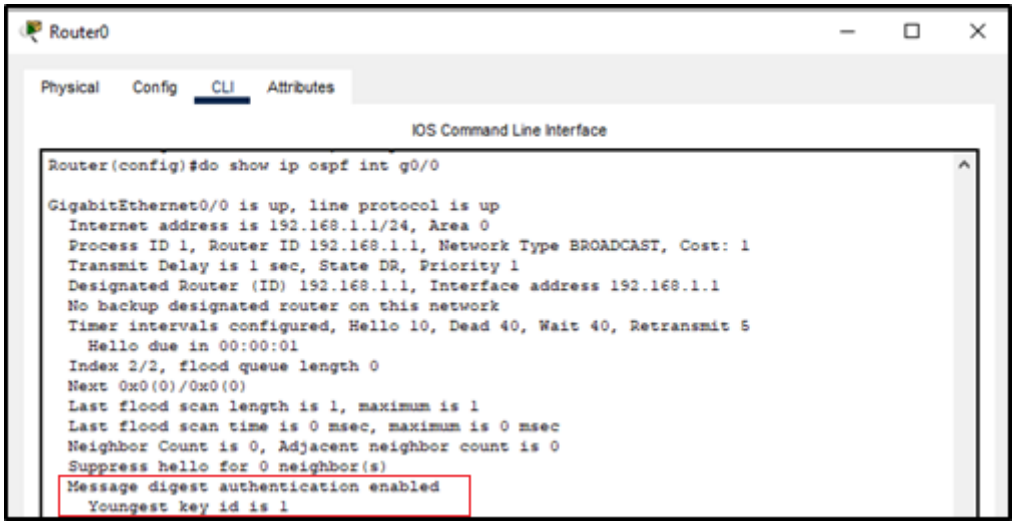
**Step 2:** Configure MD5 Key for all Routers.

```
Router>en
Router#conf t
Router(config)#router ospf 1
Router(config-router)#area 0 authentication message-digest
Router(config-router)#int g0/0
Router(config-if)#ip ospf message-digest-key 1 md5 Password
Router(config-if)#int s0/0/0
Router(config-if)#ip ospf message-digest-key 1 md5 Password
```

**Perform Step 1 and Step 2 On all the Routers.**

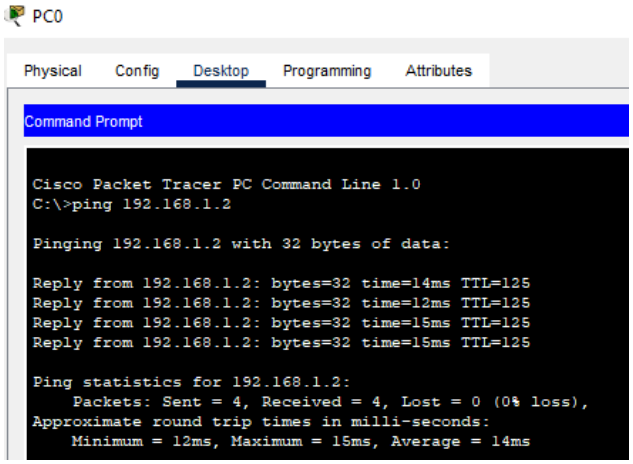
**Step 4: Verify OSPF MD5 Authentication**

```
Router>en
Router#conf t
Router(config)# do show ip ospf int g0/0
```

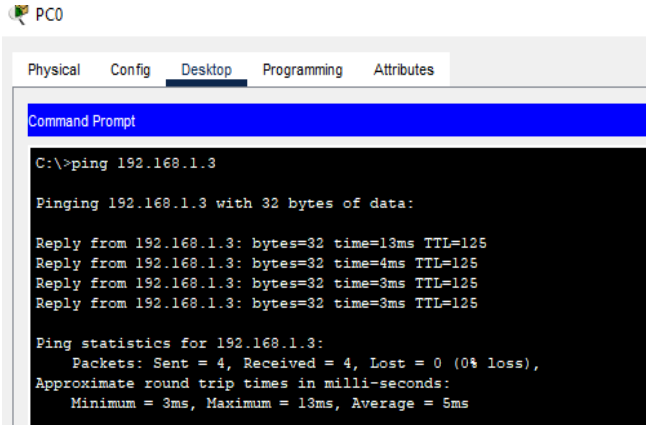



**Step 4: Testing OSPF By Pinging Each Devices.**

Trying to Ping from PC0 to NTP Server

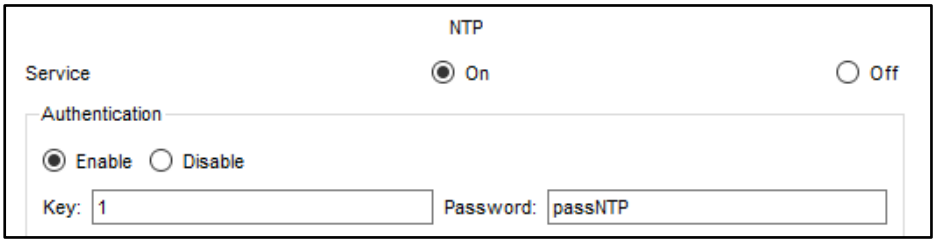



Trying to Ping from PC0 to SysLog Server



## B. Configure NTP Server

**Step 1: Enable NTP Services on NTP Server and Setup Authentication Key & Password**




**Step 2: Configure Router0, Router1, and Router2 as NTP clients.**

First Check Time on Each Router by Command

```
R0(config)#do show clock

*0:50:40.436 UTC Mon Mar 1 1993
```

**Setting Up NTP Client**

```
R0(config)#ntp authentication-key 1 md5 passNTP
R0(config)#ntp authenticate
R0(config)#ntp trusted-key 1
R0(config)#ntp server 192.168.1.2 key 1
R0(config)#ntp update-calendar
R0(config)#do show clock
23:12:48.64 UTC Wed Feb 22 2023
```

## C. Configure SysLog Server

**Step 1: Enable SYSLOG Services on SysLog Server**


**Step 2: Configure Routers to Log Messages to the Syslog Server**

**On Each Router Type:**

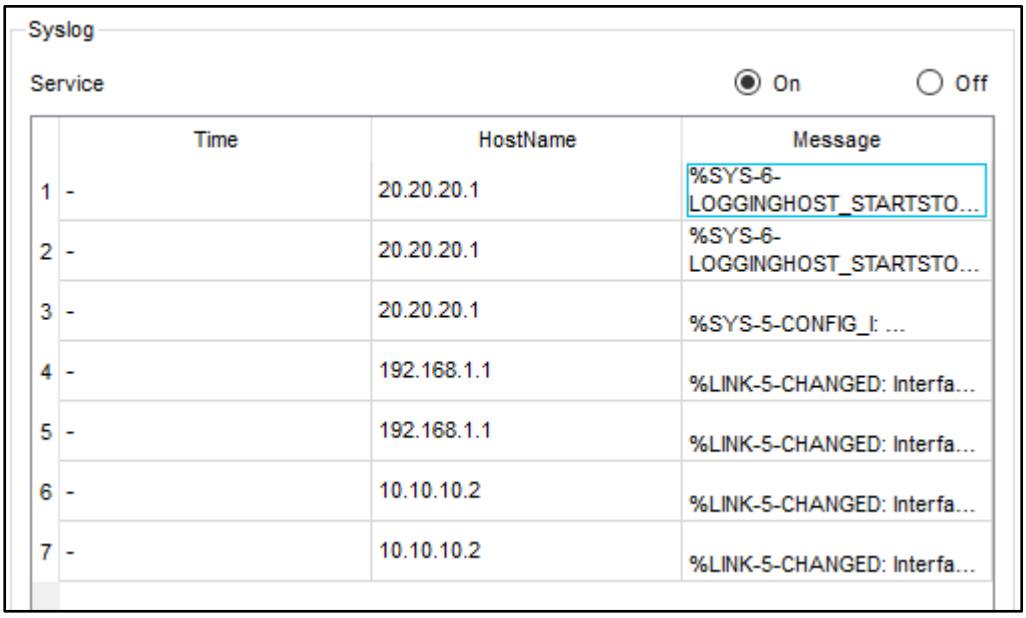
| |
|---|
| R0>en |
| R0#conf t |
| Enter configuration commands, one per line. End with CNTL/Z. |
| R0(config)#logging host 192.168.1.3 |
| R0(config)#exit |
| %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 192.168.1.3 port 514 started - CLI initiated |

**Step 3: Verify logging configuration**

Use the command show logging to verify logging has been enabled

```
R0#show loggin
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited,
          0 flushes, 0 overruns, xml disabled, filtering disabled)

ESM: 0 messages dropped
    Trap logging: level informational, 13 message lines logged
        Logging to 192.168.1.3  (udp port 514,  audit disabled,
              authentication disabled, encryption disabled, link up),
              2 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number disabled
              filtering disabled
```

**Step 4: Examine logs of the Syslog Server**




## D. Configure Router 0 to Support SSH

**Step 1: Configure a domain name**

```
R0(config)#ip domain-name example.com
```

**Step 2: Configure users for login to the SSH server**

```
R0(config)#username admin privilege 15 secret sshPass
```

**Step 3: Configure the incoming vty lines**

```
R0(config)#line vty 0 4
R0(config-line)#login local
R0(config-line)#transport input ssh
```

**Step 4: Erase existing key pairs on R3**

```
R0(config-line)#crypto key zeroize rsa
% No Signature RSA Keys found in configuration.
```

