



RAJALAKSHMI ENGINEERING COLLEGE

**An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2025-2026 ODD SEMESTER

CS19P14 INFORMATION AND SECURITY MANAGEMENT LAB

LAB MANUAL

FOURTH YEAR

SEVENTH SEMESTER

Vision

1. To be an institution of excellence in Engineering, Technology and Management Education & Research.
2. To provide competent and ethical professionals with a concern for society.

Mission

1. To impart quality technical education imbued with proficiency and humane values.
2. To provide right ambience and opportunities for the students to develop into creative, talented and globally competent professionals.
3. To promote research and development in technology and management for the benefit of the society.

S.No.	Date	Experiment	No of Hrs	Signature
1		Implementation to gather information from any PC's connected to the LAN using who is, port scanners, network scanning, Angry IP scanners etc.	6	
2		Implementation of Steganography	4	
3		Implementation of Mobile Audit and generate the report of the existing Artifacts	4	
4		Implementation of IT Audit, malware analysis and Vulnerability assessment and generate the report.	4	
5		Implementation of Cyber Forensics tools for Disk Imaging, Data acquisition, Data extraction and Data Analysis and recovery.	4	
6		Perform mobile analysis in the form of retrieving call logs , SMS log ,all contacts list using the forensics tool like SAFT.	4	
7		Implementation to identify web vulnerabilities, using OWASP project.	4	
Contact Hours:			30	
Total Periods:			30	
Hardware		Stand Alone Systems PC With Fedora		
Software		Python		
Tool		SNORT, SAFT		

CO - PO – PSO matrices of course

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CS19P14.1	3	2	2	1	1	3	-	3	-	-	-	2	2	1	1
CS19P14.2	3	2	2	2	1	2	2	2	-	-	-	-	2	1	1
CS19P14.3	3	2	2	2	2	2	2	1	-	-	-	2	3	3	3
CS19P14.4	3	2	2	2	3	2	2	2	-	-	-	2	3	3	2
CS19P14.5	3	3	2	2	3	2	2	1	-	-	-	2	3	3	3
Average	3	2.2	2	1.8	2	2.2	2.0	1.8	-	-	-	2.0	2.6	2.2	2

Note: Enter correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

If there is no correlation, put “-”

Experiment -1

Date:

Basic Firewall Configuration in Cisco Packet Tracer

Steps to Configure and Verify Firewall in Cisco Packet Tracer:

Step 1: First, open the Cisco packet tracer desktop and select the devices given below:

S.NO	Device	Model Name	Quantity
1.	PC	PC	3
2.	server	PT-Server	1
3.	switch	PT-Switch	1

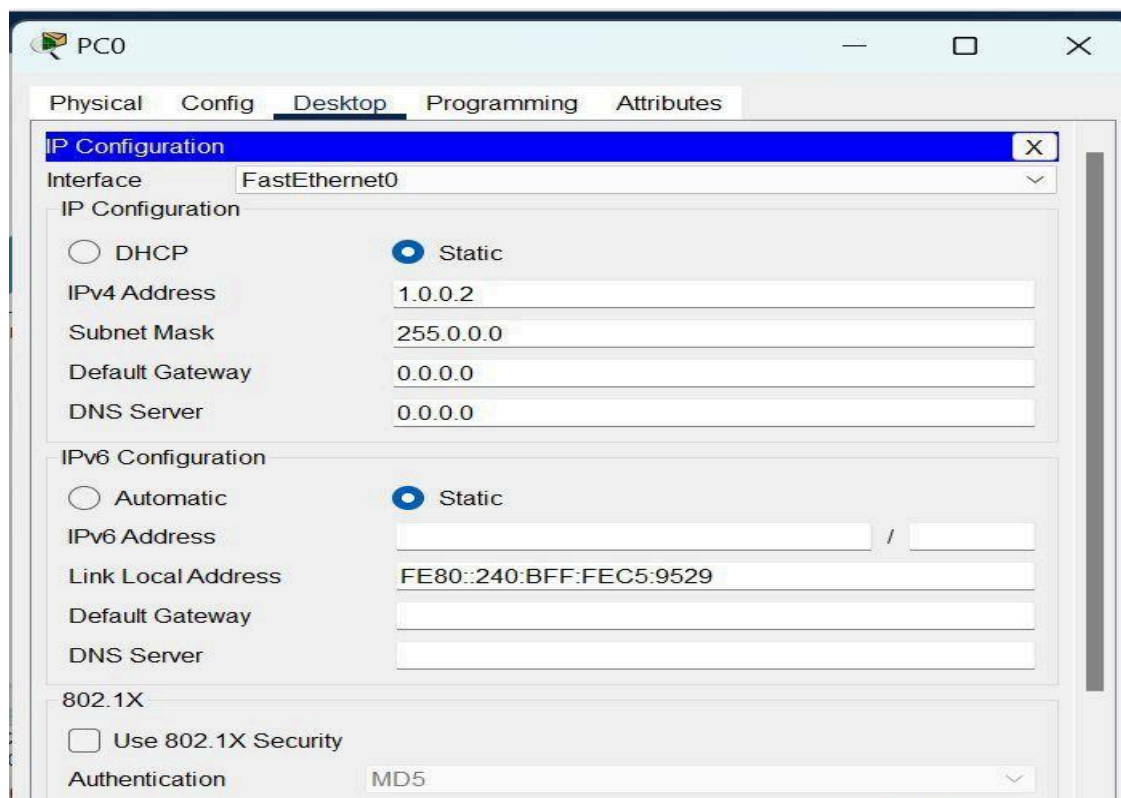
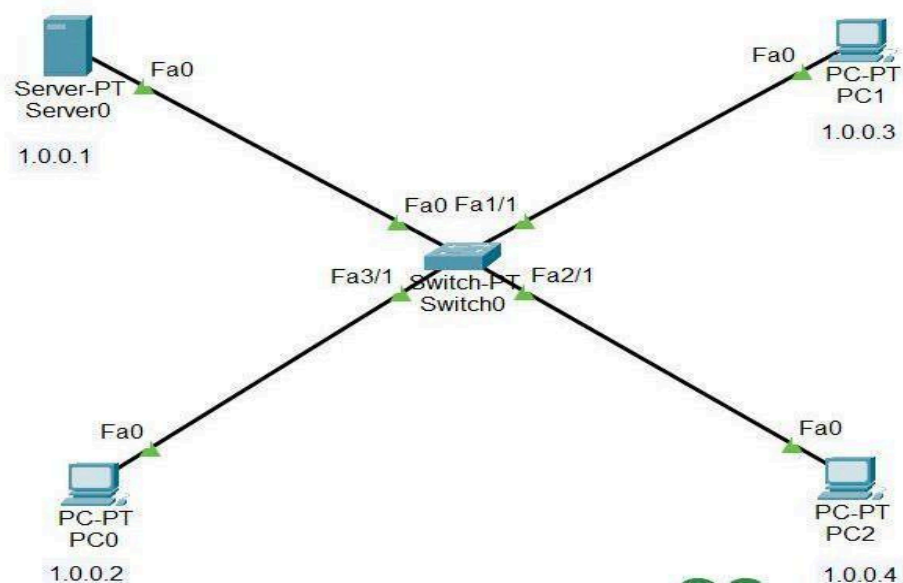
IP Addressing Table:

S.NO	Device	IPv4 Address	Subnet Mask
1.	Server	1.0.0.1	255.0.0.0
2.	PC0	1.0.0.2	255.0.0.0
3.	PC1	1.0.0.3	255.0.0.0
4.	PC2	1.0.0.4	255.0.0.0

- Then, create a network topology as shown below the image.
- Use an Automatic connecting cable to connect the devices with others.

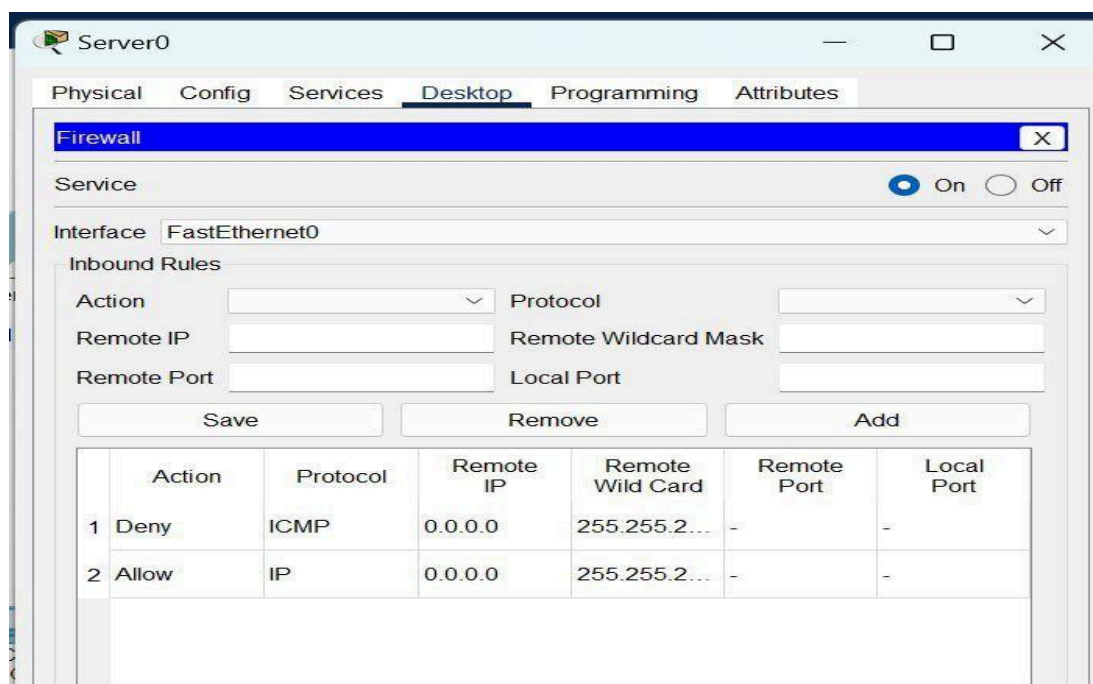
Step 2: Configure the PCs (hosts) and server with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.
- Repeat the same procedure with the server



Step 3: Configuring the firewall in a server and blocking packets and allowing web browser.

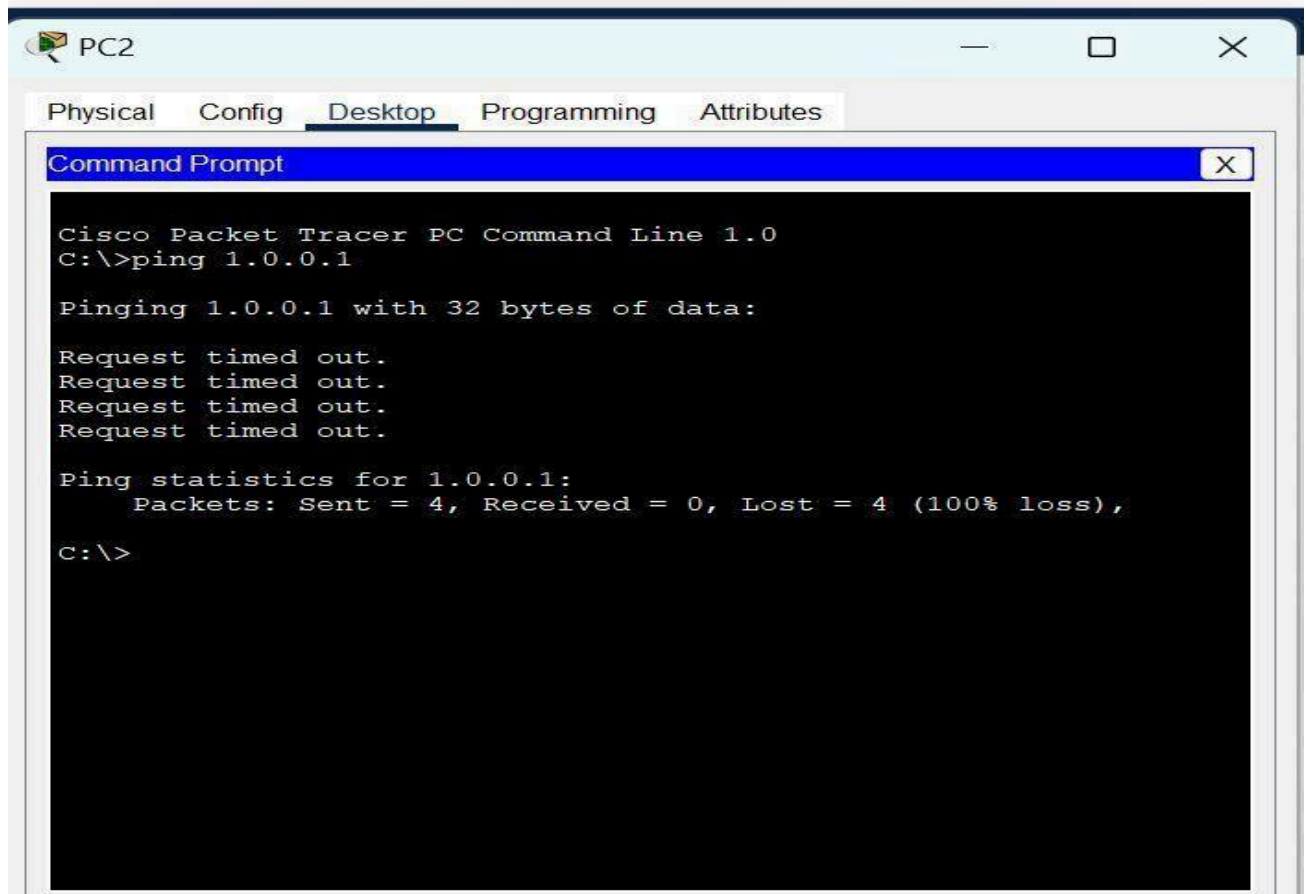
- ❑ Click on server0 then go to the desktop.
- ❑ Then click on firewall IPv4.
- ❑ Turn on the services.
- ❑ First, Deny the ICMP protocol and set remote IP to 0.0.0.0 and Remote wildcard mask to 255.255.255.255.
- ❑ Then, allow the IP protocol and set remote IP to 0.0.0.0 and Remote wildcard mask to 255.255.255.255.
- ❑ And add them.



Step 4: Verifying the network by pinging the IP address of any PC.

- ❑ We will use the ping command to do so.
- ❑ First, click on PC2 then Go to the command prompt.
- ❑ Then type ping <IP address of targeted node>.

- We will ping the IP address of the server0.
- As we can see in the below image we are getting no replies which means the packets are blocked.



Check the web browser by entering the IP address in the URL.

- Click on PC2 and go to desktop then web browser.

OUTPUT:

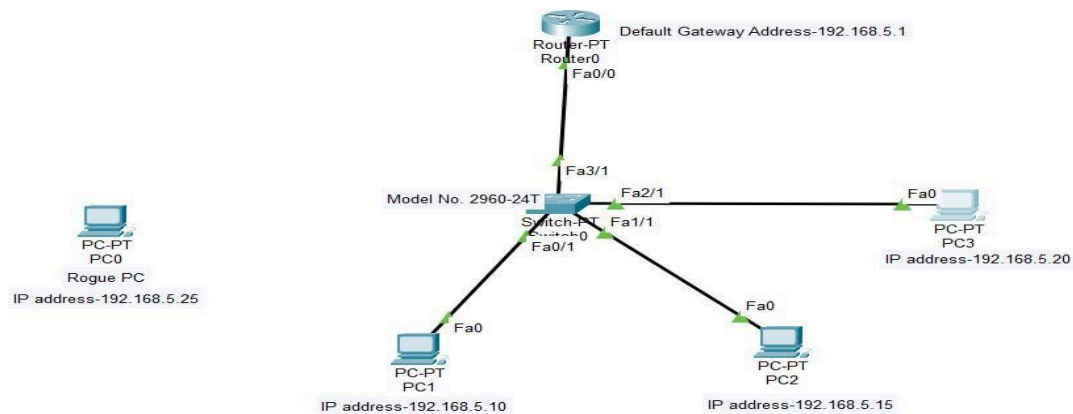
RESULT:

Experiment -2

Date:

Configure Port Security in Cisco Packet Tracer

Step-1 Build the network single network topology in packet tracer.



Step-2 Give wired connections for all the devices of PC1, PC2, PC3 and Router.

Step-3 Give IP addresses for PC1, PC2, PC3 and Router with default gateway address.

Step-3 Click on PC1, goto command prompt and ping the IP address of other PCs.

Step-4 Repeat this step-3 for other two PCs as PC2 and PC3.

```
PC1
Physical Config Desktop Programming Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.5.15

Pinging 192.168.5.15 with 32 bytes of data:

Reply from 192.168.5.15: bytes=32 time=20ms TTL=128
Reply from 192.168.5.15: bytes=32 time<1ms TTL=128
Reply from 192.168.5.15: bytes=32 time<1ms TTL=128
Reply from 192.168.5.15: bytes=32 time<1ms TTL=128

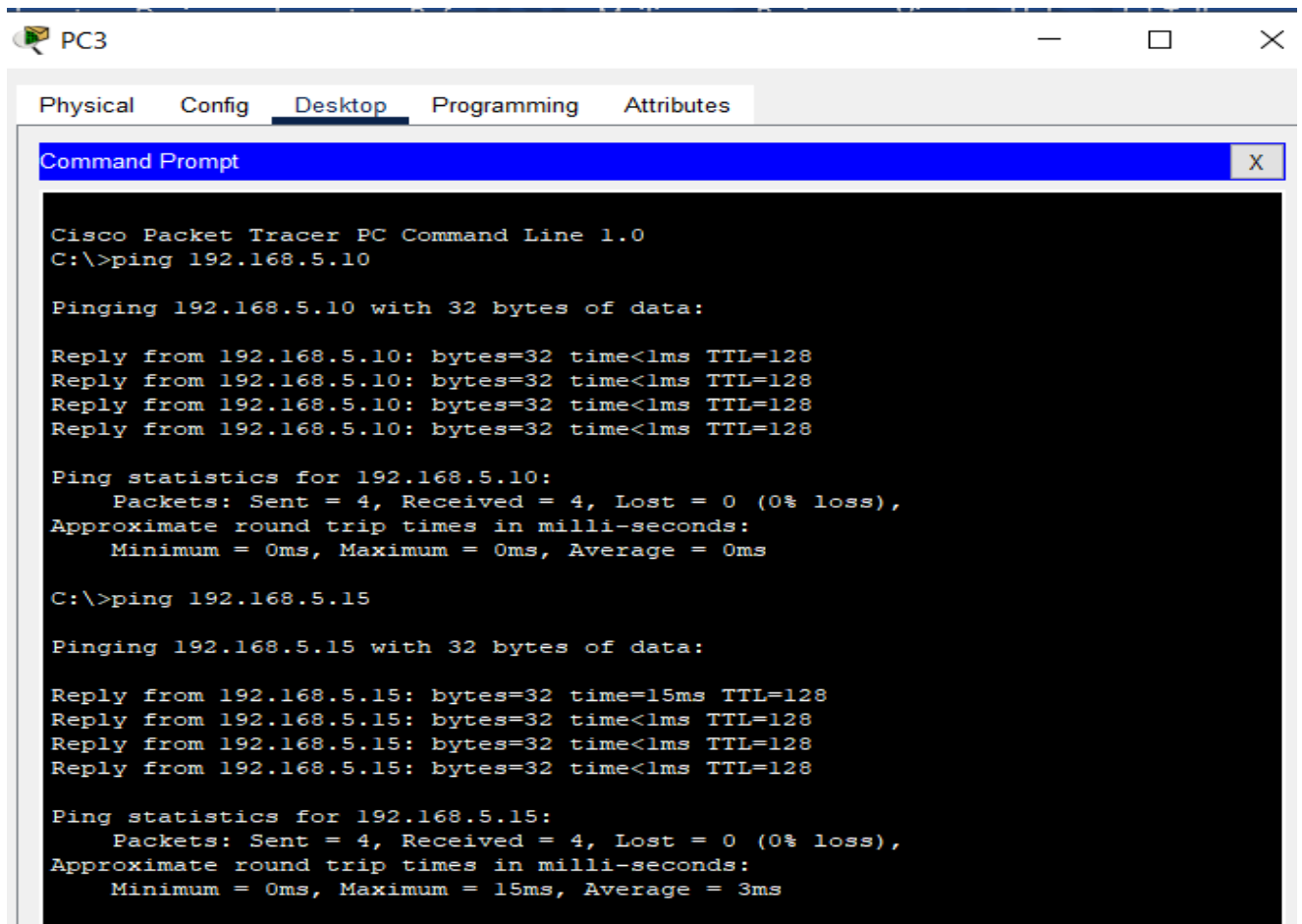
Ping statistics for 192.168.5.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 20ms, Average = 5ms

C:\>ping 192.168.5.20

Pinging 192.168.5.20 with 32 bytes of data:

Reply from 192.168.5.20: bytes=32 time<1ms TTL=128
Reply from 192.168.5.20: bytes=32 time<1ms TTL=128
Reply from 192.168.5.20: bytes=32 time<1ms TTL=128
Reply from 192.168.5.20: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.5.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



Step-5 Click on Switch 0

Goto CLI command

#enable

#configure terminal

#int fa0/1

#switchport mode access

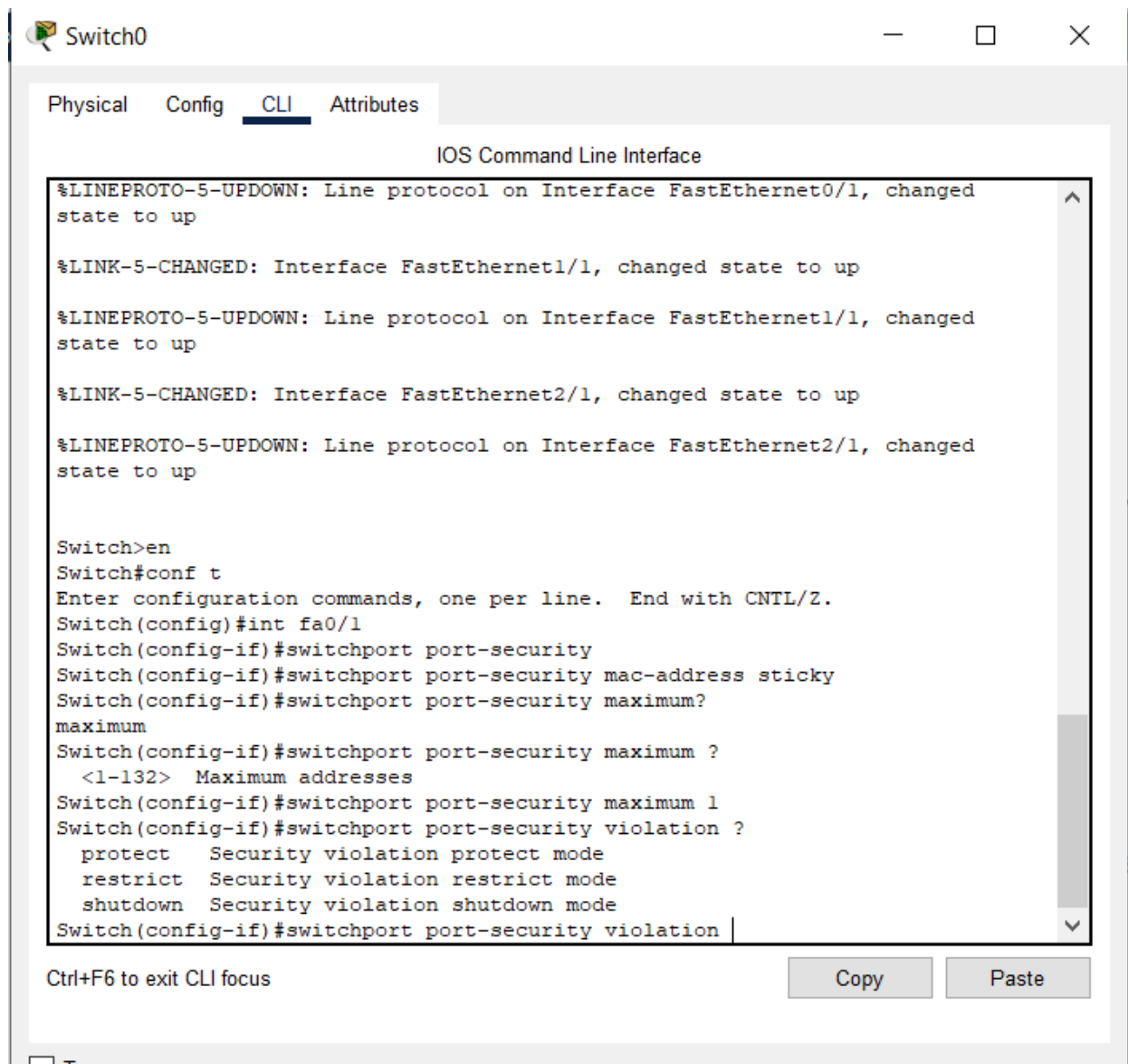
#switchport port-security

#switchport port-security mac-address sticky

#switchport port-security maximum ?

#switchport port-security maximum 1

#switchport port-security violation ?



The screenshot shows a network switch window titled "Switch0" with tabs for Physical, Config, CLI, and Attributes. The CLI tab is active, displaying the "IOS Command Line Interface". The terminal output shows several status messages for interfaces FastEthernet0/1, FastEthernet1/1, and FastEthernet2/1, indicating line protocol and link state changes. The configuration process is shown starting from the user prompt "Switch>en" to enter enable mode, then "Switch#conf t" to enter configuration mode. The user configures interface "fa0/1" with "switchport port-security", "switchport port-security mac-address sticky", and "switchport port-security maximum". The prompt "Switch(config-if)#switchport port-security maximum ?" is shown with the user inputting "<1-132>". The user then sets "switchport port-security maximum 1" and "switchport port-security violation ?". The prompt "Switch(config-if)#switchport port-security violation ?" is shown with the user inputting "protect". The terminal output shows the configuration commands and the resulting configuration for interface fa0/1. The window includes a "Ctrl+F6 to exit CLI focus" message and "Copy" and "Paste" buttons.

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up

%LINK-5-CHANGED: Interface FastEthernet1/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, changed
state to up

%LINK-5-CHANGED: Interface FastEthernet2/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed
state to up

Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fa0/1
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security maximum?
maximum
Switch(config-if)#switchport port-security maximum ?
<1-132> Maximum addresses
Switch(config-if)#switchport port-security maximum 1
Switch(config-if)#switchport port-security violation ?
protect Security violation protect mode
restrict Security violation restrict mode
shutdown Security violation shutdown mode
Switch(config-if)#switchport port-security violation
```