**Step 5: Generate the RSA encryption key pair**

```
R0(config)# crypto key generate rsa
The name for the keys will be: R0.example.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]: 1024
    % Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

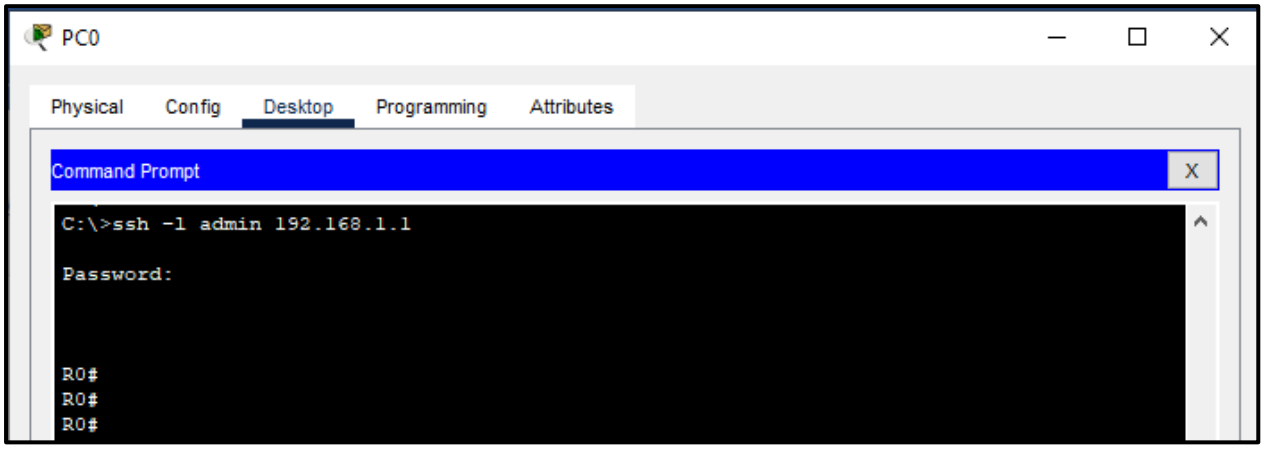
**Step 6: Verify SSH Configuration and Configure Timeout & Retry Parameter**

```
R0(config)#do show ip ssh
*Feb 22 23:50:25.247: %SSH-5-ENABLED: SSH 1.99 has been enabled
SSH Enabled - version 1.99
    Authentication timeout: 120 secs; Authentication retries: 3
```

**Step 7: Configure Timeout & Retry Parameter**

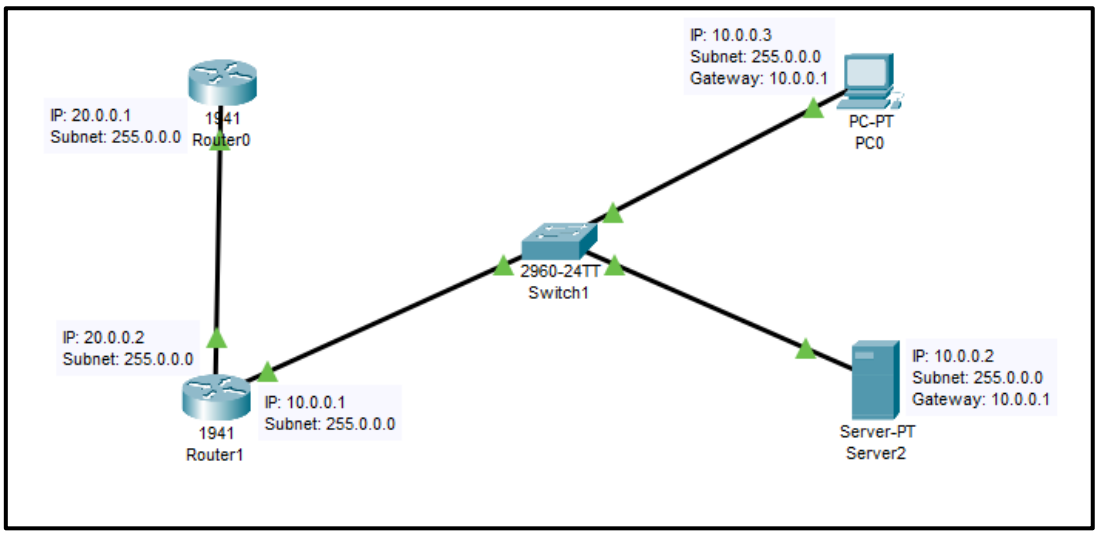
```
R0(config)#ip ssh time-out 90
R0(config)#ip ssh authentication-retries 5
R0(config)#do show ip ssh
SSH Enabled - version 1.99
Authentication timeout: 90 secs; Authentication retries: 5
```

**Step 8: Attempt to Connect Router0 via SSH from PC0**



# Practical 2: Configure AAA Authentication on Cisco Routers

**Topology:**



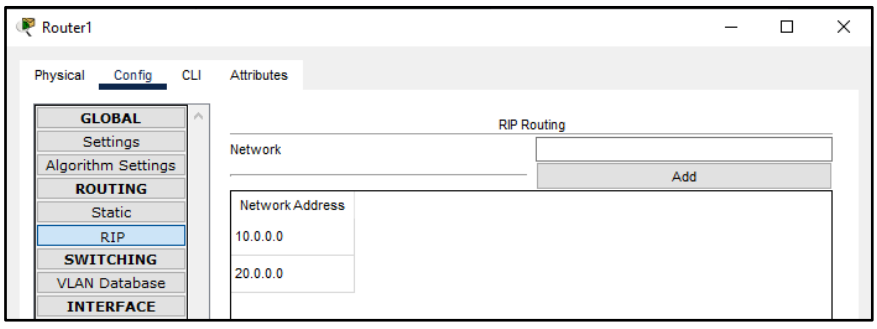

**Address Table:**

| Device | Interface | IP-Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| Router 0 | Gig0/1 | 20.0.0.1 | 255.0.0.0 | NA |
| Router 1 | Gig0/1 | 20.0.0.2 | 255.0.0.0 | |
| Router 1 | Gig0/0 | 10.0.0.1 | 255.0.0.0 | |
| PC0 | Fa0 | 10.0.0.3 | 255.0.0.0 | 10.0.0.1 |
| Server0 | Fa0 | 10.0.0.2 | 255.0.0.0 | 10.0.0.1 |

**Configure AAA Authentication:**

** Ping All the Devices to Verify the Connections **

**Step 0: Configure RIP on Both Routers**




**Step 1: Configure Local Username on Router0**

```
R1(config)#username Admin1 secret admin1pass
```

**Step 2: Configure Local AAA Authentication**

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

**Step 3: Configure the line console to use the defined AAA authentication**

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

**Step 4: Verify the AAA authentication method**

Exit from the Config Terminal and Again Try to access. It should now ask password.

```
User Access Verification

Username: Admin1
Password:
R1>
```

**Step 5: Configure domain name and crypto key for use with SSH.**

```
R1(config)#ip domain-name abcd.com
R1(config)#hostname Admin
Admin(config)#crypto key generate rsa
The name for the keys will be: Admin.abcd.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]

*Mar 1 1:20:47.698: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

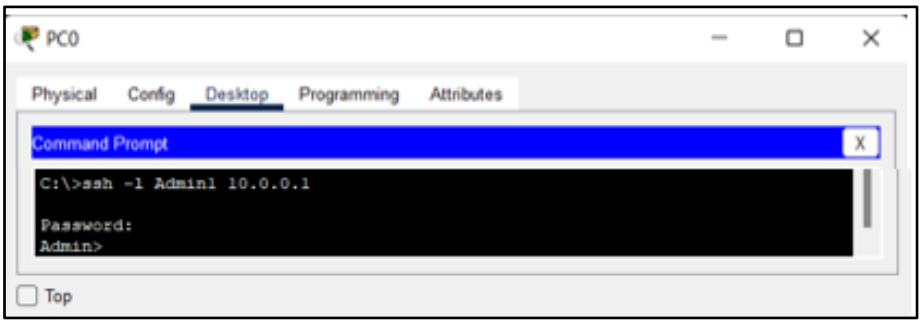
**Step 6: Configure AAA authentication for vty line On Router1**

Configure a named list called SSH-LOGIN to authenticate logins using local AAA

```
Admin(config)#aaa authentication login SSH-LOGIN local
Admin(config)#line vty 0 4
Admin(config-line)#login authentication SSH-LOGIN
Admin(config-line)# transport input ssh
```