A Switch port Security is a **network security** feature that associates specific MAC addresses of devices (such as PCs) with specific interfaces on a switch. This will enable you to restrict access to a given switch interface so that only the authorized devices can use it. If an unauthorized device is connected to the same port, you can define the action that the switch will take, such as discarding the traffic, sending an alert, or shutting down the port. There are 3 types of port security violations namely protect, restrict and shutdown mode. Each mode has been seen in step by step process.

Step-6 Now, from the 3 port-security violation we will first process for **Shutdown Mode**.

```
#switchport port-security violation shutdown
```

```
#exit
```

Now, for the next interface connection access the shutdown mode.

```
#int fa1/1
```

```
#switchport mode access
```

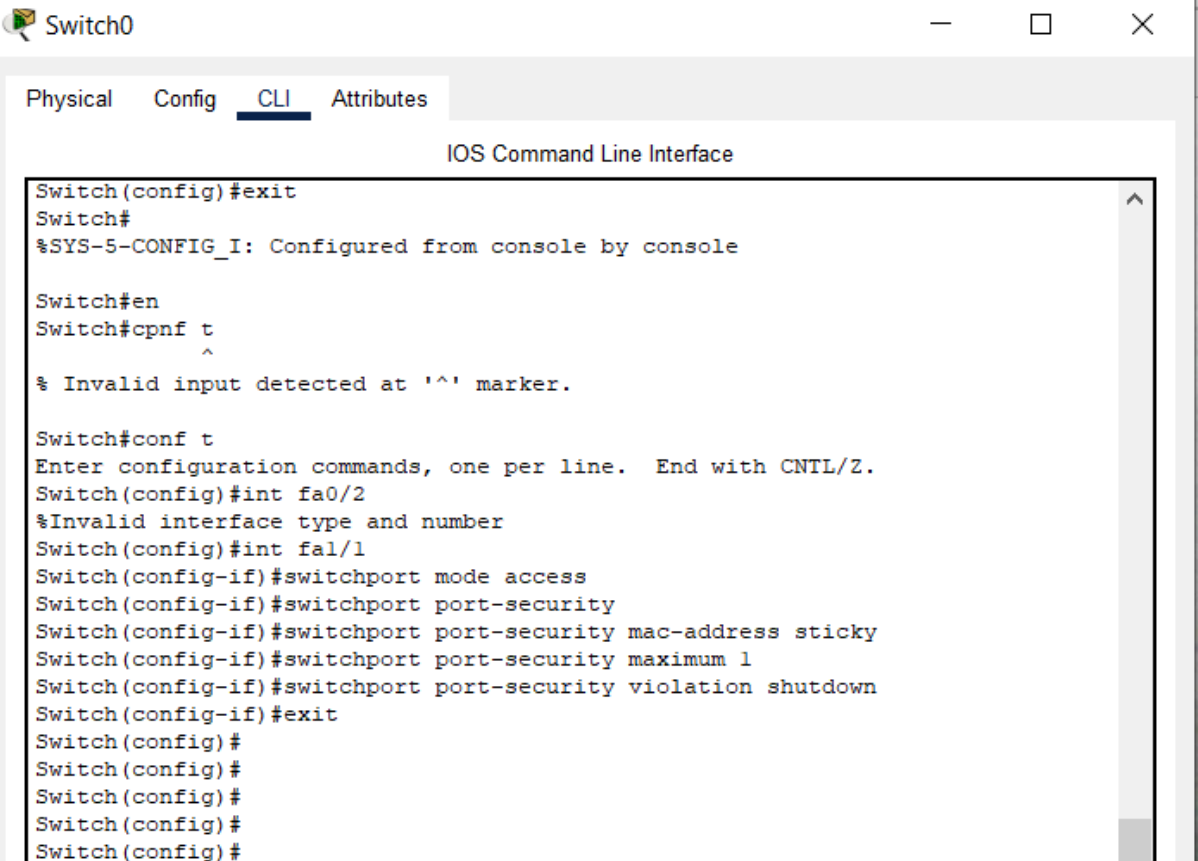
```
#switchport port-security
```

```
#switchport port-security mac-address sticky
```

```
#switchport port-security maximum 1
```

```
#switchport port-security violation shutdown
```

```
#exit
```



```
Switch0
Physical Config CLI Attributes
IOS Command Line Interface
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#en
Switch#cpnf t
      ^
% Invalid input detected at '^' marker.

Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fa0/2
%Invalid interface type and number
Switch(config)#int fa1/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security maximum 1
Switch(config-if)#switchport port-security violation shutdown
Switch(config-if)#exit
Switch(config)#
Switch(config)#
Switch(config)#
Switch(config)#
Switch(config)#
```

Now, for the next interface connection access the shutdown mode.

```
#int fa2/1
```

```
#switchport mode access
```

```
#switchport port-security
```

```
#switchport port-security mac-address sticky
```

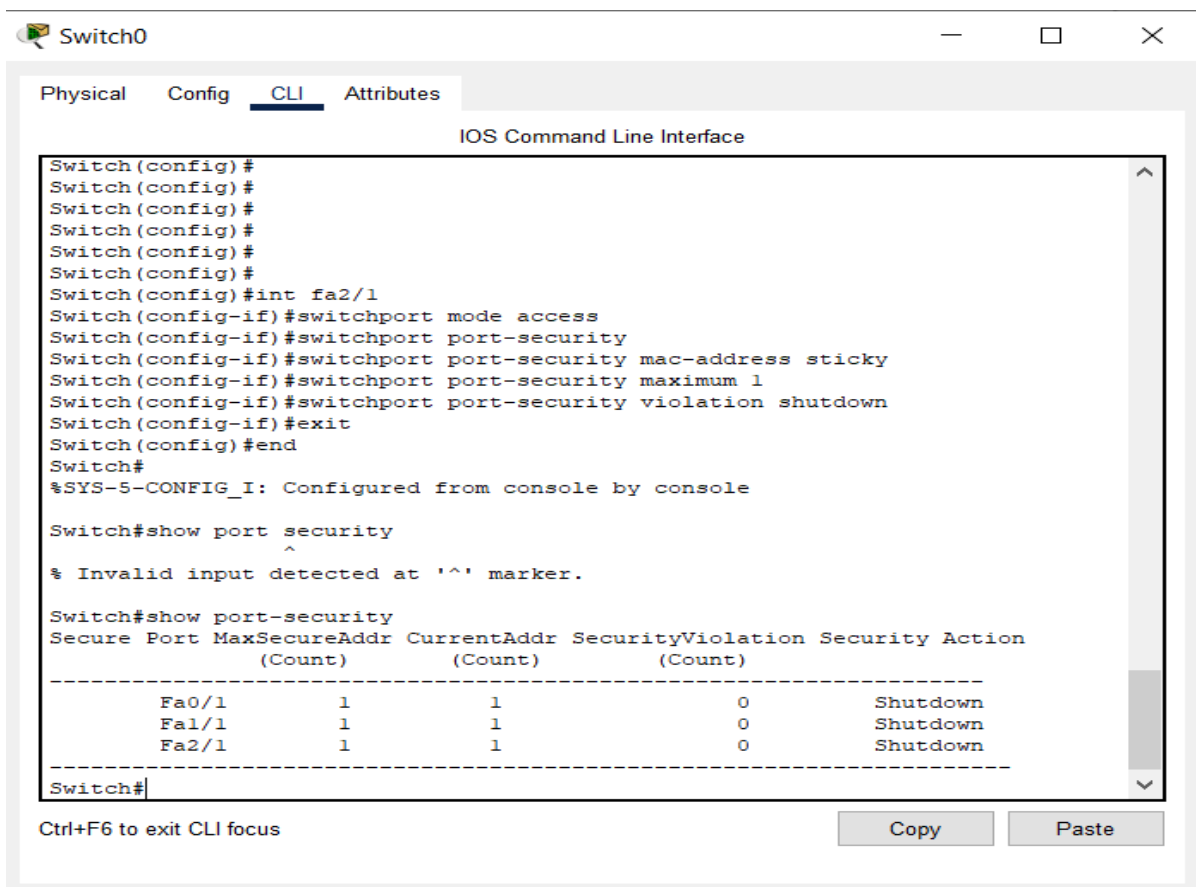
```
#switchport port-security maximum 1
```

```
#switchport port-security violation shutdown
```

```
#exit
```

```
#end
```

```
#show port-security
```



The screenshot shows a network switch CLI window titled "Switch0". The window has tabs for "Physical", "Config", "CLI", and "Attributes", with "CLI" selected. The main area displays the "IOS Command Line Interface" with the following commands and output:

```
Switch(config)#
Switch(config)#
Switch(config)#
Switch(config)#
Switch(config)#
Switch(config)#
Switch(config)#int fa2/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security maximum 1
Switch(config-if)#switchport port-security violation shutdown
Switch(config-if)#exit
Switch(config)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show port security
^
% Invalid input detected at '^' marker.

Switch#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
      (Count)      (Count)      (Count)
-----
Fa0/1          1          1          0      Shutdown
Fa1/1          1          1          0      Shutdown
Fa2/1          1          1          0      Shutdown
-----
```

At the bottom of the window, there is a status bar with the text "Ctrl+F6 to exit CLI focus" and two buttons: "Copy" and "Paste".

```
#show port-security int fa0/1
```

Switch0

Physical Config CLI Attributes

IOS Command Line Interface

Secure	Port	MaxSecureAddr (Count)	CurrentAddr (Count)	SecurityViolation (Count)	Security Action
	Fa0/1	1	1	0	Shutdown
	Fa1/1	1	1	0	Shutdown
	Fa2/1	1	1	0	Shutdown

```

Switch#
Switch#
Switch#
Switch#
Switch#
Switch#
Switch#show port-security int fa0/1
Port Security           : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses : 0
Sticky MAC Addresses    : 1
Last Source Address:Vlan : 0005.5E67.8469:1
Security Violation Count : 0

Switch#
Switch#
Switch#

```

Ctrl+F6 to exit CLI focus

Copy Paste

#show port-security int fa1/1

Switch0

Physical Config CLI Attributes

IOS Command Line Interface

```

Port Security           : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses : 0
Sticky MAC Addresses    : 1
Last Source Address:Vlan : 0005.5E67.8469:1
Security Violation Count : 0

Switch#
Switch#
Switch#show port-security int fa1/1
Port Security           : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses : 0
Sticky MAC Addresses    : 1
Last Source Address:Vlan : 000A.F3D8.3E76:1
Security Violation Count : 0

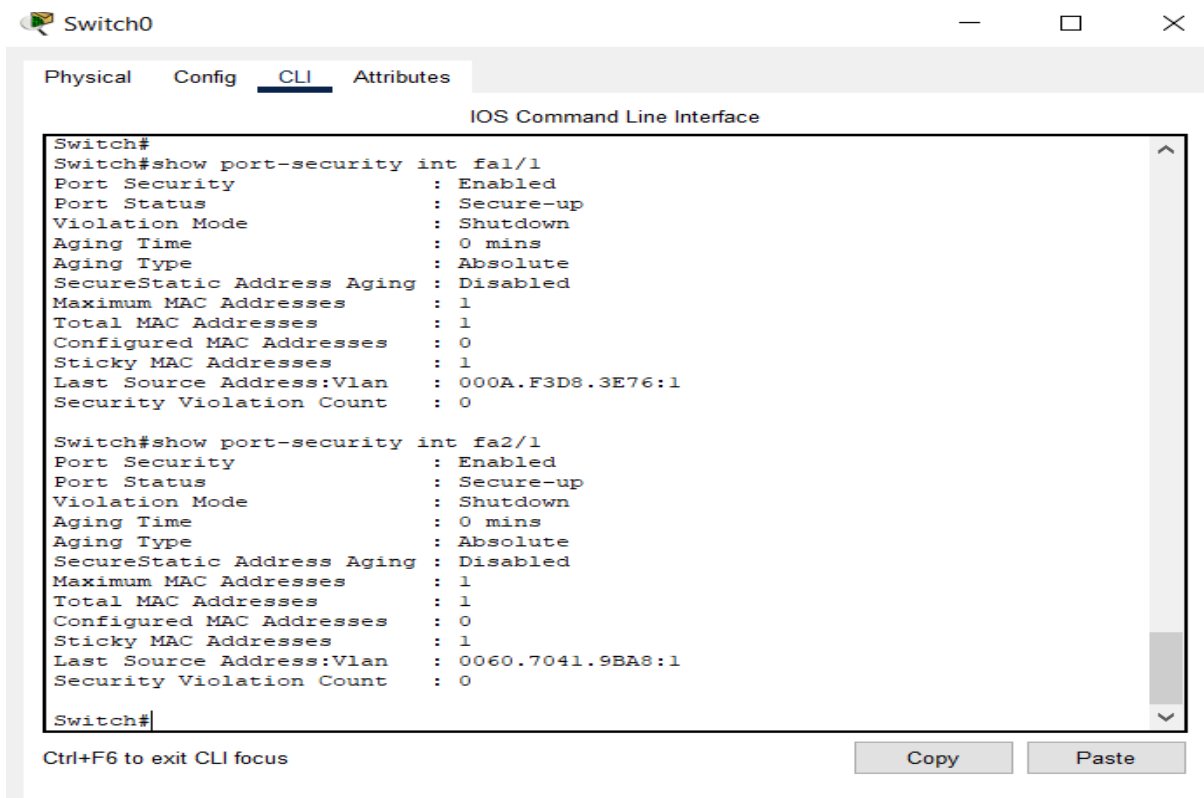
Switch#

```

Ctrl+F6 to exit CLI focus

Copy Paste

#show port-security int fa2/1



Switch0

Physical Config CLI Attributes

IOS Command Line Interface

```
Switch#
Switch#show port-security int fa1/1
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode         : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 1
Total MAC Addresses    : 1
Configured MAC Addresses : 0
Sticky MAC Addresses   : 1
Last Source Address:Vlan : 000A.F3D8.3E76:1
Security Violation Count : 0

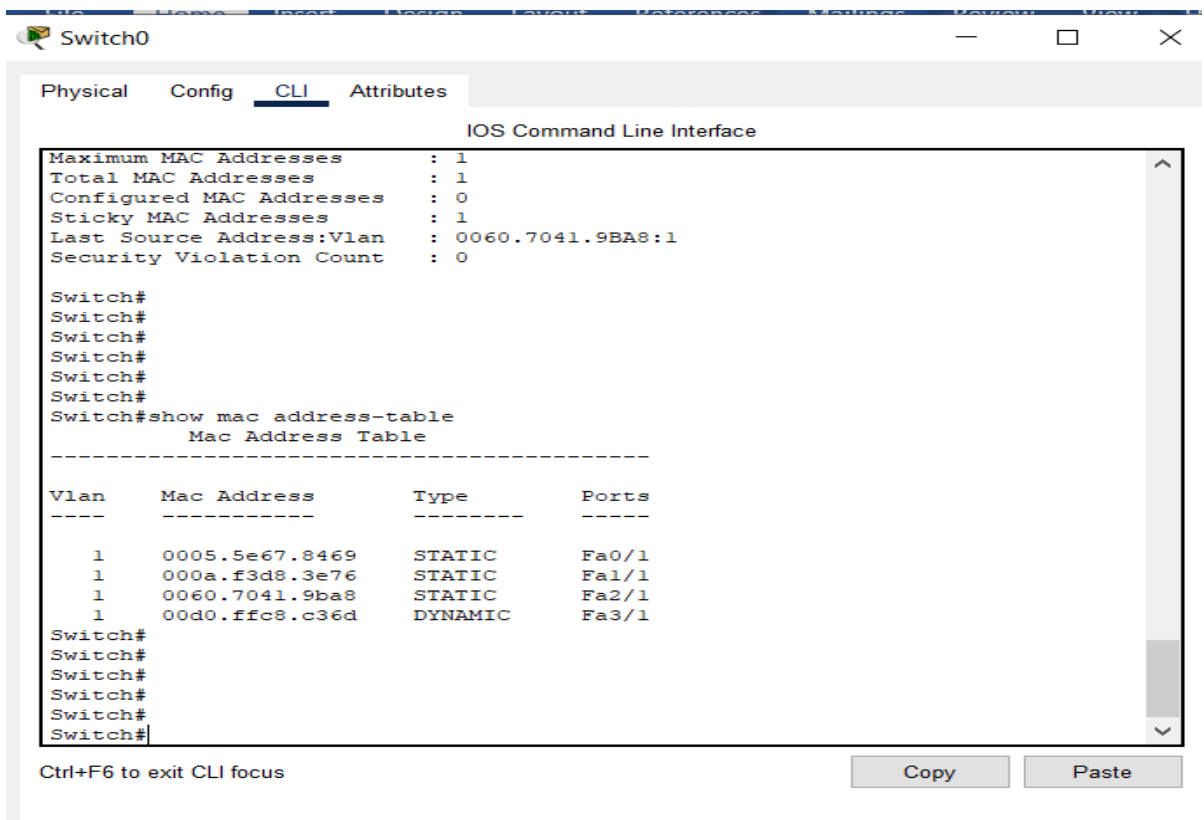
Switch#show port-security int fa2/1
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode         : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 1
Total MAC Addresses    : 1
Configured MAC Addresses : 0
Sticky MAC Addresses   : 1
Last Source Address:Vlan : 0060.7041.9BA8:1
Security Violation Count : 0

Switch#
```