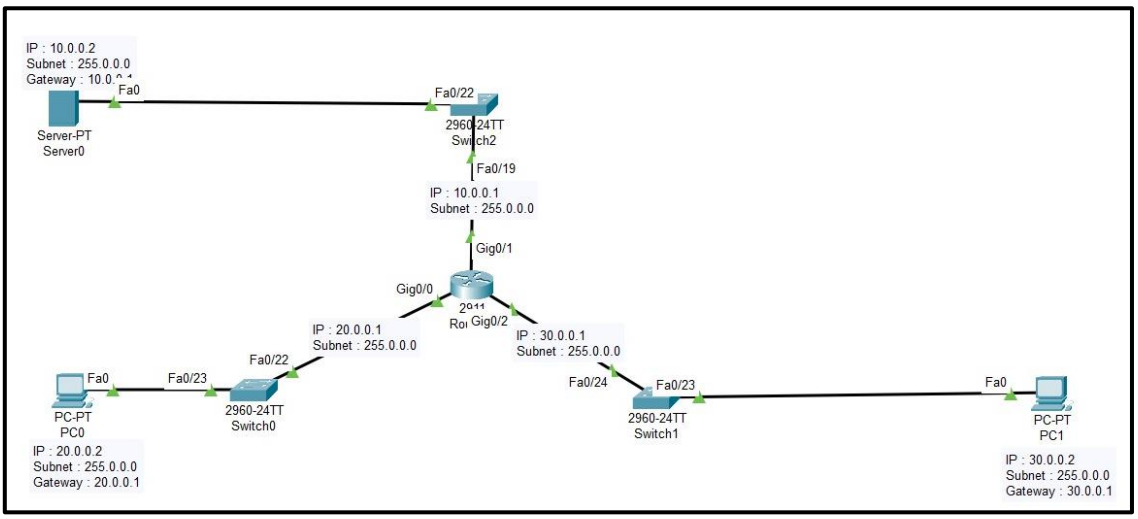
**Step 7: Verify AAA Authentication for Vty Line Via SSH**

Connect Router1 Via SSH From PC0



# Practical 3: Configuring Extended ACLs

**Topology:**

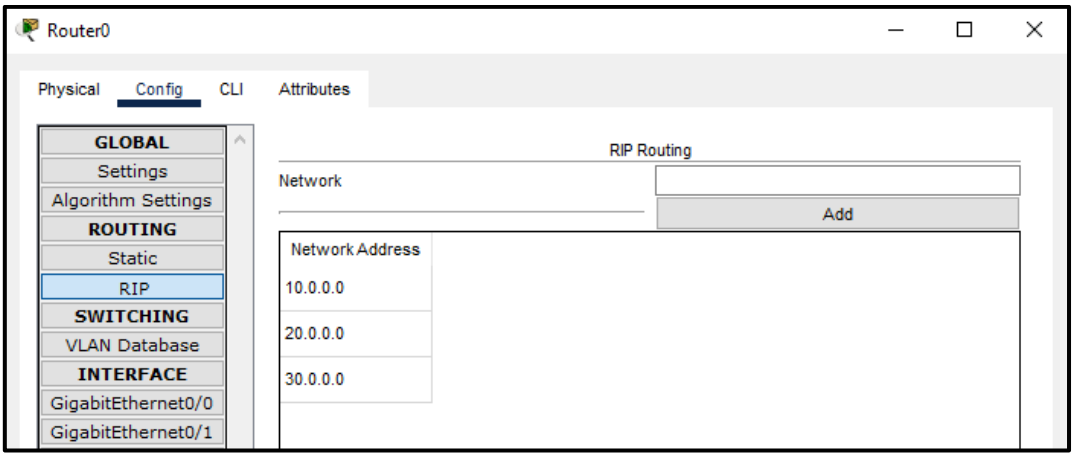



**Address Table:**

| Device | Interface | IP-Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| Router 0 | Gig0/0 | 20.0.0.1 | 255.0.0.0 | NA |
| Router 0 | Gig0/1 | 10.0.0.1 | 255.0.0.0 | |
| Router 0 | Gig0/2 | 30.0.0.1 | 255.0.0.0 | |
| PC0 | Fa0 | 20.0.0.2 | 255.0.0.0 | 20.0.0.1 |
| PC1 | Fa0 | 30.0.0.2 | 255.0.0.0 | 30.0.0.1 |
| Server0 | Fa0 | 10.0.0.2 | 255.0.0.0 | 10.0.0.0 |

** Ping All the Devices to Verify the Connections **

**Configure RIP on Router0**



## A) Configure Extended Number ACL

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

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

Router(config)#access-list 100 permit tcp 20.0.0.2 0.255.255.255 host 10.0.0.2 eq ftp
Router(config)#access-list 100 permit icmp 20.0.0.2 0.255.255.255 host 10.0.0.2
```

### Step 2: Apply the ACL on the Correct Interface to Filter Traffic

Apply ACL on int Gig0/0 & Gig0/2

```
Router(config)#int g0/0
Router(config-if)#ip access-group 100 in
Router(config)#int g0/2
Router(config-if)#ip access-group 100 in
```

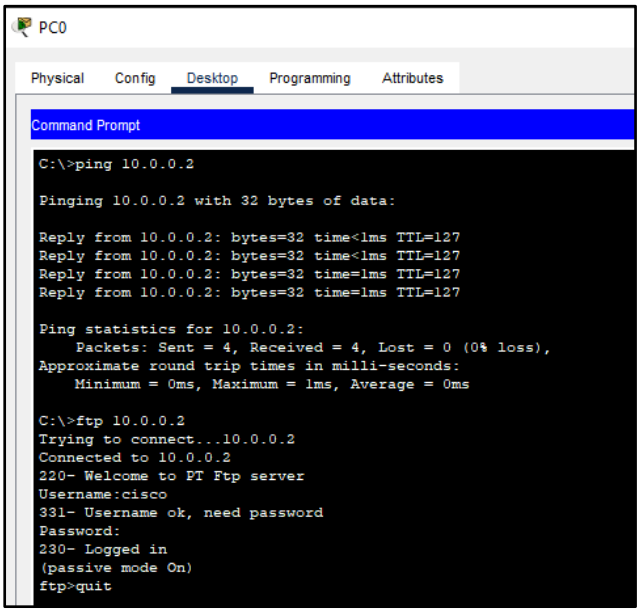
### Step 3: Verify ACL Implementation

Try to Ping Server From PC0 and PC1 & FTP from Both PC to Server

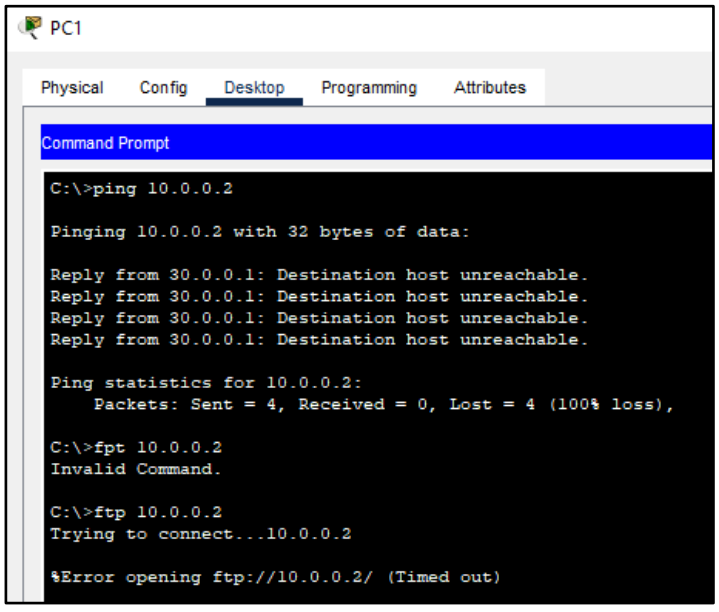
1)Server Should only permit FTP & ICMP to PC0

2) Server should deny all other sources

**Output PC0**




**Output PC1**



## B) Configure Extended Named ACL

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

Filtering WWW Traffic.

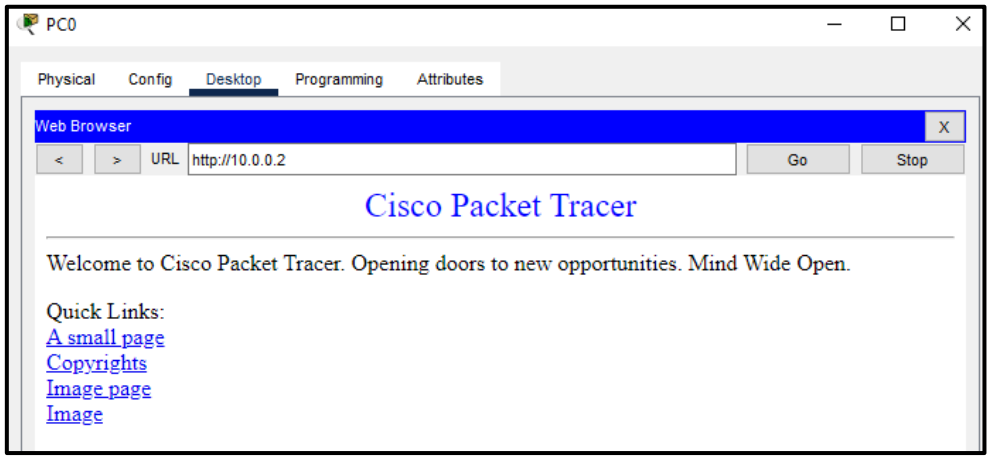
```
Router(config)ip access-list extended HTTP_ONLY
Router(config-ext-nacl)# permit tcp 20.0.0.2 0.255.255.255 host 10.0.0.2 eq www
```

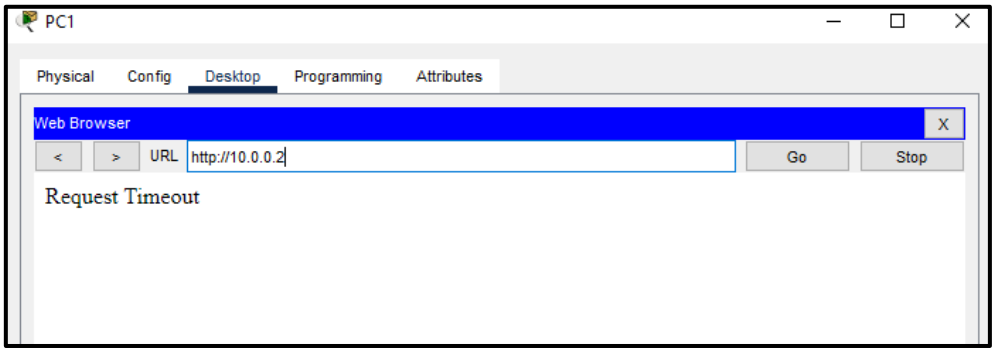
### Step 2: Apply the ACL on the correct interface to filter traffic