Ctrl+F6 to exit CLI focus

Copy Paste

#show mac address-table



Switch0

Physical Config CLI Attributes

IOS Command Line Interface

```
Maximum MAC Addresses  : 1
Total MAC Addresses    : 1
Configured MAC Addresses : 0
Sticky MAC Addresses   : 1
Last Source Address:Vlan : 0060.7041.9BA8:1
Security Violation Count : 0

Switch#
Switch#
Switch#
Switch#
Switch#
Switch#
Switch#show mac address-table
      Mac Address Table
-----
Vlan    Mac Address      Type      Ports
----    -
1       0005.5e67.8469    STATIC    Fa0/1
1       000a.f3d8.3e76    STATIC    Fa1/1
1       0060.7041.9ba8    STATIC    Fa2/1
1       00d0.ffc8.c36d    DYNAMIC   Fa3/1

Switch#
Switch#
Switch#
Switch#
Switch#
Switch#
```

Ctrl+F6 to exit CLI focus

Copy Paste

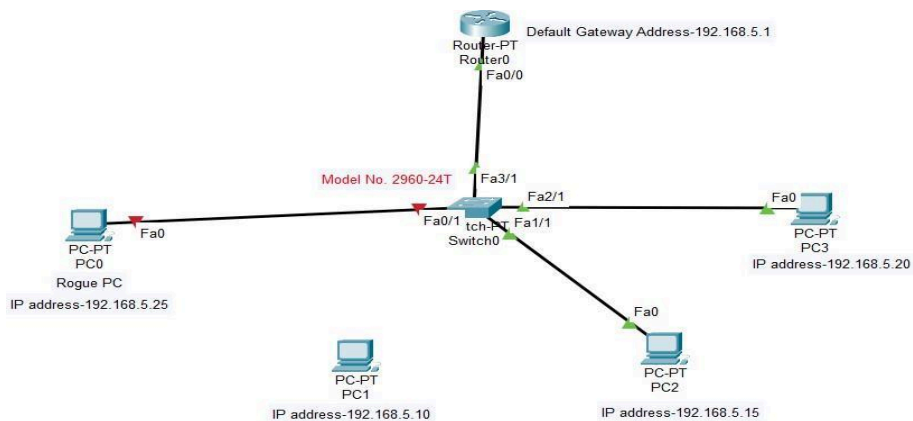
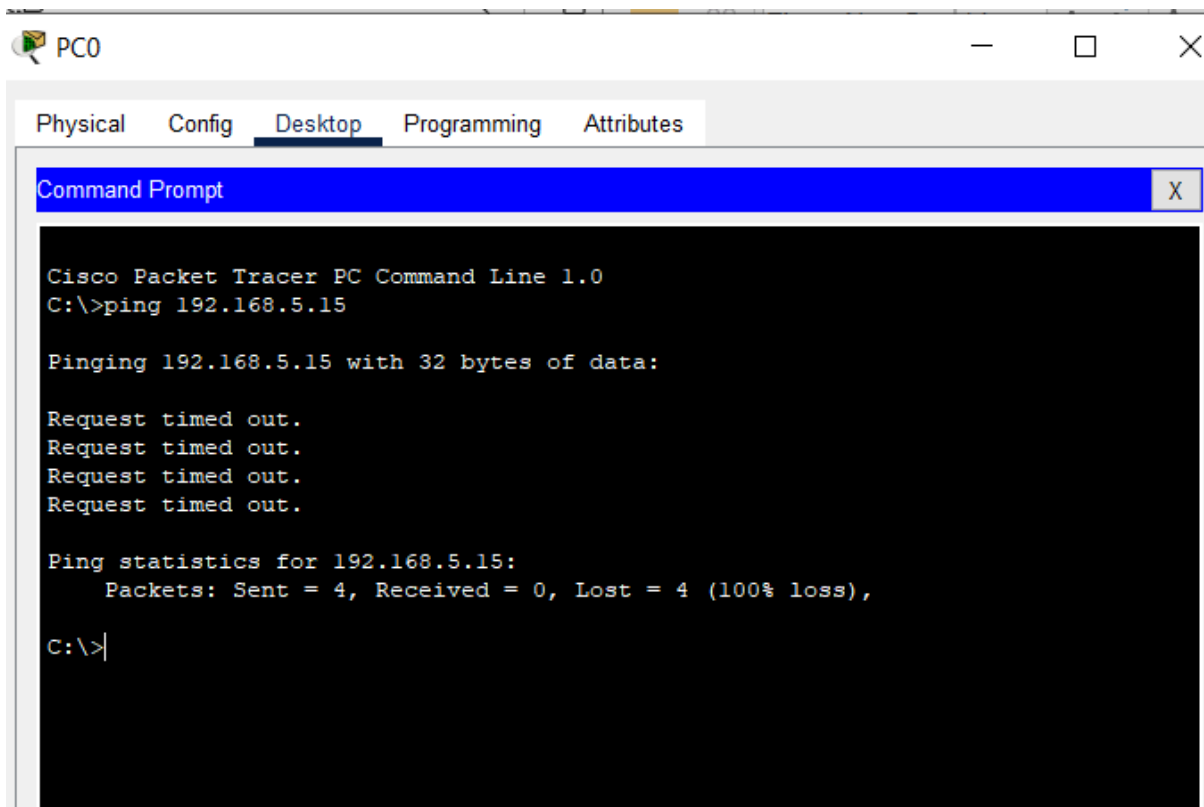
Step-7 Now, Ping all PCs.

Step-8 Now, Give IP address for Rogue PC0.

Step-9 Now, again Ping PC1, PC2 and PC3.

Step-9 Now remove the wired connection from PC1 and give connection to Rogue PC0.

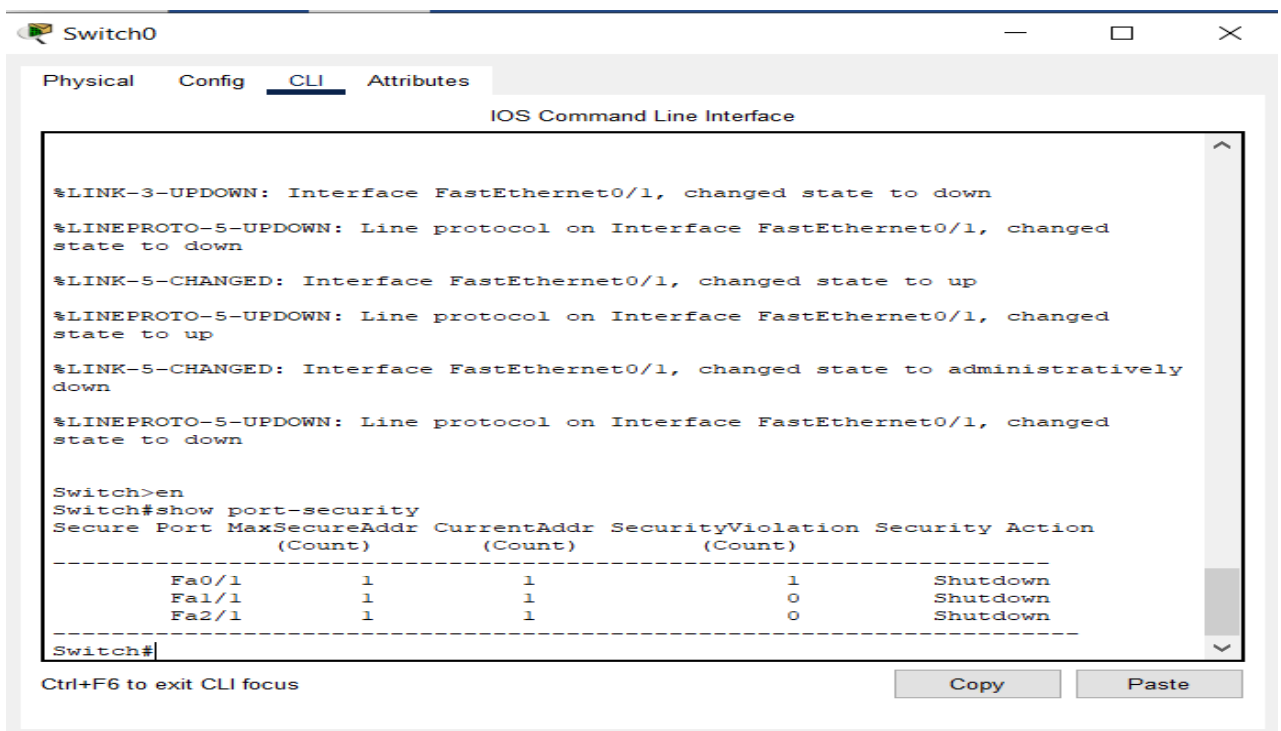
Step-10 Then, Ping the IP address of PC2 192.168.5.15 in the command prompt.



Step-11 Click on Switch

#enable

#show port-security



```
Switch0
Physical Config CLI Attributes
IOS Command Line Interface

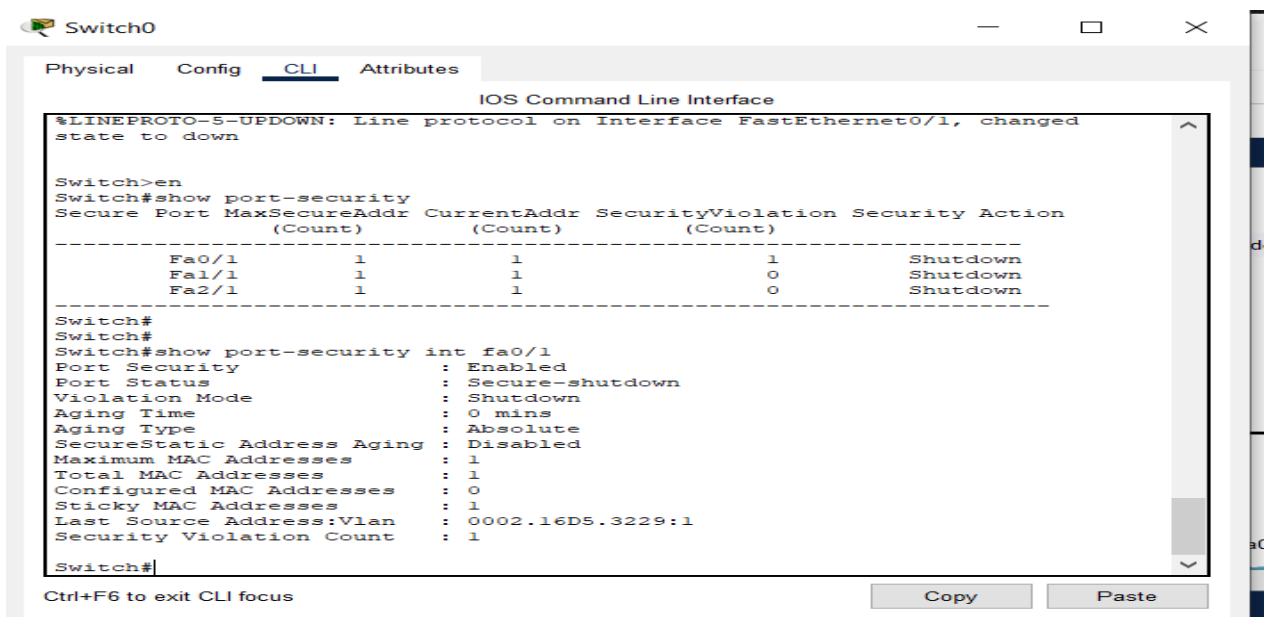
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

Switch>en
Switch#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
      (Count)      (Count)      (Count)
-----
Fa0/1          1          1          1      Shutdown
Fa1/1          1          1          0      Shutdown
Fa2/1          1          1          0      Shutdown
-----
Switch#
```

Ctrl+F6 to exit CLI focus

Copy Paste

#show port-security int fa0/1



```
Switch0
Physical Config CLI Attributes
IOS Command Line Interface

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

Switch>en
Switch#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
      (Count)      (Count)      (Count)
-----
Fa0/1          1          1          1      Shutdown
Fa1/1          1          1          0      Shutdown
Fa2/1          1          1          0      Shutdown
-----
Switch#
Switch#show port-security int fa0/1
Port Security      : Enabled
Port Status        : Secure-shutdown
Violation Mode      : Shutdown
Aging Time          : 0 mins
Aging Type          : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 1
Last Source Address:Vlan : 0002.16D5.3229:1
Security Violation Count : 1

Switch#
```

Ctrl+F6 to exit CLI focus

Copy Paste

#show mac address-table

Switch0

Physical Config CLI Attributes

IOS Command Line Interface

```

Port Security      : Enabled
Port Status       : Secure-shutdown
Violation Mode     : Shutdown
Aging Time        : 0 mins
Aging Type        : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 1
Last Source Address:Vlan : 0002.16D5.3229:1
Security Violation Count : 1

Switch#
Switch#
Switch#
Switch#show mac address-table
      Mac Address Table
-----
Vlan    Mac Address      Type      Ports
----    -
1       000a.f3d8.3e76   STATIC    Fa1/1
1       0060.7041.9ba8   STATIC    Fa2/1
1       00d0.ffc8.c36d   DYNAMIC    Fa3/1
Switch#
Switch#
Switch#
Switch#

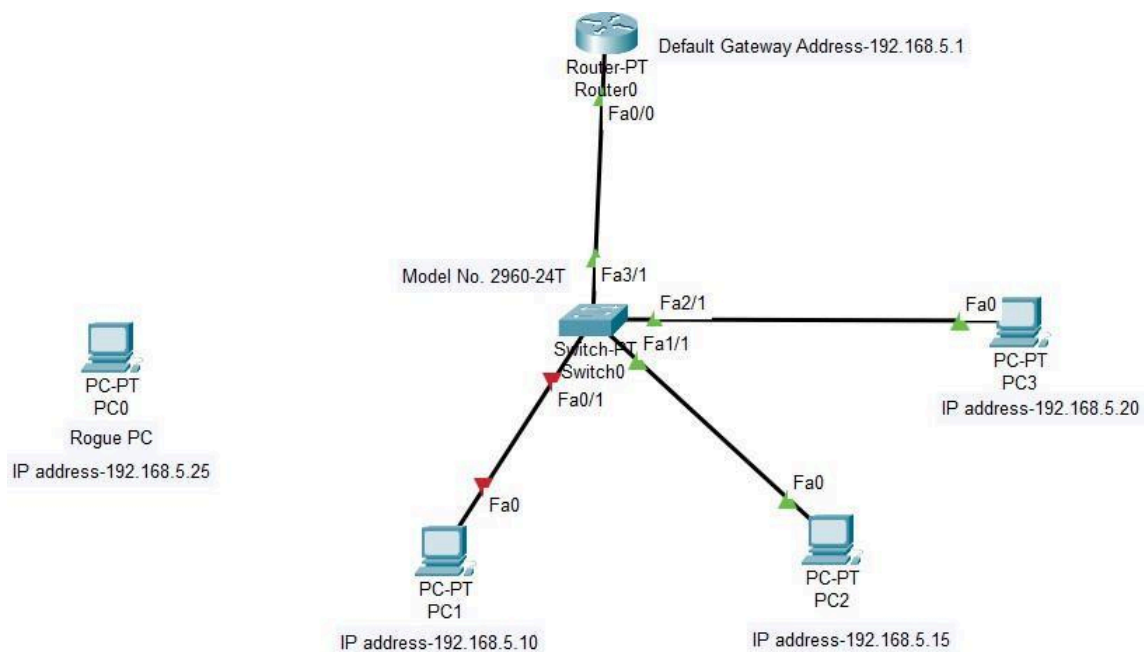
```

Ctrl+F6 to exit CLI focus

Copy Paste

Close it.

Step-12 Now, remove the wired connection from Rogue PC0 and connect the wire to PC1. But still it is in red color mode.



Step-13 Click on switch

#enable

#configure terminal

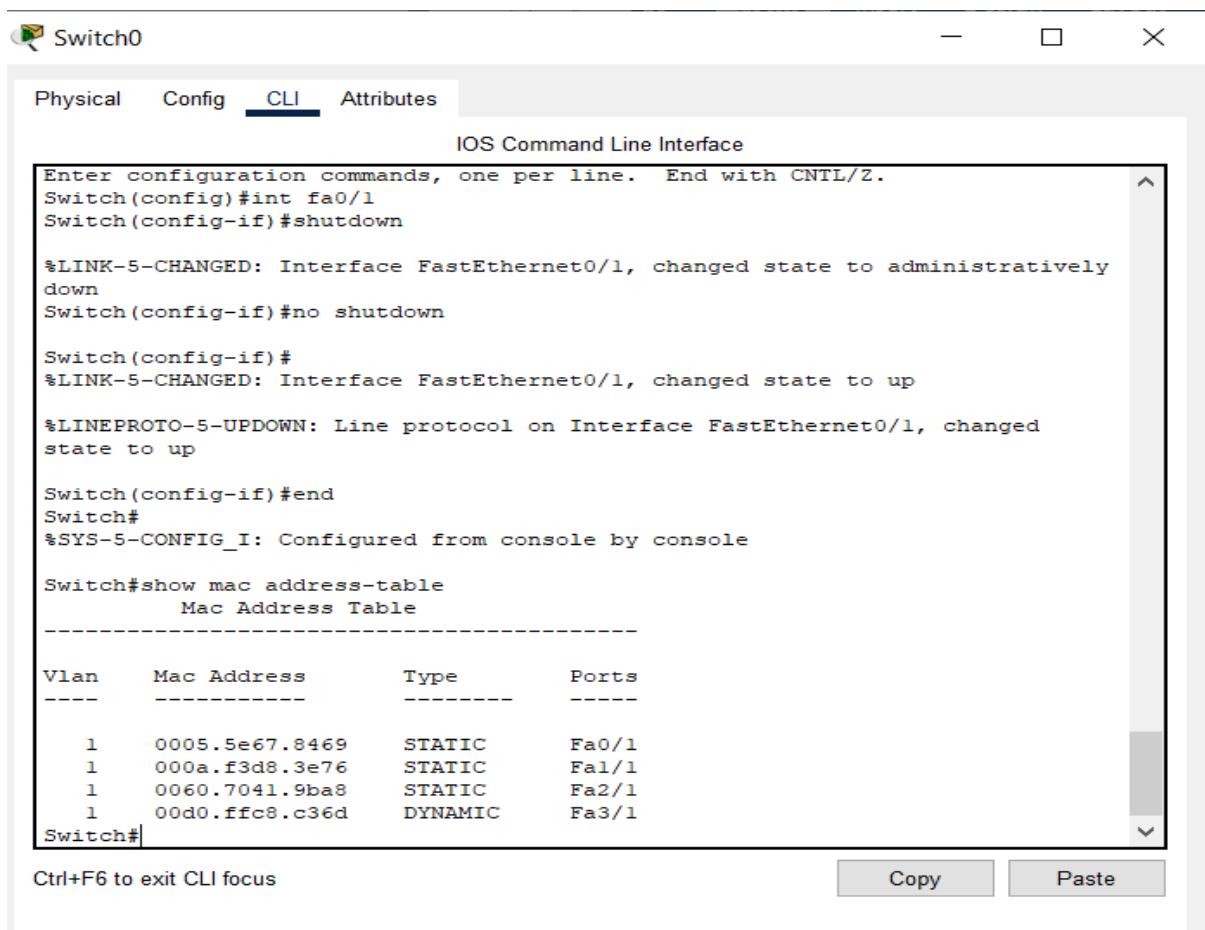
#int fa0/1

#shutdown

#no shutdown

#end

#show mac address-table



The screenshot shows the CLI interface of a switch named Switch0. The interface has tabs for Physical, Config, CLI (selected), and Attributes. The CLI tab displays the following commands and output:

```
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fa0/1
Switch(config-if)#shutdown

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively
down
Switch(config-if)#no shutdown

Switch(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up

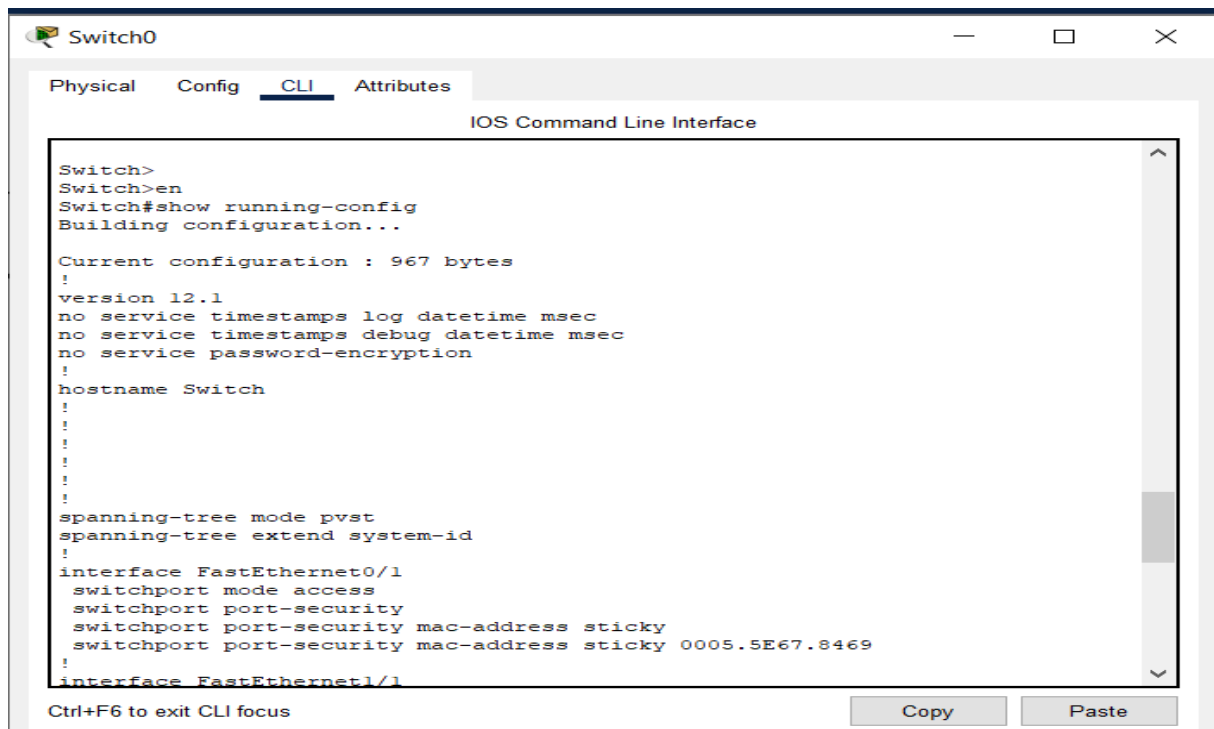
Switch(config-if)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show mac address-table
      Mac Address Table
-----
Vlan    Mac Address      Type      Ports
----    -
1       0005.5e67.8469    STATIC    Fa0/1
1       000a.f3d8.3e76    STATIC    Fa1/1
1       0060.7041.9ba8    STATIC    Fa2/1
1       00d0.ffc8.c36d    DYNAMIC   Fa3/1
Switch#
```

At the bottom of the CLI window, there is a prompt "Ctrl+F6 to exit CLI focus" and two buttons: "Copy" and "Paste".

Step-14 Click on PC1 and ping the command 192.168.5.15 and 192.168.5.20

#show running-config



Restrict Mode:

Step-1 Click on switch 0

#enable

#configure terminal

#int fa1/1

#switchport mode access

#switchport port-security

#switch port-security violation restrict

#exit

#end

#show port-security

#show port-security int fa1/1

Step-2 Now, cut the wired connection from PC2 and give connection to PC0

(Rogue).

Step-3 Give the Ping command 192.168.5.10 and 192.168.5.20

(mode of switch not changed, we will get request timeout)

Close it

Step-4 Click switch 0

#show port-security

#show port-security int fa1/1

Step-5 Now cut the connection wire connection from PC0 (Rogue) and connect to PC2.

Step-6 Check ping command from PC2 to PC1 and PC3, 192.168.5.10 and 192.168.5.20.

#show running-config

Protect Mode:

Step-1 Click on switch 0

#enable

#configure terminal

#int fa2/1

#switchport mode access

#switchport port-security

#switch port-security violation protect

#exit

#end

#show port-security

```
#show port-security int fa2/1
```

Step-2 Check ping command PC1 to PC2 and PC3,

Check ping command PC3 to PC2 and PC1.

Step-3 Now, cut the wired connection from PC3 and give connection to PC0 (Rogue).

Step-4 Give the Ping command 192.168.5.10 and 192.168.5.20

(mode of switch not changed, we will get request timeout)

Close it

Step-5 Click switch 0

```
#show port-security
```

```
#show port-security int fa1/1
```

```
#show mac address-table
```

```
#show ip int br
```

```
Ping -t 192.168.5.10
```

```
Cntrl c
```

OUTPUT:

RESULT:

Experiment -3

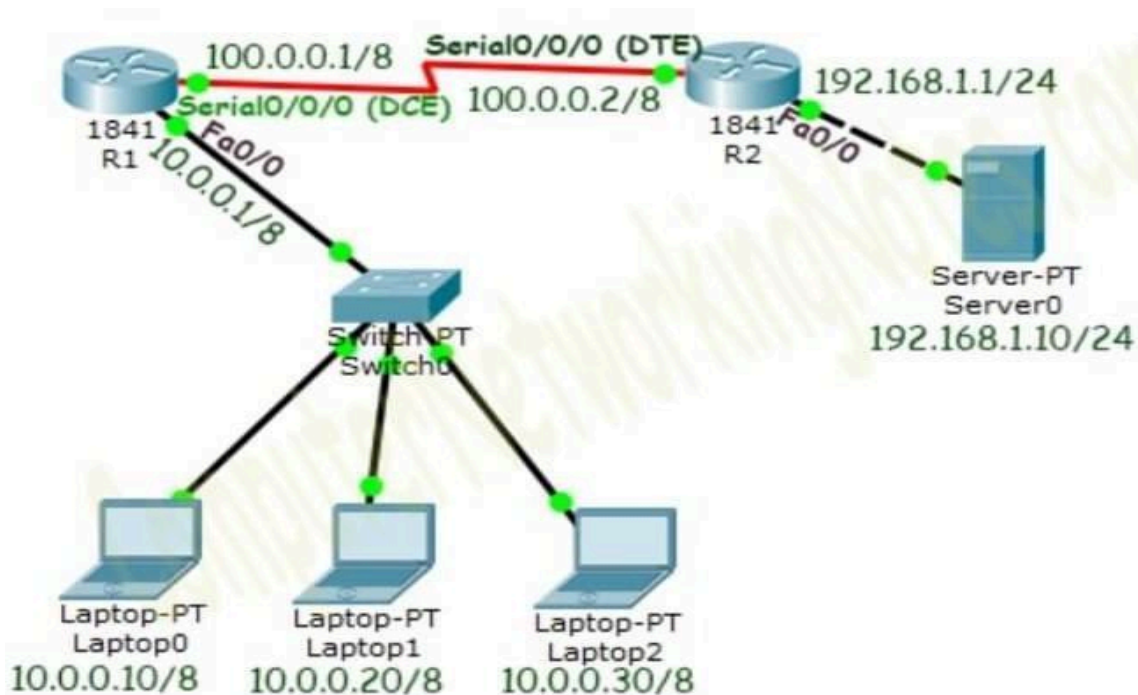
Date:

Configure Static NAT in Cisco Packet Tracer

In order to configure NAT we have to understand four basic terms; inside local, inside global, outside local and outside global. These terms define which address will be mapped with which address.

Term	Description
Inside Local IP Address	Before translation source IP address located inside the local network.
Inside Global IP Address	After translation source IP address located outside the local network.
Outside Global IP Address	Before translation destination IP address located outside the remote network.
Outside Local IP Address	After translation destination IP address located inside the remote network.

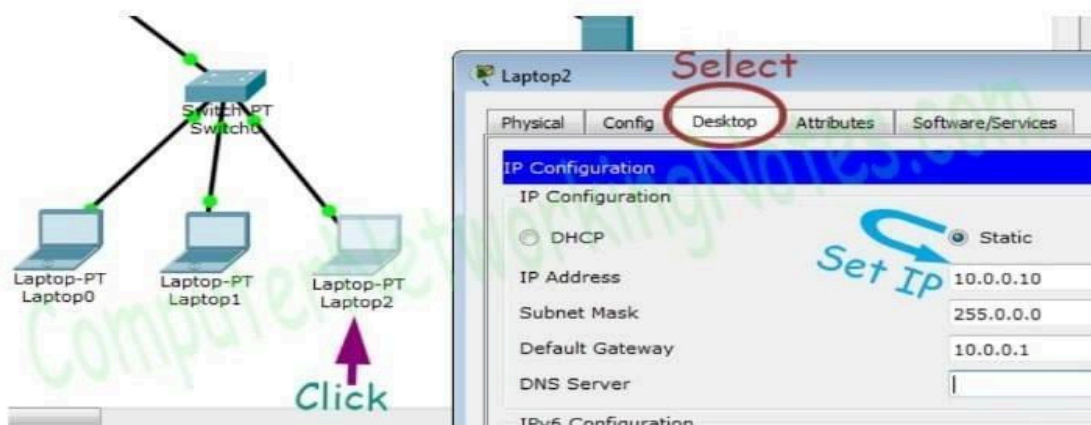
Create a practice Lab :



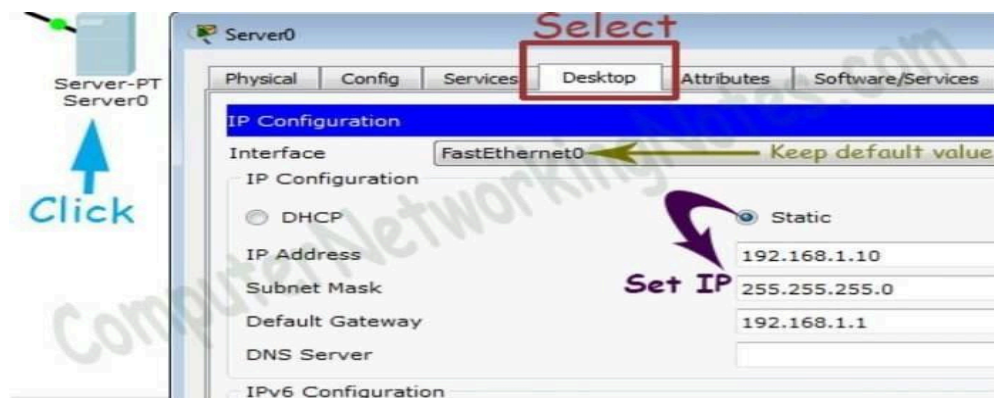
Initial IP Configuration

Device / Interface	IP Address	Connected with
Laptop0	10.0.0.10/8	Fa0/0 of R0
Laptop1	10.0.0.20/8	Fa0/0 of R0
Laptop2	10.0.0.30/8	Fa0/0 of R0
Server0	192.168.1.10/24	Fa0/0 of R1
Serial 0/0/0 of R1	100.0.0.1/8	Serial 0/0/0 of R2
Serial 0/0/0 of R2	100.0.0.2/8	Serial 0/0/0 of R2

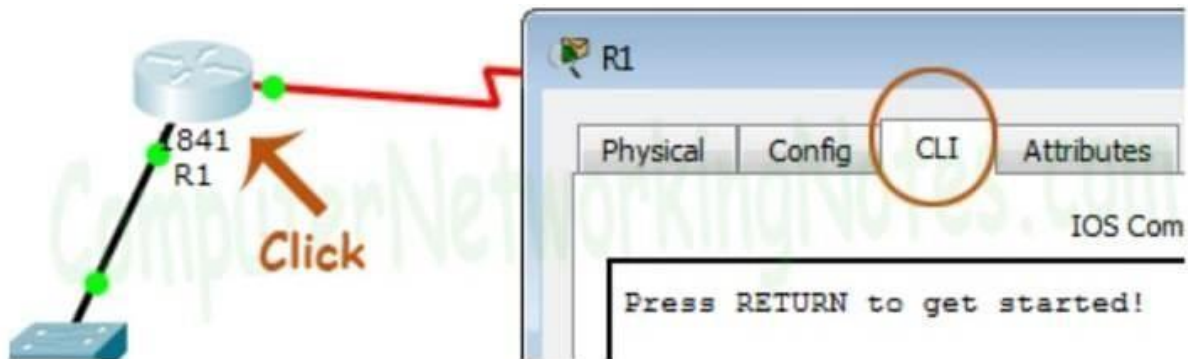
To assign IP address in Laptop click Laptop and click Desktop and IP configuration and Select Static and set IP address as given in above table.



Following same way configure IP address in Server.



To configure IP address in Router1 click **Router1** and select **CLI** and press **Enter key**.



Two interfaces of Router1 are used in topology; FastEthernet0/0 and Serial 0/0/0.

By default, interfaces on router are remain administratively down during the start up. We need to configure IP address and other parameters on interfaces before we could actually use them for routing. Interface mode is used to assign the IP address and other parameters. Interface mode can be accessed from global configuration mode. Following commands are used to access the global configuration mode.

```
Router>enable
```

```
Router# configure terminal
```

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

```
Router(config)#
```

Before we configure IP address in interfaces let's assign a unique descriptive name to router.

```
Router(config)#hostname R1
```

```
R1#
```

Now execute the following commands to set IP address in FastEthernet 0/0 interface.

```
R1(config)#interface FastEthernet0/0
```

```
R1(config-if)#ip address 10.0.0.1 255.0.0.0
```

R1(config-if)#no shutdown

R1(config-if)#exit

- interface FastEthernet 0/0 command is used to enter in interface mode.
- ip address 10.0.0.1 255.0.0.0 command assigns IP address to interface.
- no shutdown command is used to bring the interface up.
- exit command is used to return in global configuration mode.
- Serial interface needs two additional parameters clock rate and bandwidth. Every serial cable has two ends DTE and DCE. These parameters are always configured at DCE end.
- We can use show controllers interface command from privilege mode to check the cable's end.

R1(config)#exit

R1#show controllers serial 0/0/0

Interface Serial0/0/0

Hardware is PowerQUICC MPC860

DCE V.35, clock rate 2000000

[Output omitted]

- Fourth line of output confirms that DCE end of serial cable is attached.

If you see DTE here instead of DCE skip these parameters.

- Now we have necessary information let's assign IP address to serial interface.

R1#configure terminal

R1(config)#interface Serial0/0/0

R1(config-if)#ip address 100.0.0.1 255.0.0.0

R1(config-if)#clock rate 64000

R1(config-if)#bandwidth 64

R1(config-if)#no shutdown

R1(config-if)#exit

R1(config)#

Router#configure terminal Command is used to enter in global configuration mode.

Router(config)#interface serial 0/0/0 Command is used to enter in interface mode.

Router(config-if)#ip address 100.0.0.1 255.0.0.0 Command assigns IP address to interface.

Router(config-if)#clock rate 64000

In real life environment this parameter controls the data flow between serial links and need to be set at service provider's end. In lab environment we need not to worry about this value. We can use any valid rate here.

Router(config-if)#bandwidth 64

Bandwidth works as an influencer. It is used to influence the metric calculation of EIGRP or any other routing protocol which uses bandwidth parameter in route selection process.

Router(config-if)#no shutdown Command brings interface up.

Router(config-if)#exit Command is used to return in global configuration mode.

We will use same commands to assign IP addresses on interfaces of Router2. We need to provided clock rate and bandwidth only on DCE side of serial interface. Following command will assign IP addresses on interface of Router2.

Initial IP configuration in R2

Router>enable

Router#configure terminal

Router(config)#hostname R2

R2(config)#interface FastEthernet0/0

R2(config-if)#ip address 192.168.1.1 255.255.255.0

R2(config-if)#no shutdown

R2(config-if)#exit

R2(config)#interface Serial0/0/0

R2(config-if)#ip address 100.0.0.2 255.0.0.0

R2(config-if)#no shutdown

R2(config-if)#exit

R2(config)#

Configure Static NAT

Static NAT configuration requires three steps: -

1. Define IP address mapping
2. Define inside local interface
3. Define inside global interface

Since static NAT use manual translation, we have to map each inside local IP address (which needs a translation) with inside global IP address. Following command is used to map the inside local IP address with inside global IP address.

```
Router(config)#ip nat inside source static [inside local ip address] [inside global IP address]
```

For example in our lab Laptop1 is configured with IP address 10.0.0.10. To map it with 50.0.0.10 IP address we will use following command

```
Router(config)#ip nat inside source static 10.0.0.10 50.0.0.10
```

In second step we have to define which interface is connected with local the network. On both routers interface Fa0/0 is connected with the local network which need IP translation.

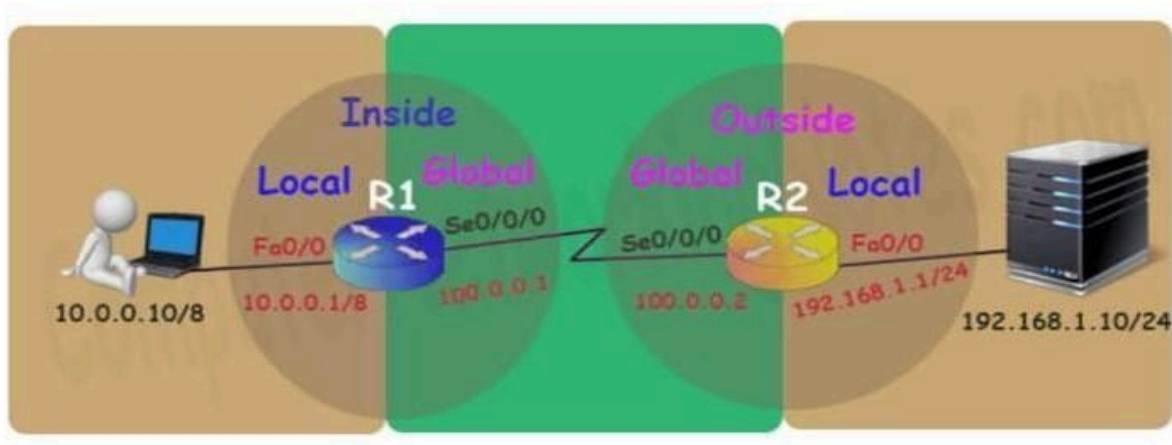
Following command will define interface Fa0/0 as inside local.

```
Router(config-if)#ip nat inside
```

In third step we have to define which interface is connected with the global network. On both routers serial 0/0/0 interface is connected with the global network. Following command will define interface Serial0/0/0 as inside global.

```
Router(config-if)#ip nat outside
```

Following figure illustrates these terms.



Let's implement all these commands together and configure the static NAT.

R1 Static NAT Configuration

```
R1(config)#ip nat inside source static 10.0.0.10 50.0.0.10
```

```
R1(config)#interface FastEthernet 0/0
```

```
R1(config-if)#ip nat inside
```

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

```
R1(config)#
```

```
R1(config)#interface Serial 0/0/0
```

```
R1(config-if)#ip nat outside
```

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

For testing purpose I configured only one static translation. You may use following commands to configure the translation for remaining address.

```
R1(config)#ip nat inside source static 10.0.0.20 50.0.0.20
```

```
R1(config)#ip nat inside source static 10.0.0.30 50.0.0.30
```

R2 Static NAT Configuration

```
R2(config)#ip nat inside source static 192.168.1.10 200.0.0.10
```

```
R2(config)#interface FastEthernet 0/0
```

```
R2(config-if)#ip nat inside
```

```
R2(config-if)#exit
```

```
R2(config)#
```


R2(config)#interface Serial 0/0/0

R2(config-if)#ip nat outside

R2(config-if)#exit

Before we test this lab we need to configure the IP routing. IP routing is the process which allows router to route the packet between different networks. Following tutorial explain routing in detail with examples

Configure static routing in R1

R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2

Configure static routing in R2

R2(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1

Testing Static NAT Configuration

We configured static NAT on R1 and R2. On R1 we mapped inside local IP address 10.0.0.10 with inside global address 50.0.0.10 while on R2 we mapped inside local IP address 192.168.1.10 with inside global IP address 200.0.0.10.

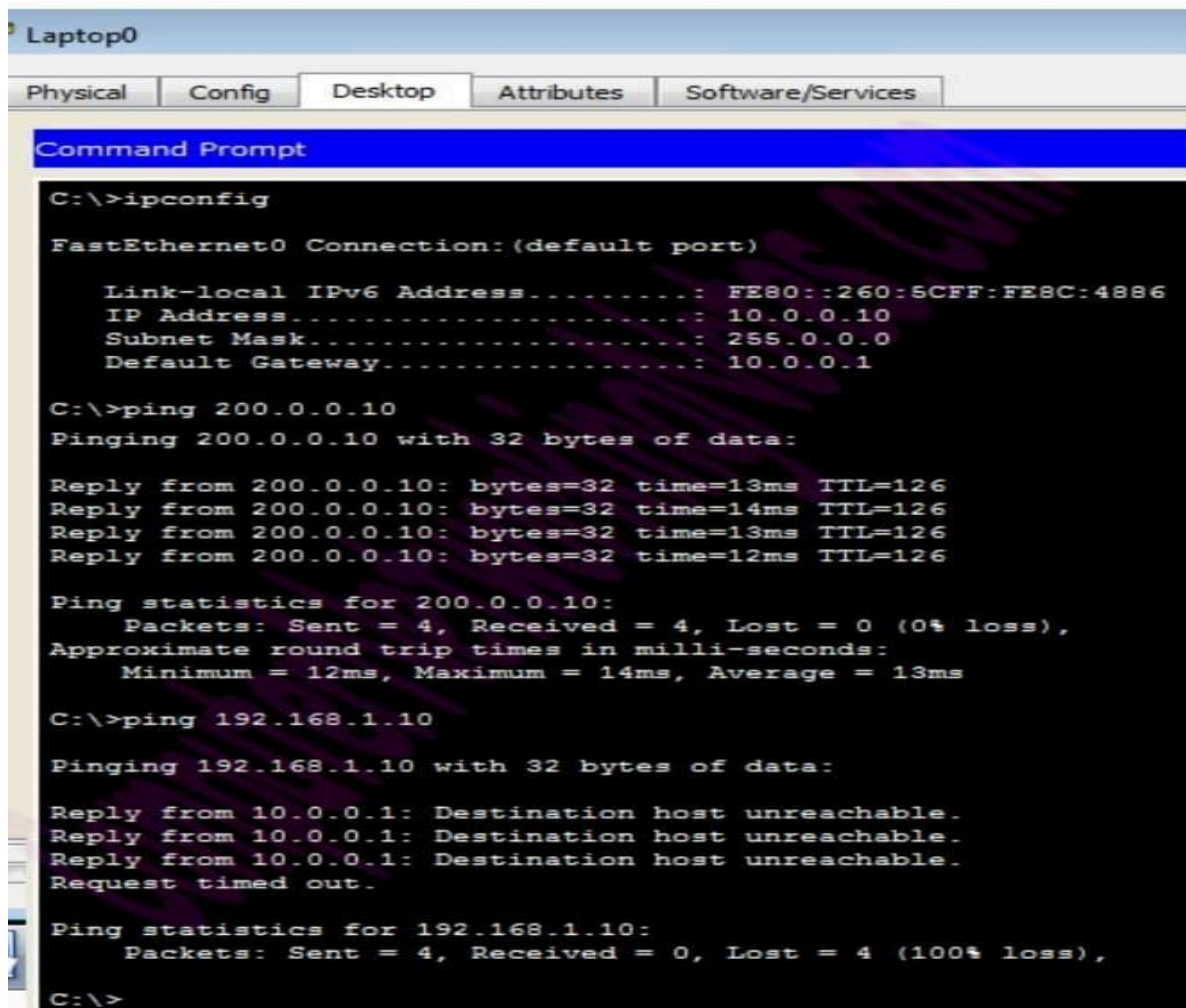
Device	Inside Local IP Address	Inside Global IP Address
Laptop0	10.0.0.10	50.0.0.10
Server	192.168.1.10	200.0.0.10

To test this setup click Laptop0 and Desktop and click Command Prompt.

Run ipconfig command.

Run ping 200.0.0.10 command.