```
Router(config)#int g0/0
Router(config-if)#ip access-group HTTP_ONLY in
Router(config)#int g0/2
Router(config-if)#ip access-group HTTP_ONLY in
```

### Step 3: Verify the ACL implementation.

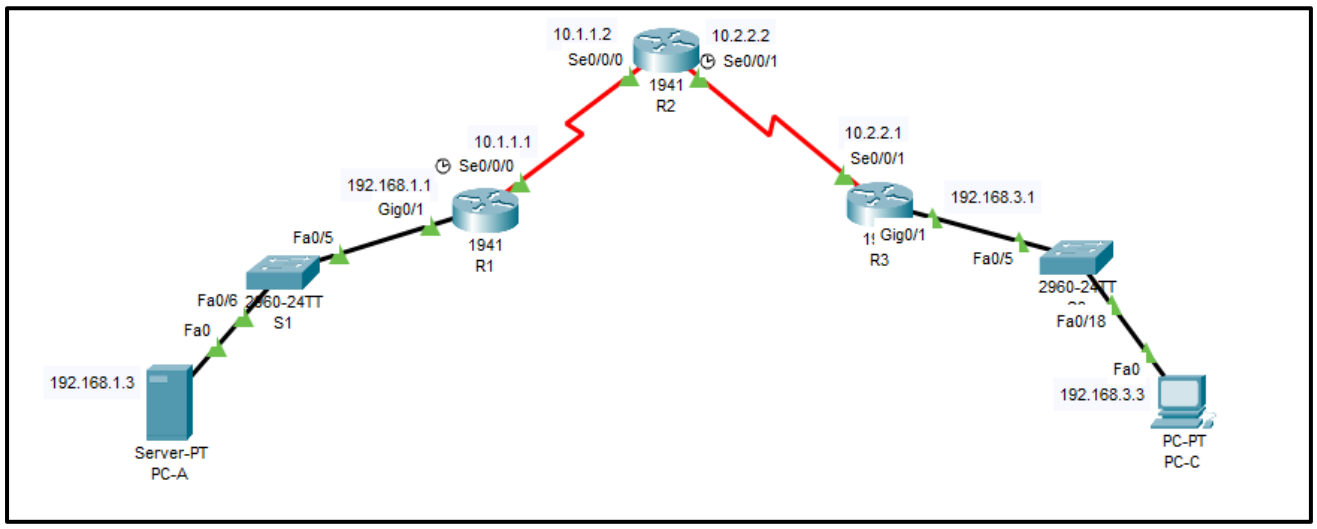
Open the web browser on PC0 and enter the IP address of Server as the URL. The connection should be successful.



# Practical 4: Configure IP ACLs to Mitigate Attacks.

**Topology:**




**Address:**

* Add Loopback Address on Router 2:

```
R2(config)#int loopback 0
R2(config-if)#ip add 192.168.2.1 255.255.255.0
R2(config-if)#no shut
```

| Device | Interface | IP-Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| Router 1 | Gig0/1 | 192.168.1.1 | 255.255.255.0 | NA |
| Router 1 | Se0/0/0 | 10.1.1.1 | 255.255.255.252 | |
| Router 2 | Se0/0/0 | 10.1.1.2 | 255.255.255.252 | |
| Router 2 | Se0/0/1 | 10.2.2.2 | 255.255.255.252 | |
| Router2 | Loopback0 | 192.168.2.1 | 255.255.255.0 | |
| Router 3 | Se0/0/1 | 10.2.2.1 | 255.255.255.252 | |
| Router 3 | Gig0/1 | 192.168.3.1 | 255.255.255.0 | |
| PC-C | Fa0 | 192.168.3.3 | 255.255.255.0 | 192.168.3.1 |
| PC-A (Server) | Fa0 | 192.68.1.3 | 255.255.255.0 | 192.168.1.1 |

** Ping All the Devices to Verify the Connections **

**Step 0: Configure RIP on all 3 routers**

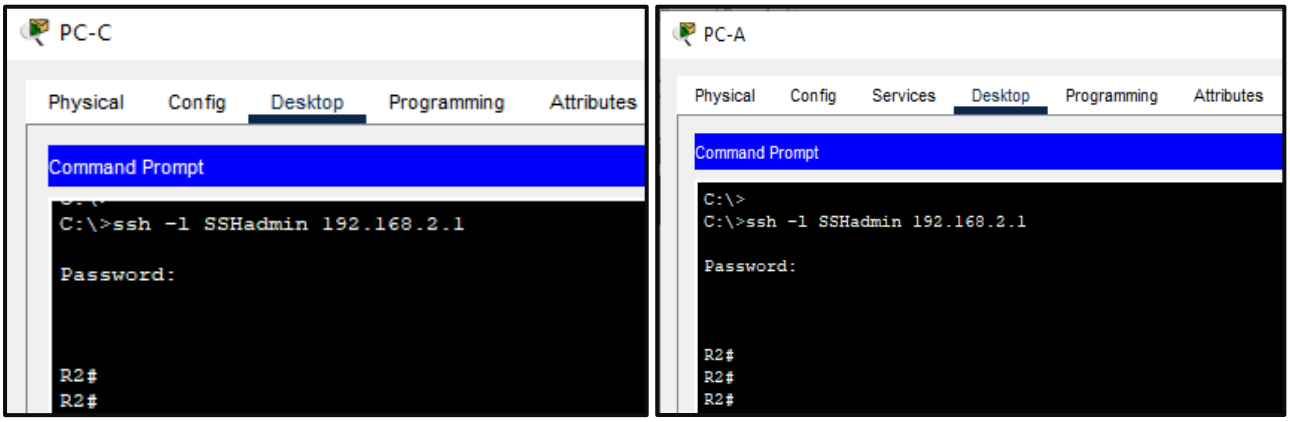
| Router 1 | Router 2 | Router 3 |
|---|---|---|
| Network Address<br>10.0.0.0<br>192.168.1.0 | Network Address<br>10.0.0.0 | Network Address<br>10.0.0.0<br>192.168.3.0 |

Try to Ping From PC-C to PC-A and vice versa to verify RIP

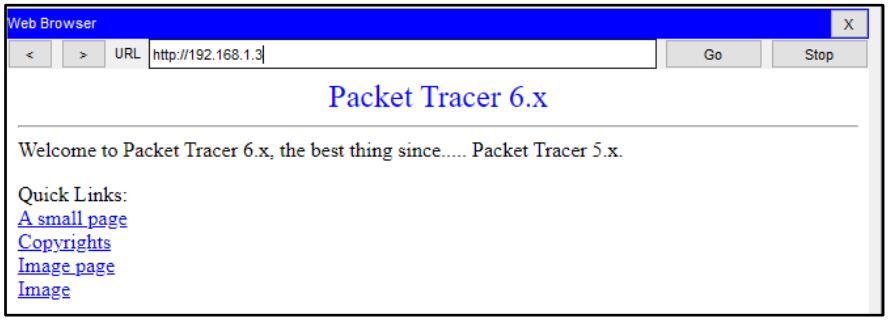
**Part 1: Configure SSH**

**Step 1:** Configure SSH on all the Router with Username SSHadmin & Password ciscosshpa55

**Step 2:** Verify SSH From PC-C to Router2 & PC-A to Router 2




**Step 3:** Open Browser On PC-C and it should have access to Web Page of PC-A (192.168.1.3)




**Part 2: Secure Access To Routers**

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

Use Command access-list to create Numbered ACL on R1,R2,R3

```
R1(config)#access-list 10 permit host 192.168.3.3
```

```
R2(config)#access-list 10 permit host 192.168.3.3
```

```
R3(config)#access-list 10 permit host 192.168.3.3
```

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

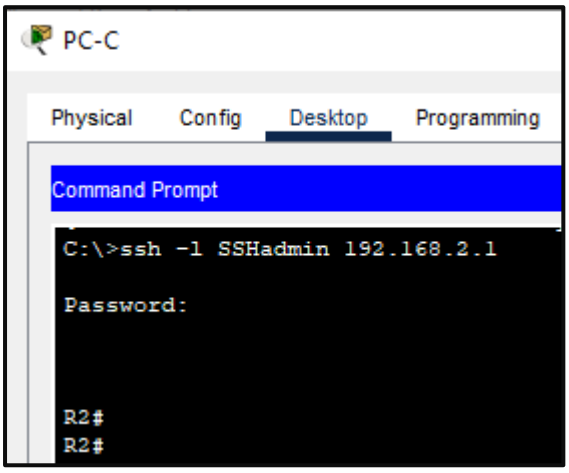
```
R1(config)#line vty 0 4
R1(config-line)#access-class 10 in
```

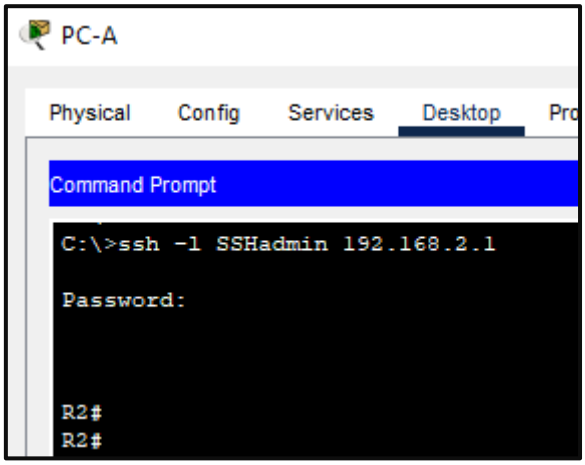
```
R2(config)#line vty 0 4
R2(config-line)#access-class 10 in
```

```
R3(config)#line vty 0 4
R3(config-line)#access-class 10 in
```

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

PC-C establish SSH to 192.168.2.1 but PC-A should fail




**Part 3: Create a Numbered IP ACL 120 on R1**

Create ACL 120 with following

- Permit any outside to access DNS, SMTP, FTP services on Server
- Deny any outside to access HTTPS service on Server
- Permit PC-C to access Router1 Via SSH

**Step 1:** Configure ACL 120 to specifically permit and deny the specified traffic

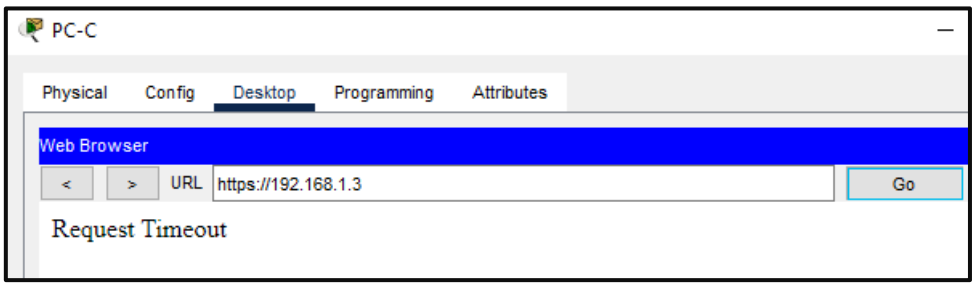
On Router 1:

```
R1(config)#access-list 120 permit udp any host 192.168.1.3 eq domain
R1(config)#access-list 120 permit tcp any host 192.168.1.3 eq smtp
R1(config)#access-list 120 permit tcp any host 192.168.1.3 eq ftp
R1(config)#access-list 120 deny tcp any host 192.168.1.3 eq 443
R1(config)#access-list 120 permit tcp host 192.168.3.3 host 10.1.1.1 eq 22
```

**Step 2: Apply ACL to Interface Se0/0/0 on Router 1**

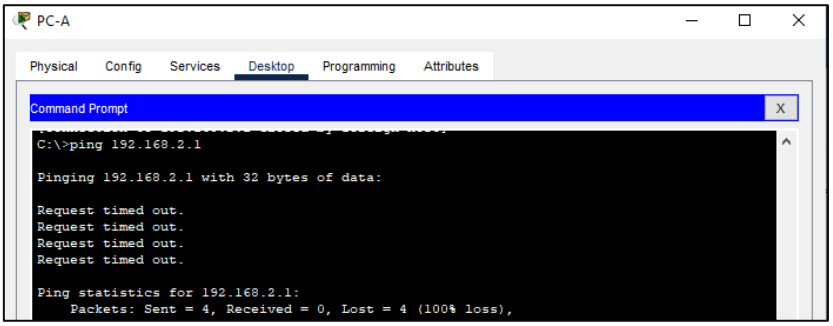
```
R1(config)#int s0/0/0
R1(config-if)#ip access-group 120 in
```

**Step 3:** Verify that PC-C cannot access PC-A via HTTPS using the web browser.



**Part 4: Modify an Existing ACL on R1**

**Step 1:** Verify that PC-A cannot successfully ping the loopback interface on R2, But PC-C can Ping

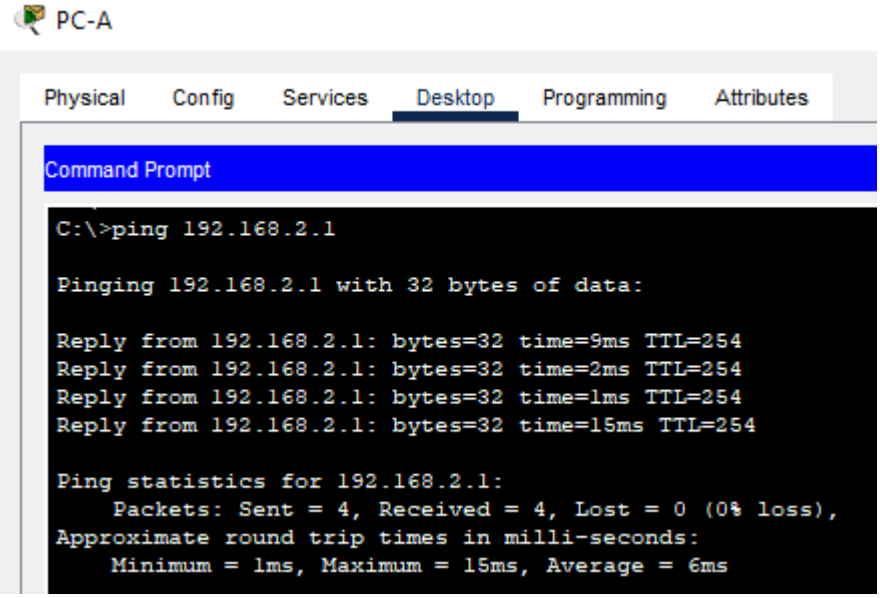



**Step 2:** Make any necessary changes to ACL 120

On Router 1:

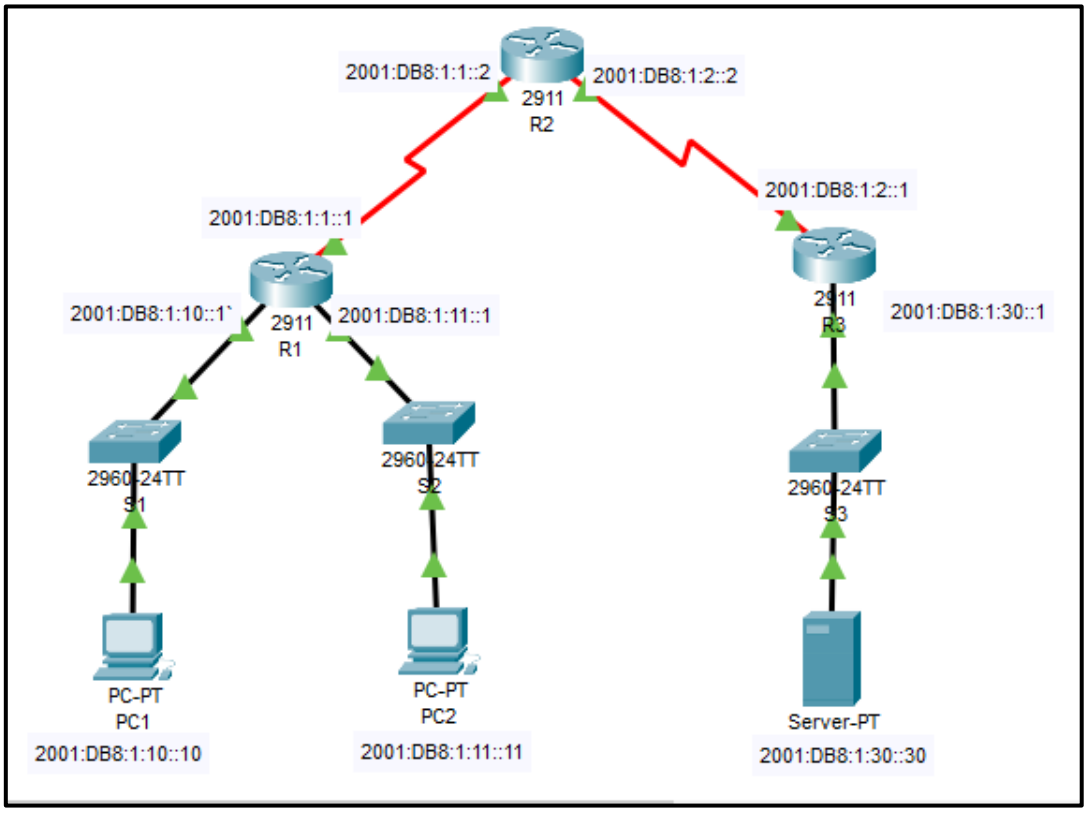
```
R1(config)# access-list 120 permit icmp any any echo-reply
R1(config)# access-list 120 permit icmp any any unreachable
R1(config)# access-list 120 deny icmp any any
R1(config)# access-list 120 permit ip any any
```

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



# Practical 5: Configuring IPv6 ACLs

**Topology:**




**Address Table:**

| Device | Interface | IPv6-Address | Default Gateway |
|---|---|---|---|
| Router 1 | Gig0/0 | 2001:DB8:1:10::1 | NA |
| Router 1 | Gig0/1 | 2001:DB8:1:11::1 | |
| Router 1 | S0/0/0 | 2001:DB8:1:1::1 | |
| Router 2 | S0/0/0 | 2001:DB8:1:1::2 | |
| Router 2 | S0/0/1 | 2001:DB8:1:2::2 | |
| Router 3 | S0/0/1 | 2001:DB8:1:2::1 | |
| Router 3 | Gig0/0 | 2001:DB8:1:30::1 | |
| PC1 | Fa0 | 2001:DB8:1:10::10 | FE80::1 |
| PC2 | Fa0 | 2001:DB8:1:11::11 | FE80::1 |
| Server0 | Fa0 | 2001:DB8:1:30::30 | FE80::3 |

** Ping All the Devices to Verify the Connections **

**Step 1: Configure IPv6 Address & RIP on All Routers**

Configure IPv6 Address on each interface of all the routers & Configure RIP unicast-routing

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ipv6 unicast-routing
Router(config)#int g0/0
Router(config-if)#ipv6 address 2001:DB8:1:10::1/64
Router(config-if)#ipv6 rip ripng enable
Router(config-if)#no shut

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

**Part 1: Configure, Apply, and Verify an IPv6 ACL**

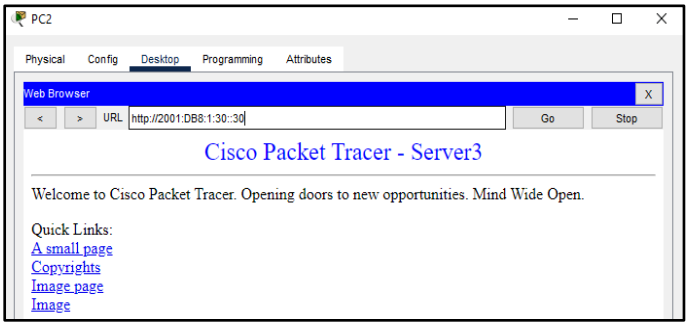
**Step 1: Configure an ACL that will block HTTP and HTTPS access**

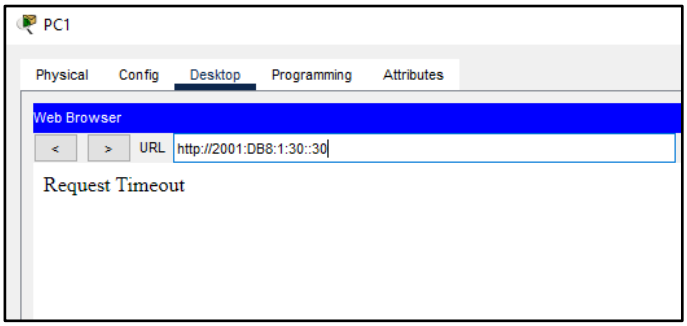
Block HTTP and HTTPS traffic from reaching Server1