Run ping 192.168.1.10 command.



```
C:\>ipconfig

FastEthernet0 Connection: (default port)

    Link-local IPv6 Address . . . . . : FE80::260:5CFF:FE8C:4886
    IP Address. . . . . : 10.0.0.10
    Subnet Mask . . . . . : 255.0.0.0
    Default Gateway . . . . . : 10.0.0.1

C:\>ping 200.0.0.10
Pinging 200.0.0.10 with 32 bytes of data:

Reply from 200.0.0.10: bytes=32 time=13ms TTL=126
Reply from 200.0.0.10: bytes=32 time=14ms TTL=126
Reply from 200.0.0.10: bytes=32 time=13ms TTL=126
Reply from 200.0.0.10: bytes=32 time=12ms TTL=126

Ping statistics for 200.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 14ms, Average = 13ms

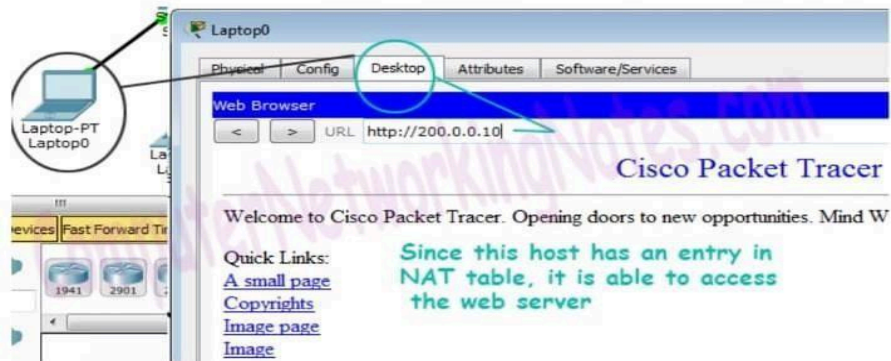
C:\>ping 192.168.1.10
Pinging 192.168.1.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

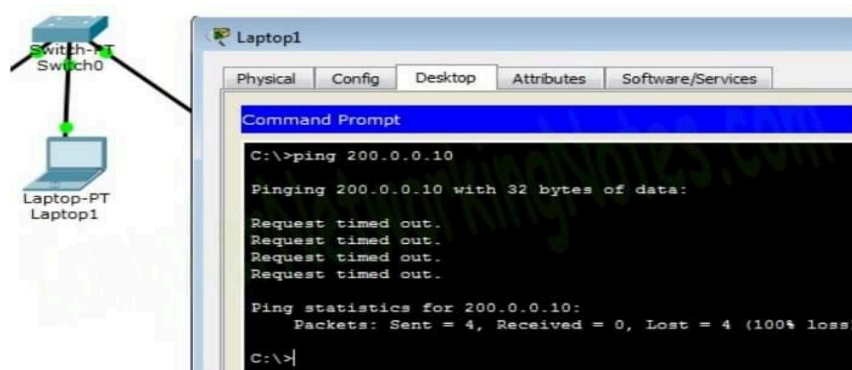
C:\>
```

- First command verifies that we are testing from correct NAT device.
- Second command checks whether we are able to access the remote device or not. A ping reply confirms that we are able to connect with remote device on this IP address.
- Third command checks whether we are able to access the remote device on its actual IP address or not. A ping error confirms that we are not able to connect with remote device on this IP address.
- Let's do one more testing. Click **Laptop0** and click **Desktop** and click **Web Browser** and access 200.0.0.10.



Above figure confirms that host 10.0.0.10 is able to access the 200.0.0.10.

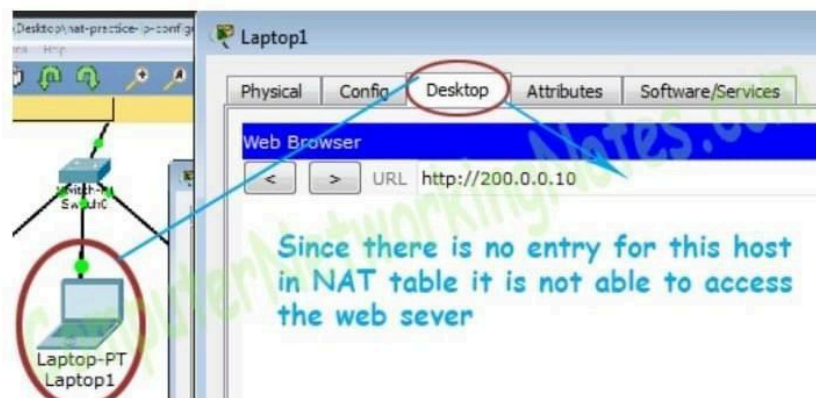
Now run ping 200.0.0.10 command from Laptop1.



Why we are not able to connect with the remote device from this host?

Because we configured NAT only for one host (Laptop0) which IP address is 10.0.0.10. So only the host 10.0.0.10 will be able to access the remote device.

To confirm it again, let's try to access web service from this host.



We can also verify this translation on router with show ip nat translation command.

Have you noticed one interesting feature of NAT in above output? Why actual outside local IP address is not listed in this field?

The actual IP address is not listed here because router is receiving packets after the translation. From R1's point of view remote device's IP address is 200.0.0.10 while from R2's point of view end device's IP address is 50.0.0.10. This way if NAT is enabled we would not be able to trace the actual end device.

Following figure illustrate this translation on router R1.

```
R1#show ip nat translations
Pro Inside global      Inside local      Outside local     Outside global
icmp 50.0.0.10:13      10.0.0.10:13      200.0.0.10:13     200.0.0.10:13
icmp 50.0.0.10:14      10.0.0.10:14      200.0.0.10:14     200.0.0.10:14
icmp 50.0.0.10:15      10.0.0.10:15      200.0.0.10:15     200.0.0.10:15
icmp 50.0.0.10:16      10.0.0.10:16      200.0.0.10:16     200.0.0.10:16
tcp 50.0.0.10:1030      10.0.0.10:1030     200.0.0.10:80      200.0.0.10:80
tcp 50.0.0.10:1031      10.0.0.10:1031     200.0.0.10:80      200.0.0.10:80
R1#
```

Following figure illustrate this translation on router R2

```
R2#show ip nat translations
Pro Inside global      Inside local      Outside local     Outside global
icmp 200.0.0.10:13      192.168.1.10:13   50.0.0.10:13      50.0.0.10:13
icmp 200.0.0.10:14      192.168.1.10:14   50.0.0.10:14      50.0.0.10:14
icmp 200.0.0.10:15      192.168.1.10:15   50.0.0.10:15      50.0.0.10:15
icmp 200.0.0.10:16      192.168.1.10:16   50.0.0.10:16      50.0.0.10:16
tcp 200.0.0.10:80        192.168.1.10:80    50.0.0.10:1030     50.0.0.10:1030
tcp 200.0.0.10:80        192.168.1.10:80    50.0.0.10:1031     50.0.0.10:1031
R2#
```

The actual IP address is not listed here because router is receiving packets after the translation. From R1's point of view remote device's IP address is 200.0.0.10 while from R2's point of view end device's IP address is 50.0.0.10. This way if NAT is enabled we would not be able to trace the actual end device.

OUTPUT:

RESULT:

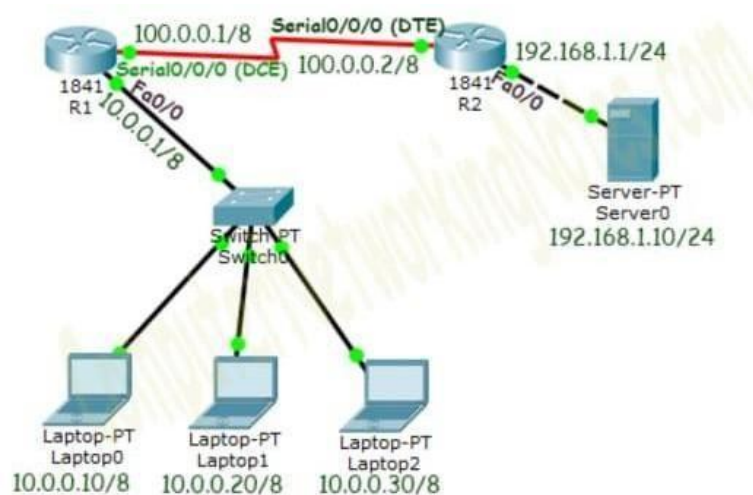
Experiment -4

Date:

Configure Dynamic NAT in Cisco Packet Tracer

This tutorial explains Dynamic NAT configuration (creating an access list of IP addresses which need translation, creating a pool of available IP address, mapping access list with pool and defining inside and outside interfaces) in detail. Learn how to configure, manage, verify and debug dynamic NAT step by step with packet tracer examples.

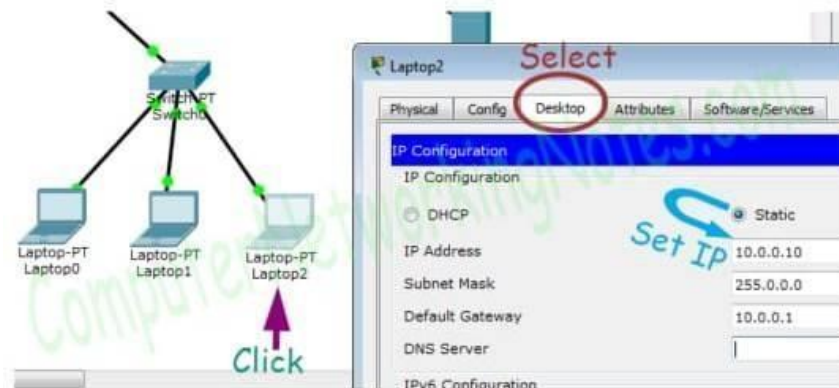
Create a practice lab as shown in following figure or download this pre-created practice lab and load in packet tracer



Initial IP Configuration

Device / Interface	IP Address	Connected with
Laptop0	10.0.0.10/8	Fa0/0 of R0
Laptop1	10.0.0.20/8	Fa0/0 of R0
Laptop2	10.0.0.30/8	Fa0/0 of R0
Server0	192.168.1.10/24	Fa0/0 of R1
Serial 0/0/0 of R1	100.0.0.1/8	Serial 0/0/0 of R2
Serial 0/0/0 of R2	100.0.0.2/8	Serial 0/0/0 of R2

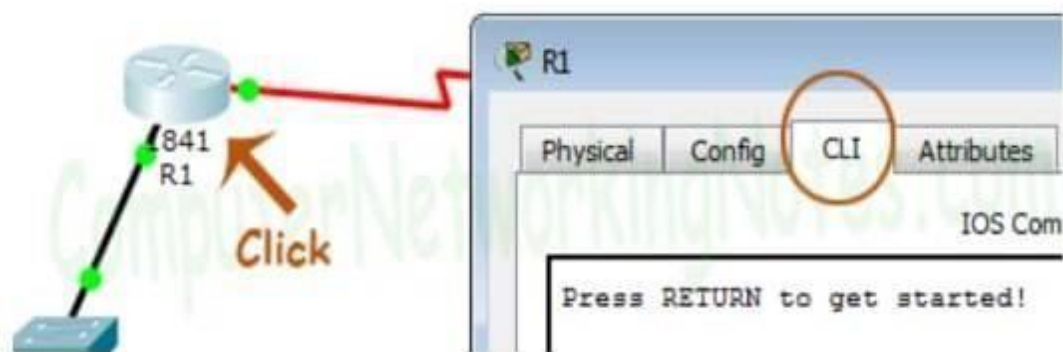
To assign IP address in Laptop click Laptop and click Desktop and click IP configuration and Select Static and set IP address as given in above table.



Following same way configure IP address in Server.



To configure IP address in Router1 click Router1 and select CLI and press Enter key.



Run following commands to set IP address and hostname.

```
Router>enable
```

```
Router#configure terminal
```

```
Router(config)#
```

```
Router(config)#hostname R1
```

```
R1(config)#interface FastEthernet0/0
```

```
R1(config-if)#ip address 10.0.0.1 255.0.0.0
```

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

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

```
R1(config)#interface Serial0/0/0
```

```
R1(config-if)#ip address 100.0.0.1 255.0.0.0
```

```
R1(config-if)#clock rate 64000
```

```
R1(config-if)#bandwidth 64
```

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

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

```
R1(config)#
```

Same way accesses the command prompt of R2 and run following commands to set IP address and hostname.

```
Router>enable
```

```
Router#configure terminal
```

```
Router(config)#hostname R2
```

```
R2(config)#interface FastEthernet0/0
```

```
R2(config-if)#ip address 192.168.1.1 255.255.255.0
```

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

```
R2(config-if)#exit
```

```
R2(config)#interface Serial0/0/0
```



```
R2(config-if)#ip address 100.0.0.2 255.0.0.0
```

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

```
R2(config-if)#exit
```

```
R2(config)#
```

That's all initial IP configuration we need. Now this topology is ready for the practice of dynamic nat.

Configure Dynamic NAT

Dynamic NAT configuration requires four steps: -

1. Create an access list of IP addresses which need translation
 2. Create a pool of all IP address which are available for translation
- Map access list with pool
3. Define inside and outside interfaces
 4. In first step we will create a standard access list which defines which inside local addresses are permitted to map with inside global address.

□ To create a standard numbered ACL following global configuration

mode command is used:-

```
Router(config)# access-list ACL_Identifier_number permit/deny matching-parameters
```

Let's understand this command and its options in detail.

Router(config)#

This command prompt indicates that we are in global configuration mode.

access-list

Through this parameter we tell router that we are creating or accessing an access list.

ACL_Identifier_number

With this parameter we specify the type of access list. We have two types of access list; standard and extended. Both lists have their own unique identifier numbers. Standard ACL uses numbers range 1 to 99 and 1300 to 1999. We can pick any number from this range to tell the router that we are working with standard ACL. This number is used in grouping the conditions under a single ACL. This number is also a unique identifier for this ACL in router.

permit/deny

An ACL condition has two actions; permit and deny. If we use permit keyword, ACL will allow all packets from the source address specified in next parameter. If we use deny keyword, ACL will drop all packets from the source address specified in next parameter.

matching-parameters

This parameter allows us to specify the contents of packet that we want to match. In a standard ACL condition it could be a single source address or a range of addresses. We have three options to specify the source address.

Any

host

A.B.C.D

Any

Any keyword is used to match all sources. Every packet compared against this condition would be matched.

Host

Host keyword is used to match a specific host. To match a particular host, type the keyword host and then the IP address of host.

A.B.C.D

Through this option we can match a single address or a range of addresses. To match a single address, simply type its address. To match a range of addresses, we need to use wildcard mask.

Wildcard mask

Just like subnet mask, wildcard mask is also used to draw a boundary in IP address. Where subnet mask is used to separate network address from host address, wildcard mask is used to distinguish the matching portion from the rest. Wildcard mask is the invert of Subnet mask. Wildcard can be calculated in decimal or in binary from subnet mask.

We have three hosts in lab. Let's create a standard access list which allows two hosts and denies one host.

```
R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
```

```
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
```

```
R1(config)#access-list 1 deny any
```

In second step we define a pool of inside global addresses which are available for translation.

Following command is used to define the NAT pool.

```
Router(config)#ip nat pool [Pool Name] [Start IP address] [End IP address]  
netmask [Subnet mask]
```

This command accepts four options pool names, start IP address, end IP address and Subnet mask.

Pool Name: - This is the name of pool. We can choose any descriptive name here.

Start IP Address: - First IP address from the IP range which is available for translation.

End IP Address: - Last IP address from the IP range which is available for translation. There are no minimum or maximum criteria for IP range for example we can have a range of single IP address or we can have a range of all IP address from a subnet.

Subnet Mask: - Subnet mask of IP range.

Let's create a pool named ccna with an IP range of two addresses.

```
R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.2 netmask 255.0.0.0
```

This pool consists two class A IP address 50.0.0.1 and 50.0.0.2.

In third step we map access list with pool. Following command will map the access list with pool and configure the dynamic NAT.

```
Router(config)#ip nat inside source list [access list name or number] pool [pool name]
```

This command accepts two options.

Access list name or number: - Name or number the access list which we created in first step.

Pool Name: - Name of pool which we created in second step.

- In first step we created a standard access list with number 1 and in second step we created a pool named ccna. To configure a dynamic NAT with these options we will use following command.

```
R1(config)#ip nat inside source list 1 pool ccna
```

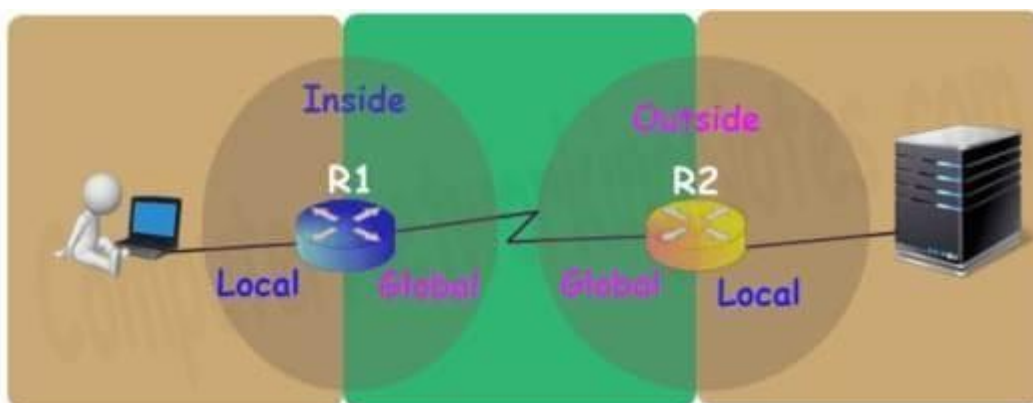
Finally we have to define which interface is connected with local network and which interface is connected with global network.

- To define an inside local we use following command

```
Router(config-if)#ip nat inside
```

- Following command defines inside global

```
Router(config-if)#ip nat outside
```



Let's implement all these commands together and configure the dynamic NAT.

R1 Dynamic NAT Configuration

```
R1#configure terminal
```

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

```
R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
```

```
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
```

```
R1(config)#access-list 1 deny any
```

```
R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.2 netmask 255.0.0.0
```

```
R1(config)#ip nat inside source list 1 pool ccna
```

```
R1(config)#interface FastEthernet 0/0
```

```
R1(config-if)#ip nat inside
```

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

```
R1(config)#interface Serial0/0/0
```

```
R1(config-if)#ip nat outside
```

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

```
R1(config)#
```

For testing purpose, here configured dynamic translations for two addresses only.

On R2 we can keep standard configuration or can configure dynamic NAT as we just did in R1 or can configure static NAT as we learnt in pervious part of this article.

Let's do a quick recap of what we learnt in previous part and configure static NAT on R2.

```
R2>enable
```

```
R2#configure terminal
```

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

```
R2(config)#ip nat inside source static 192.168.1.10 200.0.0.10
```

```
R2(config)#interface Serial 0/0/0
```

```
R2(config-if)#ip nat outside
```

```
R2(config-if)#exit
```

```
R2(config)#interface FastEthernet 0/0
```

```
R2(config-if)#ip nat inside
```

```
R2(config-if)#exit
```

```
R2(config)#
```

Before we test this lab we need to configure the IP routing. IP routing is the process which allows router to route the packet between different networks.

Configure static routing in R1

R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2

Configure static routing in R2

R2(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1

Testing Dynamic NAT Configuration

We configured dynamic NAT on R1 for 10.0.0.10 and 10.0.0.20 and static NAT on R2 for 192.168.1.10.

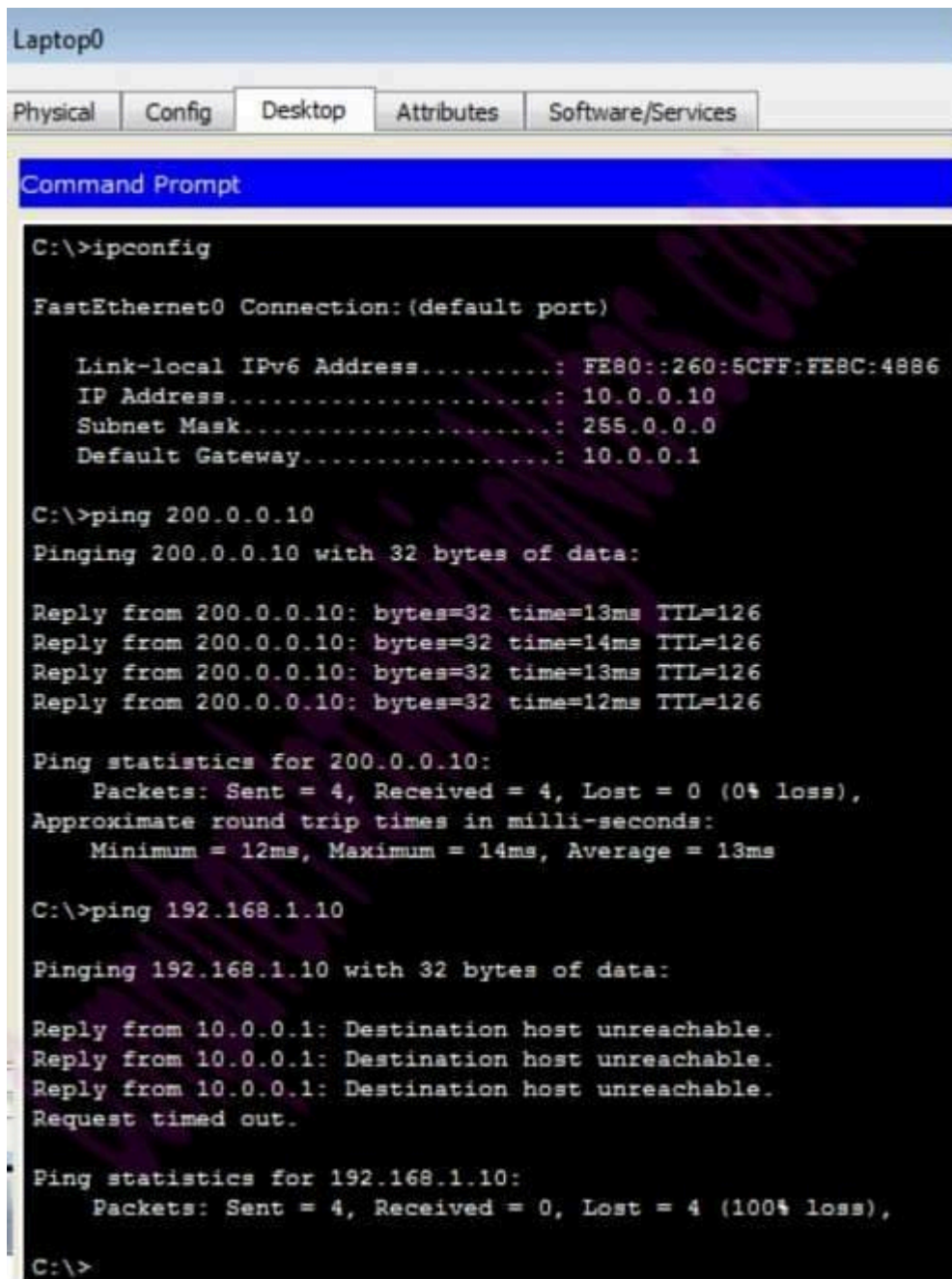
Device	Inside Local IP Address	Inside Global IP Address
Laptop0	10.0.0.10	50.0.0.1
Laptop1	10.0.0.20	50.0.0.2
Server	192.168.1.10	200.0.0.10

To test this setup click Laptop0 and Desktop and click Command Prompt.