```
R1(config)#ipv6 access-list BLOCK_HTTP
R1(config-ipv6-acl)#deny tcp any host 2001:DB8:1:30::30 eq www
R1(config-ipv6-acl)#deny tcp any host 2001:DB8:1:30::30 eq 443
R1(config-ipv6-acl)#permit ipv6 any any
R1(config-ipv6-acl)#exit
R1(config)#int g0/0
R1(config-if)#ipv6 traffic-filter BLOCK_HTTP in
```

**Step 2: Verify ACL**

1) Go to Browser of PC and Type 2001:DB8:1:30::30
2) We have Block Interface g0/0 means PC1 could not access webpage of server3.
3) PC2 can access Web Page of Sever3




**Part 2: Configure, Apply, and Verify a Second IPv6 ACL**

**Step 1: Create an access list to block ICMP**

**On Router 3:**

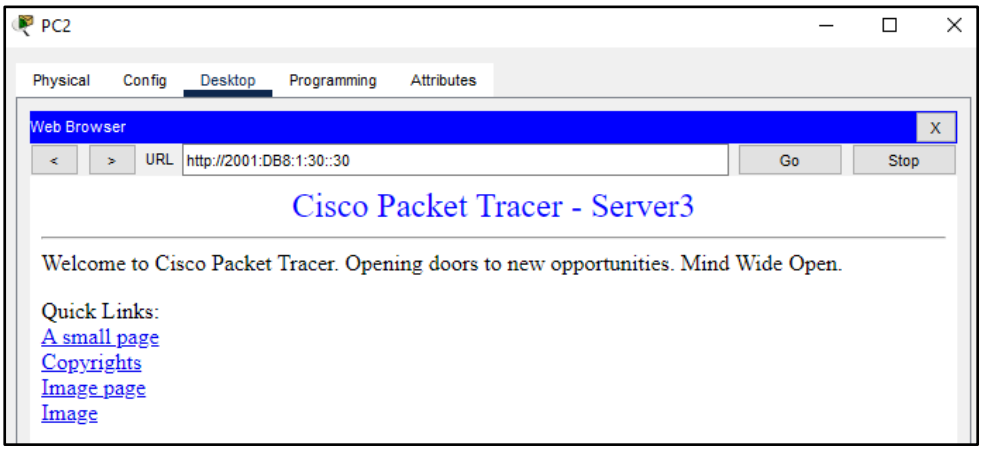
```
R3(config)#ipv6 access-list BLOCK_ICMP
R3(config-ipv6-acl)# deny icmp any any
R3(config-ipv6-acl)# permit ipv6 any any
```

**Step 2: Apply the ACL to the correct interface**

```
R3(config-ipv6-acl)#int g0/0
R3(config-if)#ipv6 traffic-filter BLOCK_ICMP out
```

**Step 3: Verify ACL**

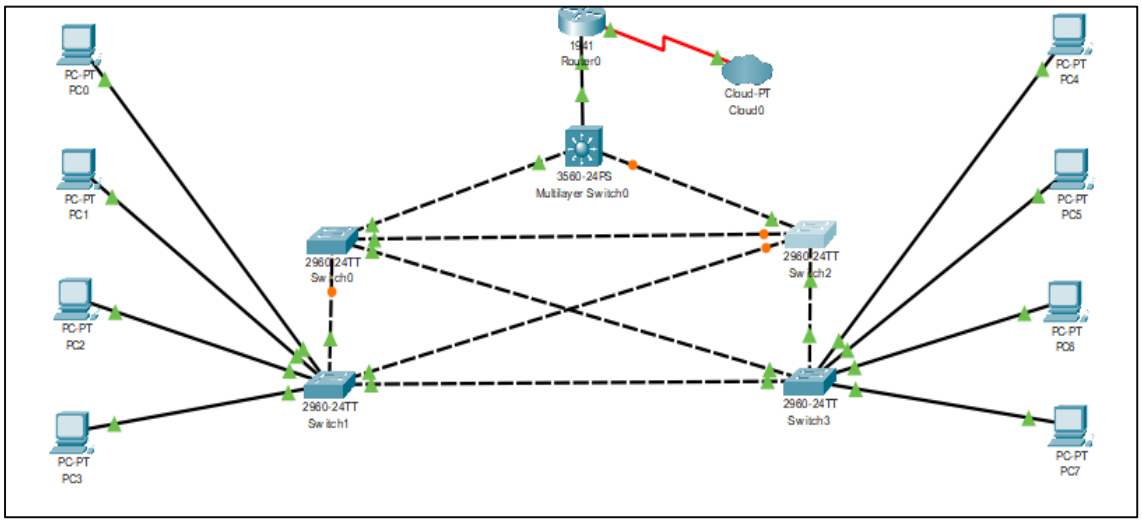
1) All the pc should fail while pinging the server.
2) PC2 Browser can access the Web Page from Server




```
C:\>ping 2001:DB8:1:30::30

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

Reply from 2001:DB8:1:2::1: Destination host unreachable.
Reply from 2001:DB8:1:2::1: Destination host unreachable.
Reply from 2001:DB8:1:2::1: Destination host unreachable.
Reply from 2001:DB8:1:2::1: Destination host unreachable.

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

# Practical 6: Layer 2 Security

**Topology:**




## Part 1: Configure Root Bridge

### Step 1: Assign Central as the primary root bridge.

**On Central Switch**:

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

**Before Assignment**

```
Switch(config)#do show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     0030.F2E9.C95A
              Cost        38
              Port        3(FastEthernet0/3)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     00D0.BAA2.B1A5
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/3            Root FWD 19        128.3    P2p
Fa0/2            Altn BLK 19        128.2    P2p
```

**After Assignment**

```
Switch(config)#spanning-tree vlan 1 root primary
Switch(config)#do show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     00D0.BAA2.B1A5
              This bridge is the root
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    24577  (priority 24576 sys-id-ext 1)
              Address     00D0.BAA2.B1A5
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/3            Desg FWD 19        128.3    P2p
Fa0/2            Desg LSN 19        128.2    P2p
```

### Step 2: Assign Switch-0 as a secondary root bridge

```
Switch(config)# spanning-tree vlan 1 root secondary
```

**Step 3: Verify the spanning-tree configuration.**

```
Switch(config)# do show spanning-tree
```

```
Switch(config)#spanning-tree vlan 1 root secondary
Switch(config)#do show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     0001.C768.3C45
              Cost        19
              Port        4(FastEthernet0/4)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28673  (priority 28672 sys-id-ext 1)
              Address     0006.2ABC.CC66
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/3            Desg LSN 19        128.3    P2p
Fa0/4            Root FWD 19        128.4    P2p
Fa0/2            Desg FWD 19        128.2    P2p
```

## Part 2: Protect against Spanning tree protocol attack

**Step 1: Enable PortFast & BPDU guard on all access ports on Switch 1 and Switch 3**

```
Switch1(config)# interface range f0/1 - 4
Switch1(config-if-range)# spanning-tree portfast
Switch1(config-if-range)# spanning-tree bpduguard enable
```

```
Switch3(config)# interface range f0/1 - 4
Switch3(config-if-range)# spanning-tree portfast
Switch3(config-if-range)# spanning-tree bpduguard enable
```

**Step 2: Enable root guard on Switch 0 and Switch 2**

```
SW-1(config)# interface range f0/23 - 24
SW-1(config-if-range)# spanning-tree guard root
```

## Part 3: Configure port security and disable unused port

**Step 1: On Switch 1 & Switch 3**

```
Switch(config)# interface range f0/1 - 22
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport port-security
Switch(config-if-range)# switchport port-security maximum 2
Switch(config-if-range)# switchport port-security violation shutdown
Switch(config-if-range)# switchport port-security mac-address sticky
```

**Step 2: Verify port security**

```
Switch#show port-security int f0/4
```

```
Switch#
Switch#show port-security int f0/4
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 2
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```

**Step 3: From PC-0 Ping to 10.1.1.11**

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.1.1.11

Pinging 10.1.1.11 with 32 bytes of data:

Reply from 10.1.1.11: bytes=32 time<1ms TTL=128
Reply from 10.1.1.11: bytes=32 time<1ms TTL=128
Reply from 10.1.1.11: bytes=32 time=1ms TTL=128
Reply from 10.1.1.11: bytes=32 time<1ms TTL=128
```

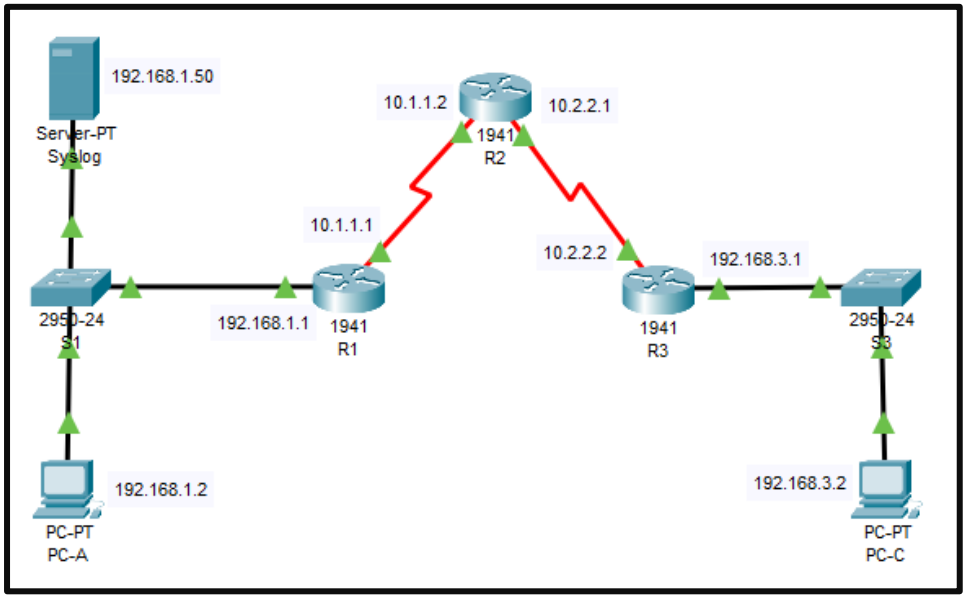
**Step 4: Verify that the switch has learned MAC address**

```
Switch#show port-security int f0/4
```

```
Switch#show port-security int f0/4
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 2
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 0004.9A88.06D1:1
Security Violation Count   : 0
```

# Practical 7: Configure IOS Intrusion Prevention System (IPS) Using the CLI

**Topology:**




**Address:**

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

**Part 1: Enable IOS IPS**

**Step 1**: Enable the Security Technology package

On Router1 :

```
R1(config)#do show version
```

```
Technology Package License Information for Module:'c1900'

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

Configuration register is 0x2102
```

**To Enable Security Technology Package type command:**

```
R1(config)#license boot module c1900 technology-package securityk9        //Accept the License
R1(config)# do reload                                                     //then type yes and press enter to reload
```

**Step 2: Verify network connectivity**

Ping from PC-C to PC-A should be successful and vice versa

PC-C

Physical Config Desktop Programming Attributes

Command Prompt

```
C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=25ms TTL=125
Reply from 192.168.1.2: bytes=32 time=3ms TTL=125
Reply from 192.168.1.2: bytes=32 time=4ms TTL=125
Reply from 192.168.1.2: bytes=32 time=24ms TTL=125

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 25ms, Average = 14ms
```

PC-A

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.3.2

Pinging 192.168.3.2 with 32 bytes of data:

Reply from 192.168.3.2: bytes=32 time=32ms TTL=125
Reply from 192.168.3.2: bytes=32 time=10ms TTL=125
Reply from 192.168.3.2: bytes=32 time=2ms TTL=125
Reply from 192.168.3.2: bytes=32 time=8ms TTL=125
```

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

On Router1 Create directory in Flash using mkdir command :

```
R1#mkdir ipsdir
Create directory filename [ipsdir]?          //Press Enter
Created dir flash:ipsdir
```

**Step 4: Configure the IPS signature storage location**

configure the IPS signature storage location to be the directory you just created:

```
R1(config)#ip ips config location ipsdir
```

**Step 5: Create an IPS rule**

```
R1(config)# ip ips name iosips
```

**Step 6: Enable Logging**

```
R1(config)#logging host 192.168.1.50
R1(config)#service timestamps log datetime msec
```

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

Retire the all signature category with the retired true command

```
R1(config)#ip ips signature-category
R1(config-ips-category)#category all
R1(config-ips-category-action)#retired true
R1(config-ips-category-action)#exit
R1(config-ips-category)#category ios_ips basic
R1(config-ips-category-action)# retired false
R1(config-ips-category-action)#exit
R1(config-ips-category)#exit
Do you want to accept these changes? [confirm]          //Press Enter
Applying Category configuration to signatures ...
```

**Step 8: Apply the IPS rule to an interface.**

Apply IPS rule with command: ip ips name direction

```
R1(config)#interface g0/1
R1(config-if)#ip ips iosips out
```

## Part 2: Modify the Signature

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

```
R1(config)#ip ips signature-definition
R1(config-sigdef)# signature 2004 0
R1(config-sigdef-sig)#status
R1(config-sigdef-sig-status)# retired false
R1(config-sigdef-sig-status)#enabled true
R1(config-sigdef-sig-status)#exit
R1(config-sigdef-sig)#engine
R1(config-sigdef-sig-engine)#event-action produce-alert
R1(config-sigdef-sig-engine)#event-action deny-packet-inline
R1(config-sigdef-sig-engine)#exit
R1(config-sigdef-sig)#exit
R1(config-sigdef)#exit
Do you want to accept these changes? [confirm]          //press enter
```

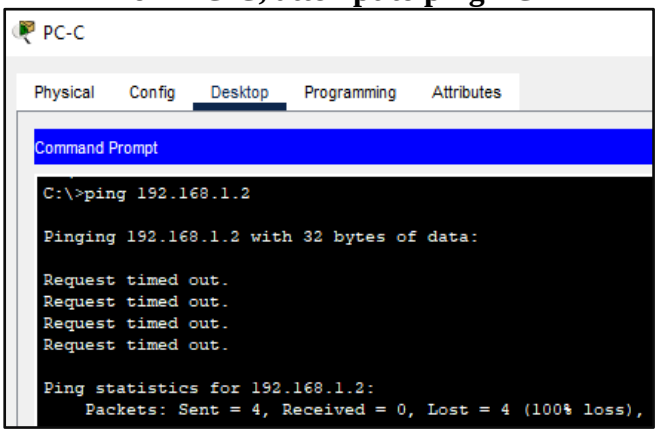
**Step 2: Use show commands to verify IPS**

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

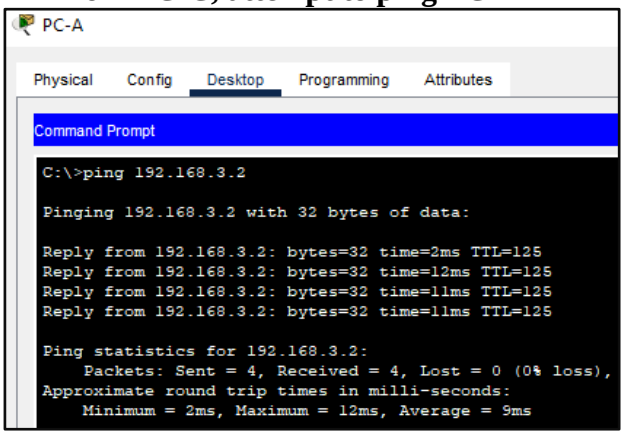
```
IPS Signature Status
    Total Active Signatures: 1
    Total Inactive Signatures: 0