Run ipconfig command.

Run ping 200.0.0.10 command.

Run ping 192.168.1.10 command.



The screenshot shows a Windows Command Prompt window titled "Laptop0". The window has tabs for "Physical", "Config", "Desktop", "Attributes", and "Software/Services". The "Config" tab is selected, and the "Command Prompt" window is open. The command prompt shows the following output:

```
C:\>ipconfig

FastEthernet0 Connection: (default port)

    Link-local IPv6 Address . . . . . : FE80::260:5CFF:FE8C:4886
    IP Address. . . . . : 10.0.0.10
    Subnet Mask . . . . . : 255.0.0.0
    Default Gateway . . . . . : 10.0.0.1

C:\>ping 200.0.0.10

Pinging 200.0.0.10 with 32 bytes of data:

Reply from 200.0.0.10: bytes=32 time=13ms TTL=126
Reply from 200.0.0.10: bytes=32 time=14ms TTL=126
Reply from 200.0.0.10: bytes=32 time=13ms TTL=126
Reply from 200.0.0.10: bytes=32 time=12ms TTL=126

Ping statistics for 200.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 14ms, Average = 13ms

C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

First command verifies that we are testing from correct NAT device.

Second command checks whether we are able to access the remote device or not. A ping reply confirms that we are able to connect with remote device on this IP address.

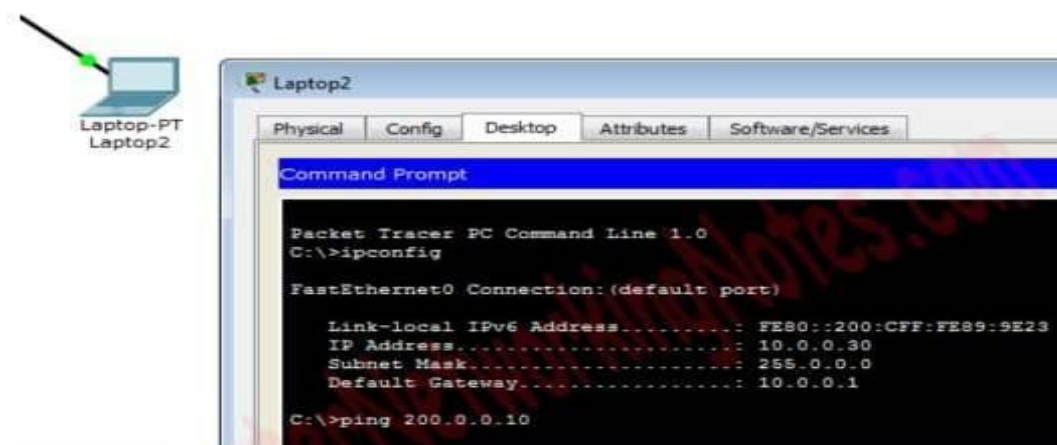
Third command checks whether we are able to access the remote device on its actual IP address or not. A ping error confirms that we are not able to connect with remote device on this IP address.

Let's do one more testing. Close the command prompt and click web server and access 200.0.0.10.

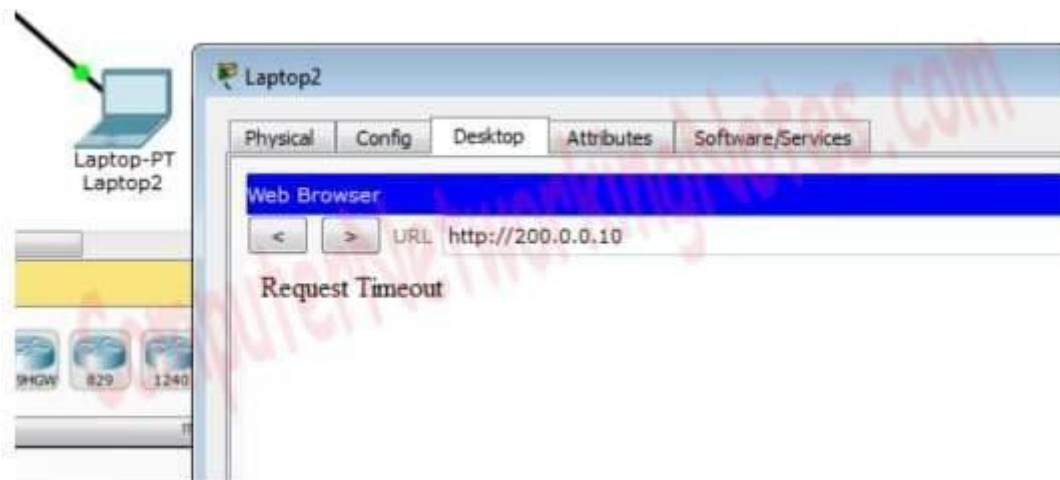


Above figure confirms that host 10.0.0.10 is able to access the 200.0.0.10. You can also do the same testing from Laptop1, result will be same.

Now run ping 200.0.0.10 command from Laptop2.



Close the command prompt and access web server from this host.



Why we are not able to connect with the remote device from this host?

Because we configured NAT only for two hosts (Laptop0 and Laptop1) which IP addresses are 10.0.0.10 and 10.0.0.20. So only the host 10.0.0.10 and 10.0.0.20 will be able to access the remote device.

We can also verify this translation on router with *show ip nat translation* command.

Following figure illustrates this translation on router R1.

```
R1>en
R1#show ip nat translations
Pro Inside global      Inside local      Outside local     Outside global
tcp 50.0.0.1:1025      10.0.0.10:1025    200.0.0.10:80     200.0.0.10:80
tcp 50.0.0.2:1025      10.0.0.20:1025    200.0.0.10:80     200.0.0.10:80
R1#
```

We did three tests one from each host, but why only two tests are listed here? Remember in first step we created an access list. Access list filters the unwanted traffic before it reaches to the NAT. We can see how many packets are blocked by ACL with following command

R1#show ip access-lists 1

```

R1#show ip access-lists 1
Standard IP access list 1
    permit host 10.0.0.10 (8 match(es))
    permit host 10.0.0.20 (2 match(es))
    deny any (3 match(es))
R1#

```

Basically it is access list which filters the traffic. NAT does not filter any traffic it only translate the address.

Following figure illustrate NAT translation on router R2

```

R2>enable
R2#show ip nat translations

```

Pro	Inside global	Inside local	Outside local	Outside global
---	200.0.0.10	192.168.1.10	---	---
tcp	200.0.0.10:80	192.168.1.10:80	50.0.0.1:1025	50.0.0.1:1025
tcp	200.0.0.10:80	192.168.1.10:80	50.0.0.2:1025	50.0.0.2:1025

```

R2#

```

OUTPUT:

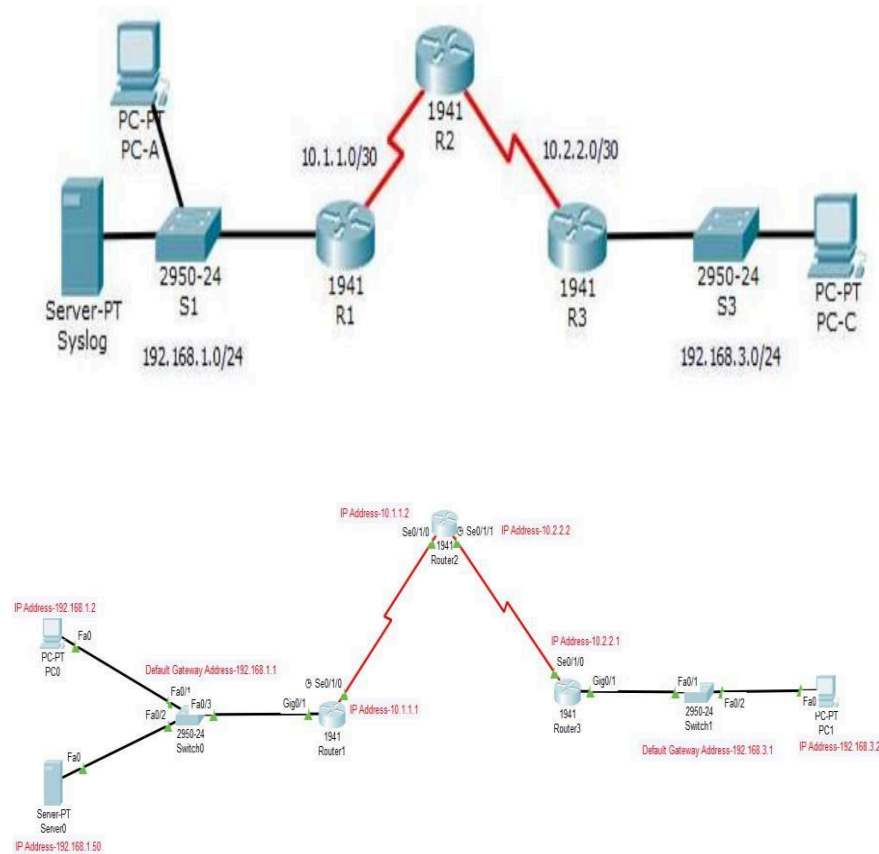
RESULT:

Experiment -5

Date:

Configure IDS/IPS in Cisco Packet Tracer

Network Topology



Addressing Table:

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

Objectives:

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

Background / Scenario

Your task is to enable IPS on R1 to scan traffic entering the 192.168.1.0 network.

The server labeled Syslog is used to log IPS messages. You must configure the router to identify the syslog

server to receive logging messages. Displaying the correct time and date in syslog messages is vital when

using syslog to monitor the network. Set the clock and configure the timestamp service for logging on the

routers. Finally, enable IPS to produce an alert and drop ICMP echo reply packets inline.

The server and PCs have been preconfigured.

User Access Authentication

Step-1 Click on Router1

```
#enable
```

```
#conf t
```

```
#username xxxx secret yyyy
```

```
#aaa new
```

```
#aaa new-model
```

```
#aaa authentication ?
```

#aaa authentication login ?

#aaa authentication login default ?

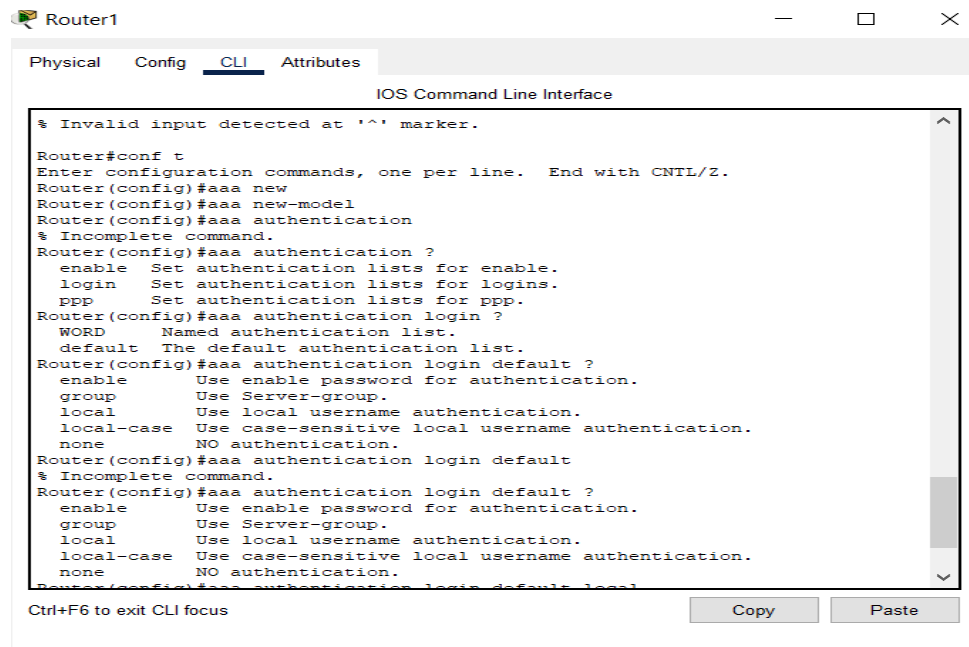
#aaa authentication login default local

#line console 0

#login authentication ?

#login authentication default

#exit



Router1

Physical Config **CLI** Attributes

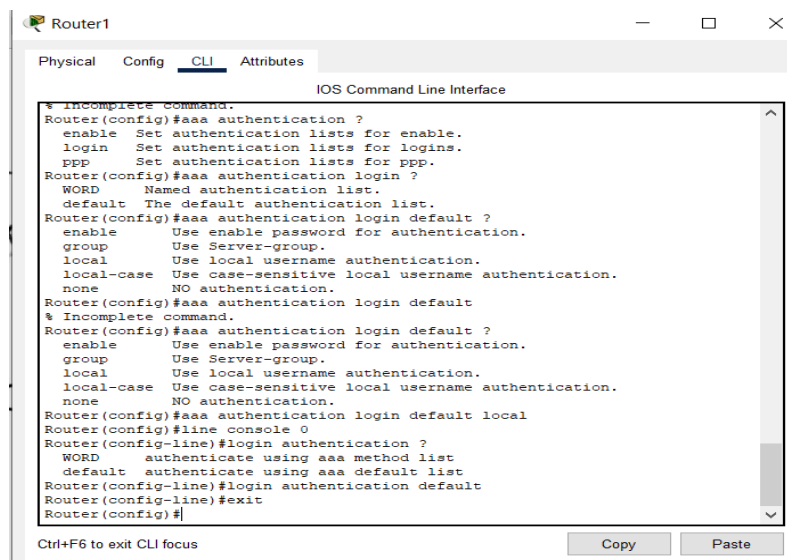
IOS Command Line Interface

```
% Invalid input detected at '^' marker.

Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#aaa new
Router(config)#aaa new-model
Router(config)#aaa authentication
% Incomplete command.
Router(config)#aaa authentication ?
    enable    Set authentication lists for enable.
    login     Set authentication lists for logins.
    ppp       Set authentication lists for ppp.
Router(config)#aaa authentication login ?
    WORD      Named authentication list.
    default   The default authentication list.
Router(config)#aaa authentication login default ?
    enable    Use enable password for authentication.
    group     Use Server-group.
    local     Use local username authentication.
    local-case Use case-sensitive local username authentication.
    none      NO authentication.
Router(config)#aaa authentication login default
% Incomplete command.
Router(config)#aaa authentication login default ?
    enable    Use enable password for authentication.
    group     Use Server-group.
    local     Use local username authentication.
    local-case Use case-sensitive local username authentication.
    none      NO authentication.
Router(config)#aaa authentication login default local
```

Ctrl+F6 to exit CLI focus

Copy Paste



Router1

Physical Config **CLI** Attributes

IOS Command Line Interface

```
% Incomplete command.
Router(config)#aaa authentication ?
    enable    Set authentication lists for enable.
    login     Set authentication lists for logins.
    ppp       Set authentication lists for ppp.
Router(config)#aaa authentication login ?
    WORD      Named authentication list.
    default   The default authentication list.
Router(config)#aaa authentication login default ?
    enable    Use enable password for authentication.
    group     Use Server-group.
    local     Use local username authentication.
    local-case Use case-sensitive local username authentication.
    none      NO authentication.
Router(config)#aaa authentication login default
% Incomplete command.
Router(config)#aaa authentication login default ?
    enable    Use enable password for authentication.
    group     Use Server-group.
    local     Use local username authentication.
    local-case Use case-sensitive local username authentication.
    none      NO authentication.
Router(config)#aaa authentication login default local
Router(config)#line console 0
Router(config-line)#login authentication ?
    WORD      authenticate using aaa method list
    default   authenticate using aaa default list
Router(config-line)#login authentication default
Router(config-line)#exit
Router(config)#
```

Ctrl+F6 to exit CLI focus

Copy Paste

Step-2 Click on Router1

#enable

#show version

#conf t

#license boot module c1900 technology-package securityk9

#yes

#end

#copy running startup

#reload

#enable

#show version

Step-3 Click on PC0 Ping PC1 IP address

Step-4 Click on PC1 ping PC0 IP address

Step-5 Click on R1

#mkdir ipsdir

(create directory filename() ?)

(create directory flash:ipsdir)

#conf t

#ip ips config location flash:ipsdir

#ip ips name iosips

#ip ips notify log

#exit

#clock set 19:25:59 9 July 2023

#conf t

#service timestamps log datetime msec

#logging host 192.168.1.50

#ip ips signature-category

#category all

#retired true

#exit

#category ios_ips basic

#retired false

#exit

#exit

Do you want to accept these changes? [Confirm]

#int g0/1

#ip ips iosips out

#ip ips signature-definition

#signature 2004 0

#status