IPS Packet Scanning and Interface Status
    IPS Rule Configuration
      IPS name iosips
    IPS fail closed is disabled
    IPS deny-action ips-interface is false
    Fastpath ips is enabled
    Quick run mode is enabled
    Interface Configuration
      Interface GigabitEthernet0/1
        Inbound IPS rule is not set
        Outgoing IPS rule is iosips
```

**Step 3: Verify that IPS is working properly**

**From PC-C, attempt to ping PC-A**

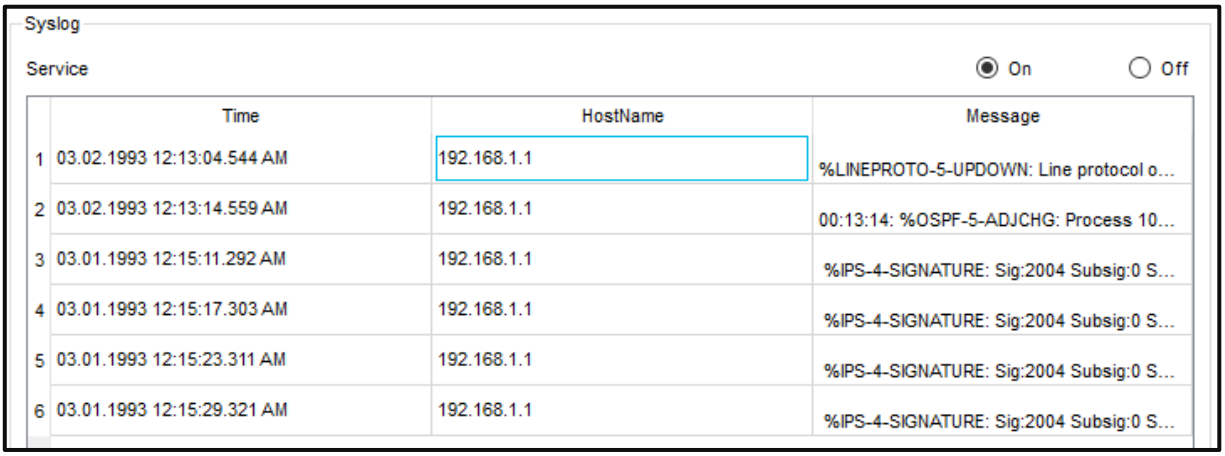



**From PC-C, attempt to ping PC-A**




Ping From PC-C to PC-A failed due to IPS rule for event-action of an echo request was set to "deny packet inline"

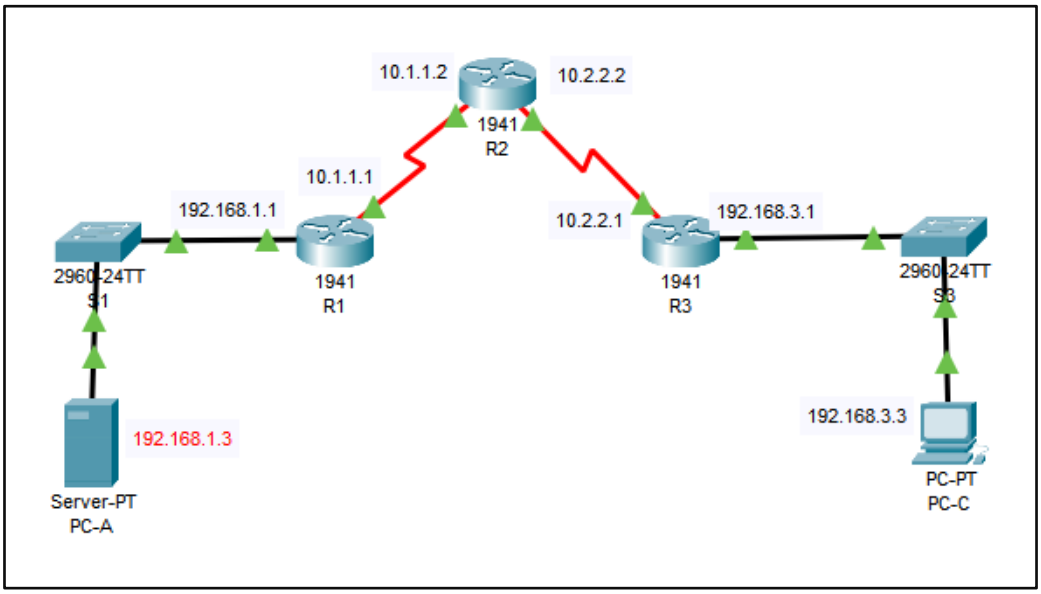
**Step 4: View the syslog messages**




| | Time | HostName | Message |
|---|---|---|---|
| 1 | 03.02.1993 12:13:04.544 AM | 192.168.1.1 | %LINEPROTO-5-UPDOWN: Line protocol o... |
| 2 | 03.02.1993 12:13:14.559 AM | 192.168.1.1 | 00:13:14: %OSPF-5-ADJCHG: Process 10... |
| 3 | 03.01.1993 12:15:11.292 AM | 192.168.1.1 | %IPS-4-SIGNATURE: Sig:2004 Subsig:0 S... |
| 4 | 03.01.1993 12:15:17.303 AM | 192.168.1.1 | %IPS-4-SIGNATURE: Sig:2004 Subsig:0 S... |
| 5 | 03.01.1993 12:15:23.311 AM | 192.168.1.1 | %IPS-4-SIGNATURE: Sig:2004 Subsig:0 S... |
| 6 | 03.01.1993 12:15:29.321 AM | 192.168.1.1 | %IPS-4-SIGNATURE: Sig:2004 Subsig:0 S... |

# Practical 8: Configuring a Zone-Based Policy Firewall (ZPF)

**Topology:**




Address:

| Device | Interface | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| R1 | G0/1 | 192.168.1.1 | 255.255.255.0 | N/A |
| | S0/0/0 (DCE) | 10.1.1.1 | 255.255.255.252 | N/A |
| R2 | S0/0/0 | 10.1.1.2 | 255.255.255.252 | N/A |
| | S0/0/1 (DCE) | 10.2.2.2 | 255.255.255.252 | N/A |
| R3 | G0/1 | 192.168.3.1 | 255.255.255.0 | N/A |
| | S0/0/1 | 10.2.2.1 | 255.255.255.252 | N/A |
| PC-A | NIC | 192.168.1.3 | 255.255.255.0 | 192.168.1.1 |
| PC-C | NIC | 192.168.3.3 | 255.255.255.0 | 192.168.3.1 |

## Part 1: Create the Firewall Zones on R3

**Step 1**: Enable the Security Technology package

On Router 3 :

**R3(config)#do show version**

```
Technology Package License Information for Module:'c1900'

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

Configuration register is 0x2102
```

**To Enable Security Technology Package type command:**

```
R3(config)#license boot module c1900 technology-package securityk9        //Accept the License
R3(config)# do reload                                                     //then type yes and press enter to reload
```

**Step 2: Create an internal zone & external zone**

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

## Part 2: Identify Traffic Using a Class-Map

**Step 1: Create an ACL that defines internal traffic**

Use the access-list command to create extended ACL 101 to permit all IP protocols from the 192.168.3.0/24 source network to any destination

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

**Step 2: Create a class map referencing the internal traffic ACL**

```
R3(config)#access-list 101 permit ip 192.168.3.0 0.0.0.255 any
R3(config)#class-map type inspect match-all IN-NET-CLASS-MAP
R3(config-cmap)#match access-group 101
R3(config-cmap)#exit
```

## Part 3: Specify Firewall Policies

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

```
R3(config)# policy-map type inspect IN-2-OUT-PMAP
```

**Step 2**: Specify a class type of inspect and reference class map IN-NET-CLASS-MAP

```
R3(config-pmap)#class type inspect IN-NET-CLASS-MAP
```

**Step 3**: Specify the action of inspect for this policy map.

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

## Part 4: Apply Firewall Policies

**Step 1:** Create a pair of zones

```
zone-pair security IN-2-OUT-ZPAIR source IN-ZONE destination OUT-ZONE
```

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

```
R3(config-sec-zone-pair)#service-policy type inspect IN-2-OUT-PMAP
```

**Step 3:** Assign interfaces to the appropriate security zones

```
R3(config)#interface g0/1
R3(config-if)# zone-member security IN-ZONE
R3(config-if)#exit
R3(config)#interface s0/0/1
R3(config-if)# zone-member security OUT-ZONE
R3(config-if)#exit
```

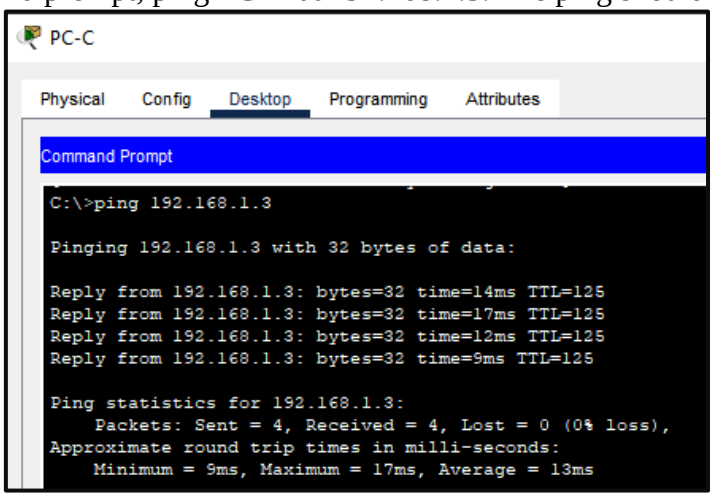
**Step 4:** Copy the running configuration to the startup configuration

## Part 5: Test Firewall Functionality from IN-ZONE to OUT-ZONE

Verify that internal hosts can still access external resources after configuring the ZPF.

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

From the PC-C command prompt, ping PC-A at 192.168.1.3. The ping should succeed.




## Part 6: Test Firewall Functionality from OUT-ZONE to IN-ZONE

Verify that external hosts CANNOT access internal resources after configuring the ZPF.

**Step 1: From the PC-A server command prompt, ping PC-C.**

From the PC-A command prompt, ping PC-C at 192.168.3.3. The ping should fail