```
#retired false
#enabled true
#exit
#engine
#event-action produce-alert
#event-action deny-packet-inline
#exit
#exit
#exit
[Confirm]
#end
#show ip ips all
```

Step-6 Ping PC1

(Now, the request connection should be timeout the packets between the devices should deny the packets from the given IP address. This ping should fail. This PC2 the IPS rule for event-action of an echo request was set to deny-packet-inline.)

Step-7 Ping PC0

(Now, the request should be successful....)

Step-8 Check syslog (in server)

Physical Config **Services** Desktop Programming Attributes**SERVICES** ^

HTTP

DHCP

DHCPv6

TFTP

DNS

SYSLOG

AAA

NTP

EMAIL

FTP

IoT

VM Management

Radius EAP

Syslog

Syslog

Service

☒ On ☐ Off

	Time	HostName	Message
1	03.09.2022 06:09:37.905 PM	192.168.1.1	%IPS-4-SIGNATUR...

OUTPUT:

RESULT:

NETWORK SECURITY AUDIT CHECKLIST

Performing a network security audit is an effective way to monitor and evaluate the health of your network infrastructure. Consider any two IT company, **create suitable audit questionnaires and prepare a network security audit checklist for the given 10 steps.** So that you can take your network from uncomfortably vulnerable to confidently secure.

1. Define the scope of the audit

Decide which devices, operating systems, and access layers should be included in the audit.

2. Determine threats

Make a list of potential cybersecurity threats. These can include things such as malware, DDoS attacks, and risks from BYOD/at-home devices.

3. Review and edit internal policies

Understand which policies your company currently operates under, and which should be updated or added. Potential policies include a network security policy, privacy policy, remote access policy, data backup, and more. Also, review the procedure management system.

4. Ensure the safety of sensitive data

Limit who has access to sensitive data, and where that data is stored. Consider having separate storage for this important data, and ensure that it is not stored on a laptop.

5. Inspect the servers

Check that your server configurations are properly set up. Inspect the DNS and WINS servers, binding orders, static addr assignments, and backup network services. Additionally, ensure all network software is up to date.

6. Examine training logs and use log monitoring

Prevent human error by creating mandatory, comprehensive training processes so that employees and clients conduct operations safely. Also use software automation to continually and regularly check logs for new updates, patches, firewalls, and devices. A best practice is to remove inactive devices from the system.

7. Safe internet access

Data encryption, malware scanning, bandwidth restrictions, and port blocking are all potential measures to ensure that employees access and interact with wireless networks in a safe manner.

8. Penetration testing

Perform static testing for a high-level overview of vulnerabilities in your applications, and perform dynamic testing for more specific findings of your system. Locate all potential access points and remove any unauthorized points in your system.

9. Share the network security audit with the team

Work with the necessary people to share and implement what you have found. Create full transparency with employees.

10. Have regular network security audits

An audit should be performed one to two times per year to reduce the threat of cyber risks. Make it a normal part of your system maintenance routine.

Requires Check list format

Sr.No.	Audit Questionnaire	Company A (Yes/No)	Company B (Yes/No)	Remarks
Step 1	Define the scope of the audit			
1	Security camera has been installed to monitor the data center?	Yes	No	

	Check for, Security cameras Monitored by, Recording period Recording media Retention period Administered by			
--	---	--	--	--

Sample Questionnaire

Sr. No.	Audit Questionnaire	Document available Yes/No.	Comments
	Physical Security		
1	Do you have policy that addresses the physical security of the Data Center?		
2	Do you maintain register for entry/exit to data center? Is it records the purpose to visit the data center?		
3	Do you have electronic access control (Swipe Card) mechanism for entry/exit to data center?		
4	Do you take access control review, at what frequency?		
5	Do you allow the temporary access to data center? Is it recorded? Do you remove the temporary access as soon as work gets completed?		
6	What process is followed if any new person visit the data center? Do you escort the person visiting data center?		

7	Do you have control on door automatic lock? (Audio / visual alarm if door open for more than specified period or signs on both sides indicating it is to be closed and locked and contact to notify if it is found unsecured)		
---	--	--	--

8	Security camera has been installed to monitor the data center? Check for, Security cameras Monitored by, Recording period Recording media Retention period Administered by		
	Environmental and Electric control		
1	Does data center has an adequate and safe fire-suppression system with associated detectors (Heat, Smoke, and Temperature monitor)?		
2	When it was last tested? Smoke alarms - Test report Temperature monitoring system - Test report Fire extinguisher - Check expiry date. or Fire control system (Inert gas) - Test report		
3	Does the data center have redundant cooling system?		
4	Do you standard checklist for testing? Is the procedure documented?		
5	Do you have UPS system to backup		

	your data center electricity?		
6	Do you have updated details on Current electric load capacity of data center?		
7	What is the backup capacity of UPS System?		
8	When was the last UPS system tested? (Please check the test report)		

	Change management		
1	Do you have change management policy ? Is it addresses the process to be followed for changes in the data center?		
2	Do the changes in data center documented?		
3	What is the review mechanism for change management?		
	Inventory control		
1	Do you maintain the inventory of assets in data center?		
2	Is the inventory of assets in data center are up to date?		
3	Do you review asset inventory in data center? At what frequency?		
4	Are all the assets in data center are properly labeled?		

5	Do you have contact details of vendor for relevant systems in data center in case of an emergency?		
	Incident Management		
1	Does your company have an Incident Management policy?		
2	What methodologies are being adopted for Incident Management?		
3	Are management responsibilities and procedures established to ensure quick, effective and orderly response to information security incidents?		
4	Are the roles and responsibilities defined for incident management?		
5	Are the incidents documented and reported?		
6	Root cause analysis done to avoid the same?		

7	Do you have emergency procedures for a. Instructions for shutting off utilities. b. Instructions for powering down equipment. c. Instructions for activating/deactivating d. fire suppression equipment. e. Personnel evacuation. F. Security valuable assets.		
	Disaster recovery/Continuity management		
1	Do you have Disaster Recovery plan in place for Data center?		

2	Are all the processes documented in case of Disaster Recovery?		
3	Does the Disaster Recovery Policy addresses the following: Specifies roles and responsibilities relative to planning, testing, oversight administration, and accountability		
4	At what frequency do you test the Disaster Recovery site?		
5	Are test reports documented and approved from concern manager?		

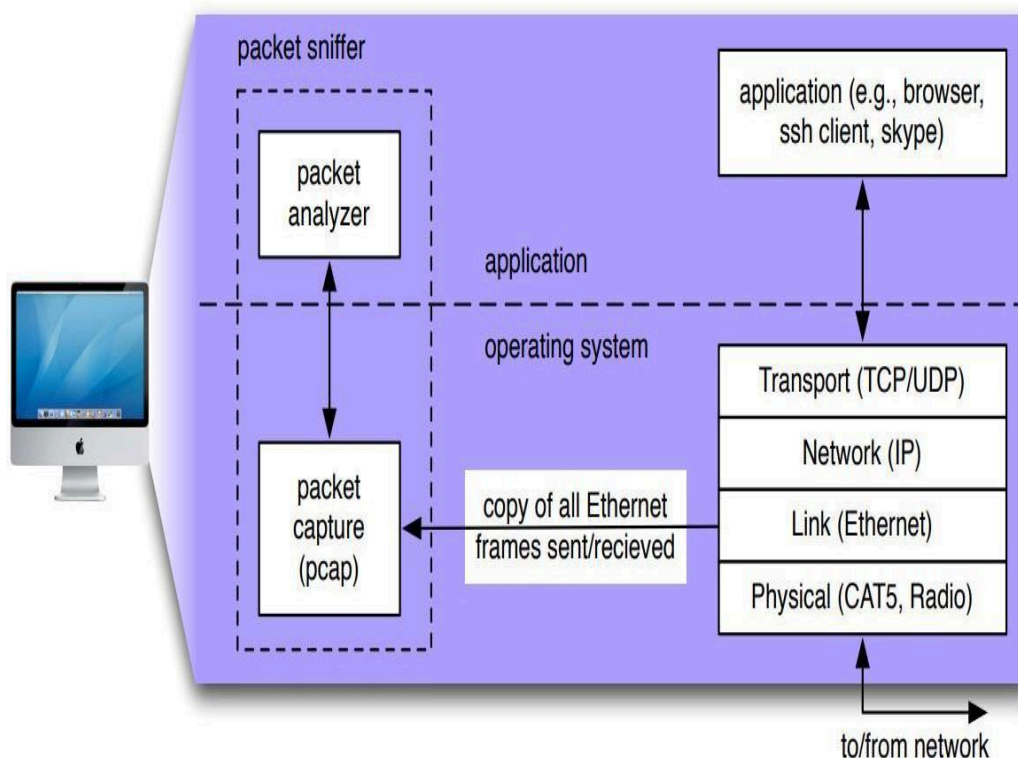
Result:

Experiment-7

Date:

ANALYSIS OF TRAFFIC PACKETS USING WIRESHARK

The basic tool for observing the messages exchanged between executing protocol entities is called a packet sniffer. Packet sniffers are a basic tool for observing the messages on a network. As the name suggests, a packet sniffer captures (“sniffs”) messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/received from/by application and protocols executing on your machine.



The figure above shows the structure of a packet sniffer. At the right are the protocols (in this case, Internet protocols) and applications (such as a web

browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle, is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. As you know, messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In the figure, the assumed physical media is an Ethernet, and so all upper layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/ received from/by all protocols and applications executing in your computer. The existence of the packet capture box in this figure should give you cause to pause and think, particularly down two trains of thought. Firstly, it shows that any packet in a shared medium (Ethernet, Wi-Fi, etc) can be captured and examined without notification of the sender or receiver. You cannot rely on common link-layer protocols to protect your secrets or your privacy online. At a minimum, you should be using encryption protocols (generally buried in the application layer, though sometimes found elsewhere) to protect all network traffic you generate or receive. Secondly, you have the ability to act as the “bad guy” and capture the network traffic of other people, examine it and exploit what you find. You need to learn to use this tool in a responsible fashion. Remember the movie quote: “With great power comes great responsibility!” We will use a filter to ensure Wireshark doesn’t display traffic other than your own, but this is purely a voluntary measure. Please act ethically and responsibly in your use of Wireshark. The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must “understand” the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol. The packet

analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string “GET,” “POST,” or “HEAD,” as shown in Figure 2.8 in the text. We will be using the Wireshark packet sniffer [wireshark.org] for these labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Macintosh, Windows, and Linux/Unix computers. It’s an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support that includes a userguide. It operates in computers using Ethernet, Token-Ring, FDDI, serial (PPP and SLIP), 802.11 wireless LANs, and ATM connections (if the OS on which it's running allows Wireshark to do so).

Procedures

Step-1: Get Wireshark

1. In order to run Wireshark, you will need to have access to a computer that supports both Wireshark and the libpcap or WinPCap packet capture library. The libpcap software will be installed for you, if it is not installed within your operating system, when you install Wireshark. See wireshark.org/download.html for a list of supported operating systems and download sites
2. Download the Wireshark binary from wireshark.org/download.html and install it. Make sure to also download the Wireshark user guide.

3. The Wireshark FAQ has a number of helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.
4. You may need to disable anti-virus protection software (McAfee, I'm looking at you!) before your own IP address will show up in captured data.
5. You should be connected to an Ethernet connection. If you only have WiFi, you'll need to figure out how to set your WiFi physical layer into monitor mode, which may be difficult or impossible, depending on your operating system. Failure to follow this instruction will mean you only see traffic originating or being sent to your own computer, which is sub-optimal for these labs.

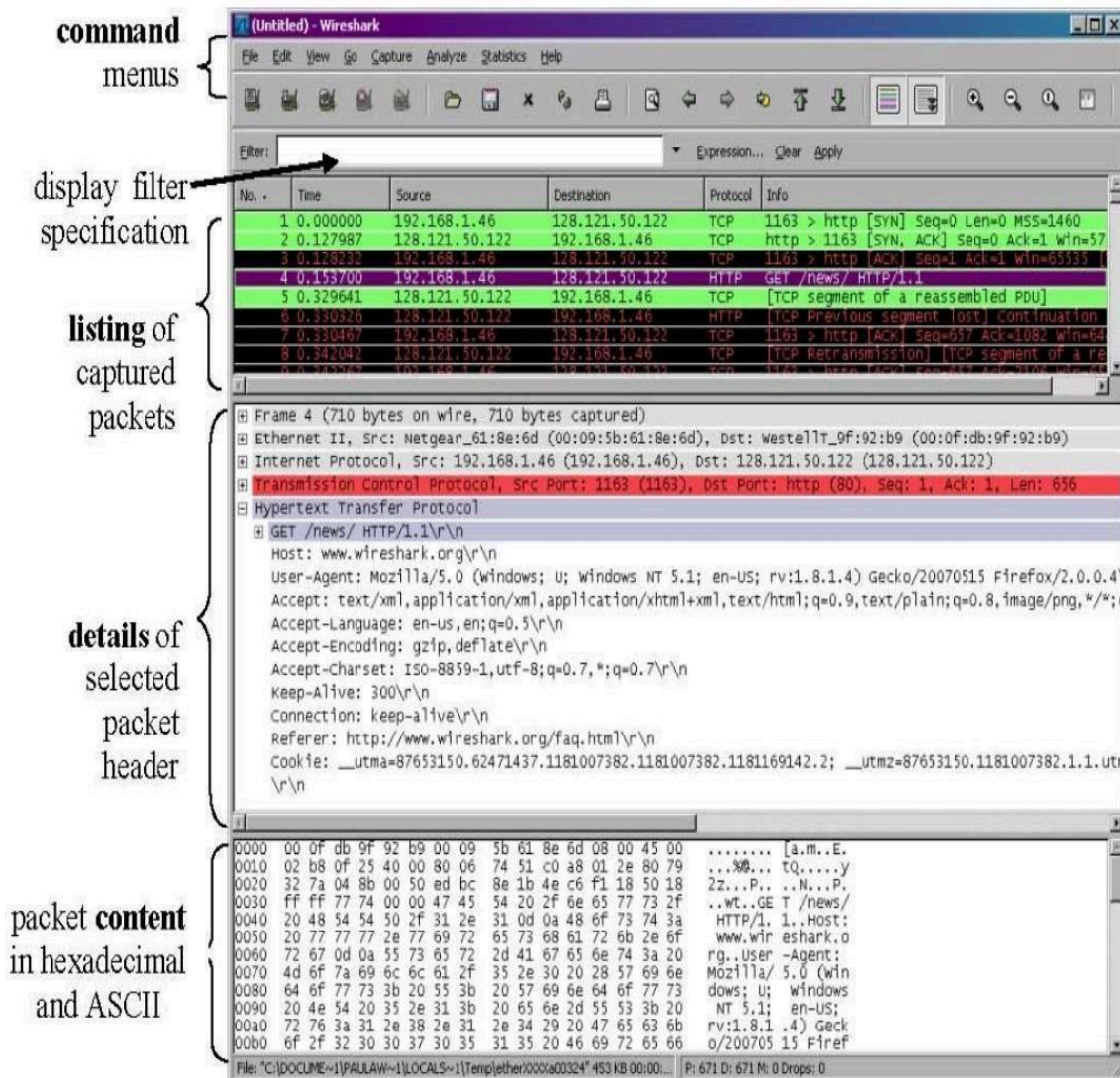
Step-2: Run Wireshark

1. When you run the Wireshark program, the Wireshark graphical user interface will be displayed. Initially, no data will be displayed in the various windows. By the way, the pictures I show in this lab guide may differ, perhaps substantially, from the interface you see on your computer, depending on your installed version and operating system. Be flexible.

Step-3: The Wireshark interface has five major components:

1. The **command menus** are standard pulldown menus located at the top of the window or in your menu-bar. Also included is a toolbar (shown in the figure). Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.
2. The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can

be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.



3. The **packet-header details window** provides details about the packet selected (highlighted) in the packet listing window. (To select a packet in the packet listing window, place the cursor over the packet's one-line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded

or minimized by clicking on the plus-or-minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.

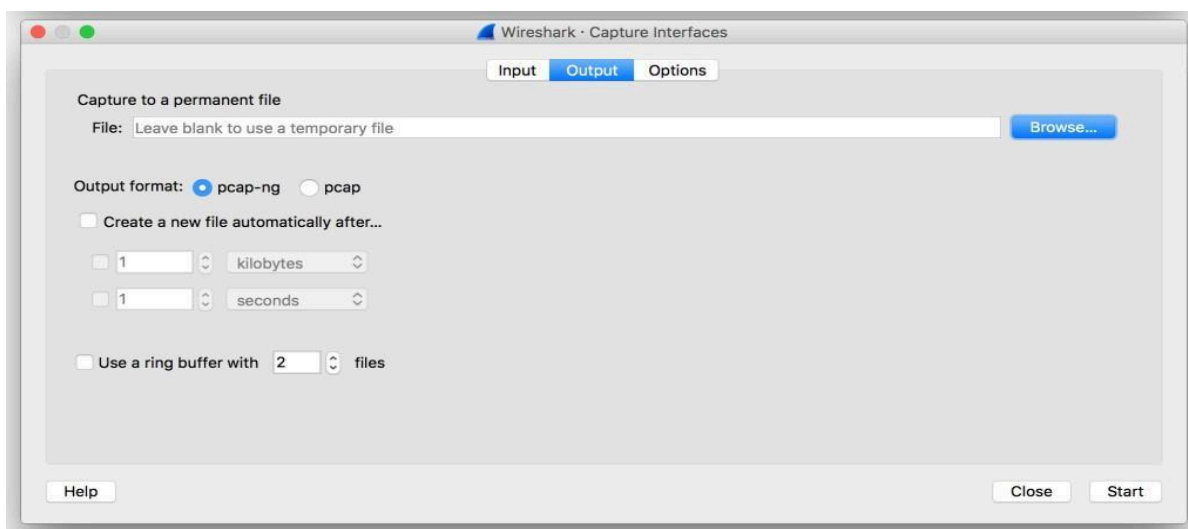
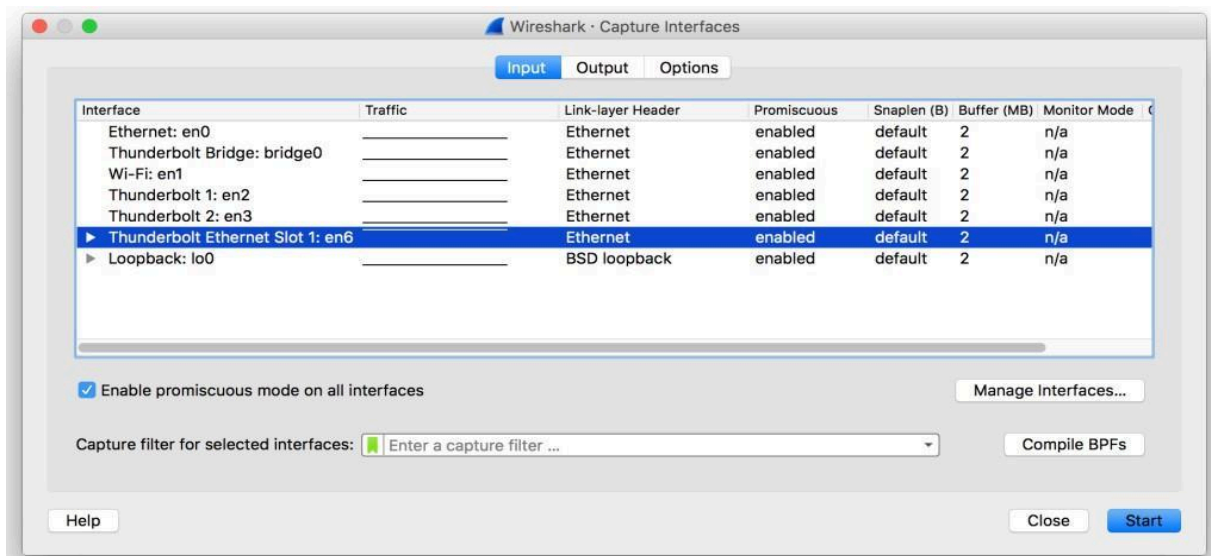
4. The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

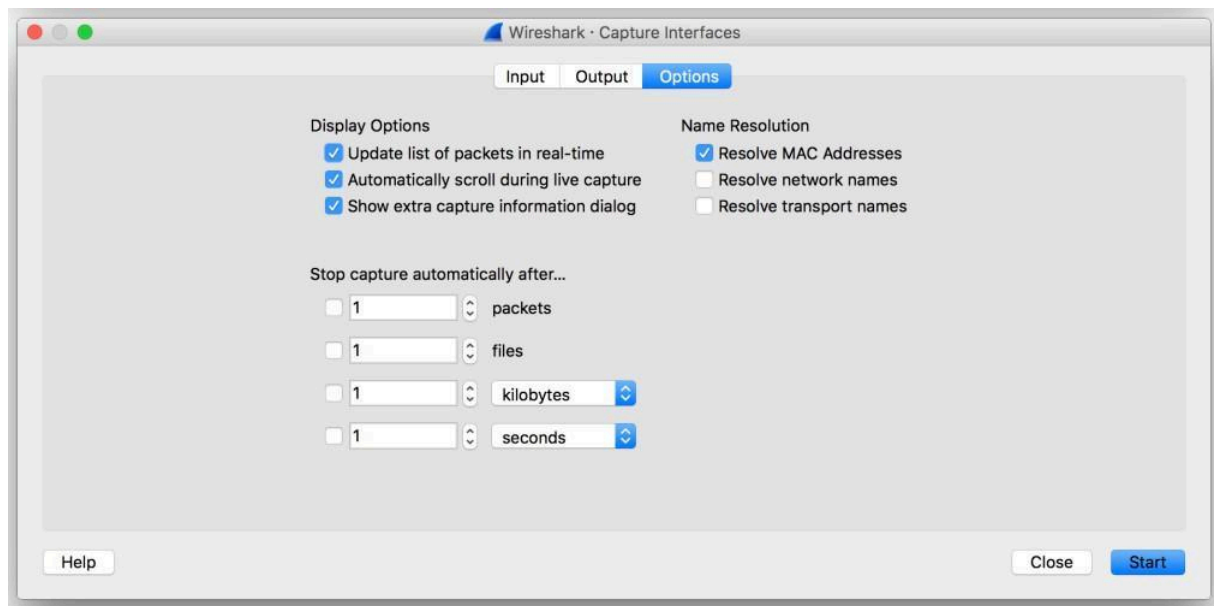
5. Towards the top of the Wireshark graphical user interface, is the **packet display filter field**, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

Step-4: Take Wireshark for a “Test Run.” The best way to learn about any new piece of software is to try it out! Do the following:

1. Start up your favorite web browser, which will display your selected homepage.
2. If you are using a proxy (especially a host-based one), disable it if possible. You want to examine uncached network traffic.
3. Start up the Wireshark software. You will initially see a window similar to that shown above, except that no packet data will be displayed in the packet-listing, packet header, or packet-contents window, since Wireshark has not yet begun capturing packets.
4. To begin packet capture, select the Capture pull down menu and select Options. This will cause the “Wireshark: Capture Interfaces” window to be displayed. There are three sections to this window: Input, Output and Options, as shown below. The Input window allows you to select which interface you will

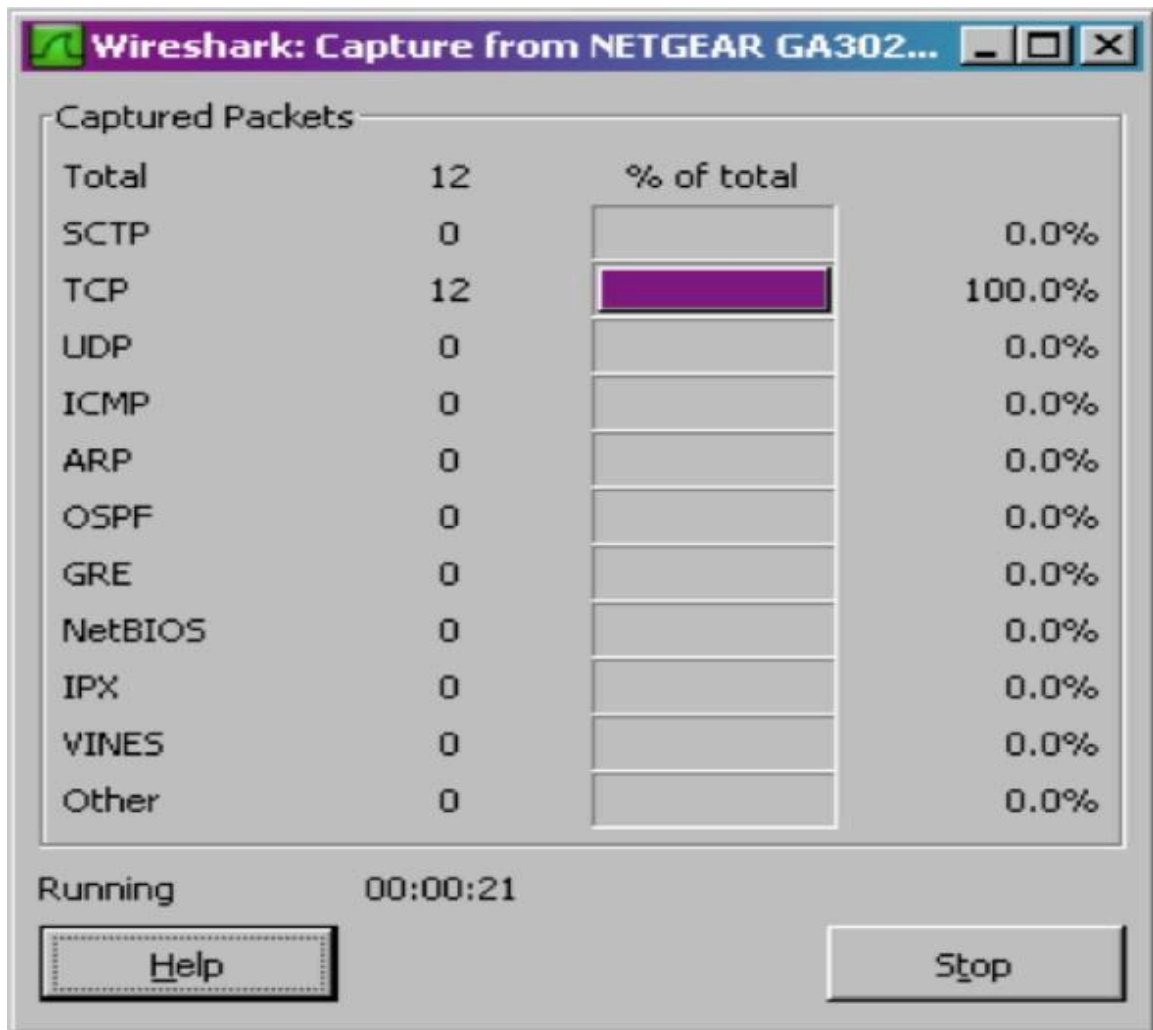
use for capture. You can see that the computer where I took this screenshot has Wi-Fi and a bunch of Ethernet interfaces, as well as the loopback interface. Only one of them is in use, so I'll pick that one. The Output window lets you choose to dump all the collected packets in to a file. This is handy for scripting (wouldn't you love to grab a 1MB capture file at midnight every night? Who wouldn't?) Note that you can limit the file sizes. I generally don't touch anything in this window. The Options window lets you specify when the capture should quit (in packets, files, size or time), controls the listing section of the main window during the capture (update or not? Scroll or not?) as well as choose to resolve names or not.





5. You can use most of the default values in the Options window, but check “Show extra capture information dialog.” The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the Interface pull down menu at the top of the Capture Options window. In case your computer has more than one active network interface (e.g., if you have both a wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets. After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin - all packets visible to your network interface (including those being sent/received from/ by your computer) are now being captured by Wireshark!

6. {Note: I can't get this window to appear on new versions of WireShark. You might, on your version and your operating system} Once you begin packet capture, a packet capture summary window will appear. This is the window that you decided not to hide in the previous step. This window summarizes the number of packets of various types that are being captured, and (importantly!) contains the Stop button that will allow you to stop packet capture. Don't stop packet capture yet.



7. While Wireshark is running, enter the URL <http://www.ece.cmu.edu/~ini740/Lab0/lab0.html> (Those are three zeros, not the letter o) and have that page displayed in your browser. Make sure to clear your browser cache if you have previously displayed this webpage -- you want to get it across the internet, not from your cache. In order to display this page, your browser will contact the HTTP server at www.ece.cmu.edu and exchange HTTP messages with the server in order to download this page in the text. The Ethernet frames containing these HTTP messages will be captured by Wireshark.

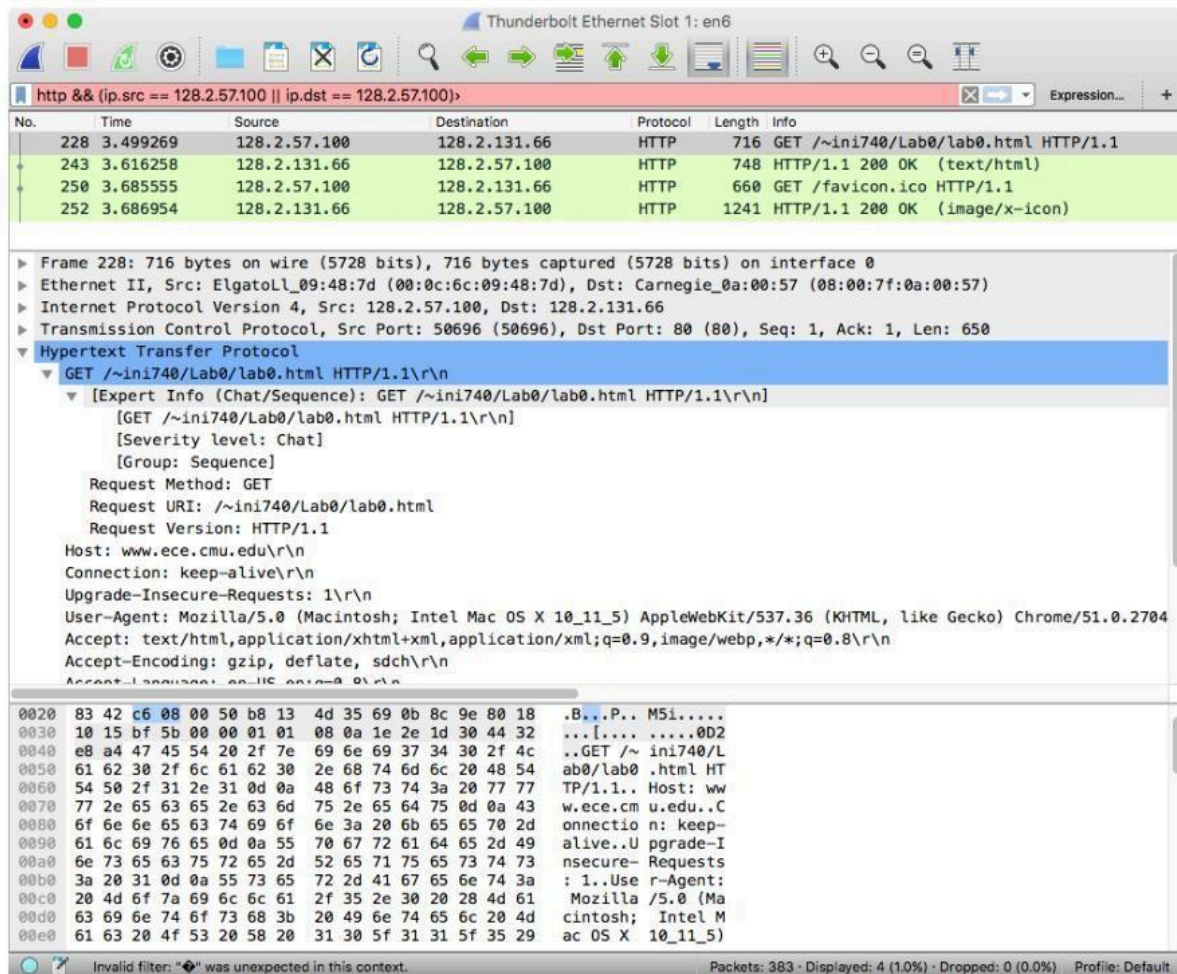
8. After your browser has displayed the lab0.html page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the

Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. The main Wireshark window should now look similar to the figure on page 3. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the `www.ece.cmu.edu` web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well. Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user (as well as data sent via various protocols by other computers on your network). We'll learn much more about these protocols as we progress through the course! For now, you should just be aware that there is often much more going on than “meet's the eye”!

9. Type in `http` (all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select “Apply” in the filter toolbar. This will cause only HTTP message to be displayed in the packet-listing window. Add the filter `ip.src == || ip.dst ==` to filter out traffic that isn't going to or from your computer. This will keep other people's traffic private and get rid of lots of HTTP exchanges from other computers that you don't care about. Filters are combined with C operators. For example, if your IP address is `169.1.19.87`, then your filter should be `http && (ip.src == 169.1.19.87 || ip.dst == 169.1.19.87)`.

10. Select the first `http` message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer to the `www.ece.cmu.edu` HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window.³ By clicking the triangles to the left side of the packet details window, minimize the amount of

Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. Maximize the amount information displayed about the HTTP protocol. Your Wireshark display should now look roughly like this figure. (Note, in particular, the minimized amount of protocol information for all protocols except HTTP, and the maximized amount of protocol information for HTTP in the packet-header window).



11. To use Wireshark effectively, you need to learn how to filter the results so you aren't wading through too much data. Wireshark uses two different filters, one to filter the results that get captured and another to filter the results that are displayed. Unfortunately, both use different languages to specify the filter. You've already been introduced to display filters, which use a C-like set of operators. You can also use a more English-like term to describe the same

operators. For instance, the filter you used earlier `http && (ip.src == 169.1.19.87 || ip.dst == 169.1.19.87)` can also be specified as `http and (ip.src eq 169.1.19.87 or ip.dst eq 169.1.19.87)`. Another powerful operator you should know about is “contains” which, you might have guessed, does a substring match. The actual values being combined can come from any of the protocols and any of the protocol fields that Wireshark knows about (called “dissector” in Wireshark lingo). So, you might search for HTTP traffic from Macintosh computers with `http.user_agent contains AppleWebKit`. Take a look at the Wireshark User Manual, section 6.3-6.5 for more details about Display filters.

12. The capture time of each packet is quite important, so is displayed in the packet listing area as the second column. By default, this time is "number of seconds since the beginning of capture." However, you have control over what is displayed. Explore the View → Time Display Format menu to see display formats as well as precision choices. Also, of interest is the ability to change the time reference so that all times are displayed relative to the capture time of a chosen packet. First, chose a packet from the display list by clicking on it. Then, go to the Edit → Set/Unset Time Reference, which will toggle your choice to use the chosen packet as the reference. When set, you will see the time for that packet changed to "*REF*" All other packet's time has been changed to seconds before or after the capture of that reference packet. This is a particularly handy way to figure out round-trip-time. Set the request packet as the reference, then find the reply packet. The time given on that packet will be the number of seconds it took fom the request packet for it to arrive. No arithmetic necessary!

13. The display filter language is also used to define rules that Wireshark uses to assign colors to particular packets in the user interface. Take a look in Chapter

10.3 of the Wireshark User Guide to learn about coloring rules. Using the captured packets, practice temporary color changes by selecting a packet and then pressing 1, 2, etc. Also, examine the coloring rules dialog and experiment

with defining permanent coloring rules (you might want to export the default set of coloring rules before messing around with them).

14. Capture filters are also quite useful. They let you restrict the amount of data you collect in the first place. Whereas display filters don't actually change the contents of the data that Wireshark collects, merely which of the packets that have been captured are displayed. Capture filters are entered in the "Filter" field of the "Capture Options" dialog box. The capture language is based on tcpdump and requires a bit more protocol knowledge to use. For now, simply experiment with host to ensure you don't capture data from other network users.

15.Exit Wireshark

RESULT:

Experiment-8

IMAGE STEGANOGRAPHY

Aim: To hide a secret message within a cover-media in such a way that others cannot discern the presence of the hidden message.

Algorithm:

Step 1: Start

Step 2: Input: Cover_Image, Secret_Message, Secret_Key;

Step 3: Transfer Secret_Message into Text_File;

Step 4: Zip Text_File;

Step 5: Convert Zip_Text_File to

Binary_Codes; **Step 6:** Convert Secret_Key

into Binary_Codes; **Step 7:** Set Bits Per Unit to

Zero;

Step 8: Encode Message to

Binary_Codes; **Step 9:** Add by 2 unit for

bits Per Unit; **Step 10:** Output:

Stego_Image;

Step 11: End

CODE:

```
import cv2

import string

import os

d={}

c={}

for i in range(255):

    d[chr(i)]=i

    c[i]=chr(i)

x=cv2.imread(r"C:\Users\TCS\Desktop\img.jpg")

i=x.shape[0]

j=x.shape[1]

print(i,j)

key=input("Enter key to edit(Security Key) : ")

text=input("Enter text to hide : ")

kl=0

tln=len(text)

z=0

n=0

m=0

l=len(text)

for i in range(l):
```

```

x[n,m,z]=d[text[i]]^d[key[kl]]

n=n+1

m=m+1

m=(m+1)%3

kl=(kl+1)%len(key)

cv2.imwrite("encrypted_img.jpg",x)

os.startfile("encrypted_img.jpg")

print("Data Hiding in Image completed successfully.")

#x=cv2.imread("encrypted_img.jpg")

kl=0

tln=len(text)

z=0

n=0

m=0

ch = int(input("\nEnter 1 to extract data from Image : "))

if ch == 1:

    key1=input("\n\nRe enter key to extract text : ")

    decrypt=""

    if key == key1 : for i in range(l):

        decrypt+=c[x[n,m,z]^d[key[kl]]]

        n=n+1

```

```
m=m+1 m=(m+1)%3
kl=(kl+1)%len(key)
print("Encrypted text was : ",decrypt)
else:
print("Key doesn't matched.")
else:
print("Thank you. EXITING.")
```

Output:

RESULT:

ANALYSIS OF MALWARE**Aim:**

To write a yara script to detect spyeye, a type of malware file.

Description:

YARA is the name of a tool primarily used in malware research and detection. It provides a rule-based approach to create descriptions of malware families based on textual or binary patterns. A description is essentially a YARA rule name, where these rules consist of sets of strings and a boolean expression. The language used has traits of Perl compatible regular expressions. YARA by default comes with modules to process PE, ELF analysis, as well as support for the open-source Cuckoo sandbox.

Algorithm:

1. Fill the meta section with author name, description of file and version of script.
2. In strings section, fill either text strings or hexadecimal string
3. Specify condition for detecting the malware based on the strings and filesize.
4. If no output comes then spy-eye is not found
5. Else spyeye malicious file detected by yara.

Yara Script:

```
rule spyeye : banker
```

```
{
```

```
meta:
```

author = "Ben"

description = "SpyEye X.Y memory" date = "2022-05-25"

version = "1.0" filetype = "memory"

strings:

\$g = "bot_version"

\$h = "bot_guid"

condition:

any of (\$g,\$h) and filesize >50000

Output:

```
[root@localhost Downloads]# ll malware.exe
```

```
-rw-r--r--. 1 root root 148480 May 26 11:17 malware.exe
```

```
[root@localhost Downloads]# yara spyeye.yara malware.exe
```

```
spyeye malware.exe
```

RESULT:

Experiment-10

Date:

N-STALKER

Aim:

To find out the web application security using N-Stalker tool.

Description:

N-Stalker is a leader on Web Application Security Assessment technology. It currently develops and maintains N-Stalker Web Application Security Scanner suite, a software product aimed on scanning and finding security vulnerabilities in Web Applications. It can play significant role in application security testing. This is trusted when it comes to browser level vulnerabilities. Some of the features are-

- HTTP Fingerprinting
- Parallel Web Crawling
- Server-side technology discoverer
- Automatic False Positive Prevention Engine
- Component-oriented Web Crawler
- Component-oriented Scanning Engine
- IDS Evasion Fuzzing Test
- Web form autocomplete mechanism

Algorithm:

Step-1 Open the N-Stalker application

Step-2 Click New Scan

Step-3 Type the web application URL as www.rajalakshmi.org

Step-4 Choose scan policy as manual test and click Next button

Step-5 Click Optimize button

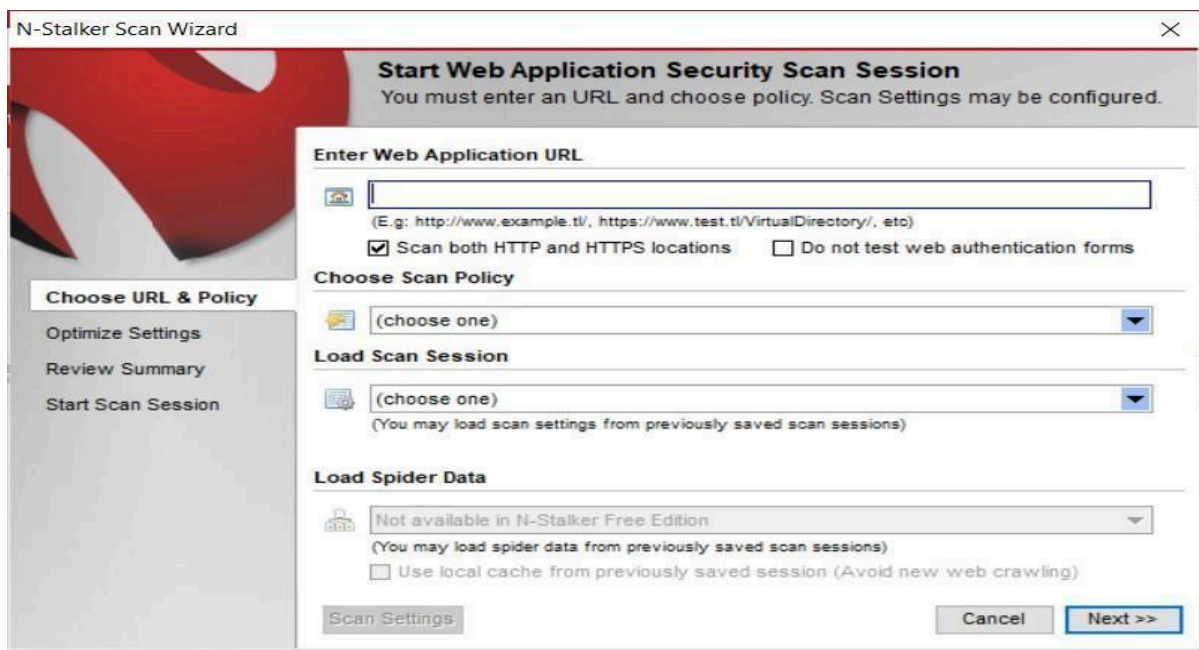
Step-6 Click Start Session button

Step-7 Next press Start Scan

button **Step-8** Save the scan

results.

Output:




The screenshot shows the 'N-Stalker Scan Wizard' window. The title bar reads 'N-Stalker Scan Wizard'. The main heading is 'Start Web Application Security Scan Session' with a subtitle 'You must enter an URL and choose policy. Scan Settings may be configured.' On the left, a sidebar contains a list of steps: 'Choose URL & Policy' (highlighted), 'Optimize Settings', 'Review Summary', and 'Start Scan Session'. The main area contains the following sections: 'Enter Web Application URL' with a text input field and a hint '(E.g: http://www.example.tl/, https://www.test.tl/VirtualDirectory/, etc)'; a checkbox 'Scan both HTTP and HTTPS locations' which is checked, and an unchecked checkbox 'Do not test web authentication forms'; 'Choose Scan Policy' with a dropdown menu showing '(choose one)'; 'Load Scan Session' with a dropdown menu showing '(choose one)' and a note '(You may load scan settings from previously saved scan sessions)'; and 'Load Spider Data' with a dropdown menu showing 'Not available in N-Stalker Free Edition', a note '(You may load spider data from previously saved scan sessions)', and an unchecked checkbox 'Use local cache from previously saved session (Avoid new web crawling)'. At the bottom, there are three buttons: 'Scan Settings', 'Cancel', and 'Next >>'.

Start Web Application Security Scan Session

You must enter an URL and choose policy. Scan Settings may be configured.


Enter Web Application URL




(E.g: <http://www.example.tl/>, <https://www.test.tl/VirtualDirectory/>, etc)

☒ Scan both HTTP and HTTPS locations ☐ Do not test web authentication forms

Choose Scan Policy


 Manual Test (Crawl through the URL and standby for manual attack)

Load Scan Session

 (choose one)

(You may load scan settings from previously saved scan sessions)


Load Spider Data

 Not available in N-Stalker Free Edition

(You may load spider data from previously saved scan sessions)

☐ Use local cache from previously saved session (Avoid new web crawling)

Scan Settings Cancel Next >>



Start Scan

Start Proxy

Close Session

Session Control

Threads #

8

Threads Control

Engine & Crawler Settings

URL Restriction Settings

Session Mgmt & Filters

Spider Control

Encode URI (WAF)

Timeout 15

HTTP Control

HTTP Settings

Track Spider

Debug HTTP

Control Options

FP Keyword Filter

False-Positive Control

URL

POLICY

THREADS

Website Tree

Scanner

Dashboard

Site Sequence

Allowed Hosts

Rejected Hosts

Objects

Cookies

Scripts

Comments

Web Forms

E-mails

Broken pages

Hidden Fields

Information Leakage

Vulnerabilities

Scanner Dashboard

Progress Status

Step 1

Spider

Not Tested

Info Gather

Step 3

Run Modules

Not Tested

Sig Scanner

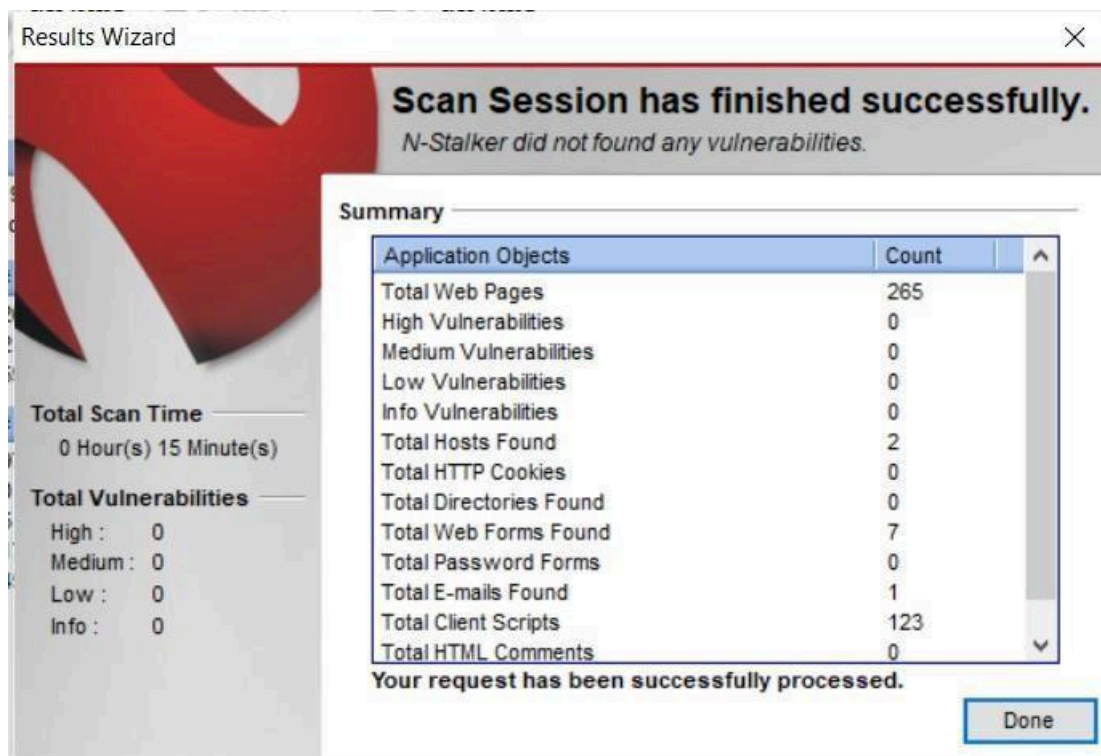
Progress Details

Scan Session	
Start Time	Jun 9, 2022 11:27:43
Duration	0 Hours 0 Minutes

Spider Engine	
Crawled URLs	0
Crawled Hosts	2
Default Page Size	0

Scan Engine	
Total Requests	7
Failed Requests	0
Attacks Sent	5
404 Errors	1
302 Redirection	5

Network	
Bytes Sent	2,067
Bytes Received	91,586
Avg Response Time	0.04 s
Avg Transfer Rate	3.36 Mb/s
Requests/Minute	0



RESULT: