

CSEC 744: Network Security

Name : Shriram Karpoora Sundara Pandian Course

Title : DHCP, ARP, and IP

Lab : Free Style

(I am using Cisco Packet tracer for my lab and not using physical device.)

Rogue DHCP server attack

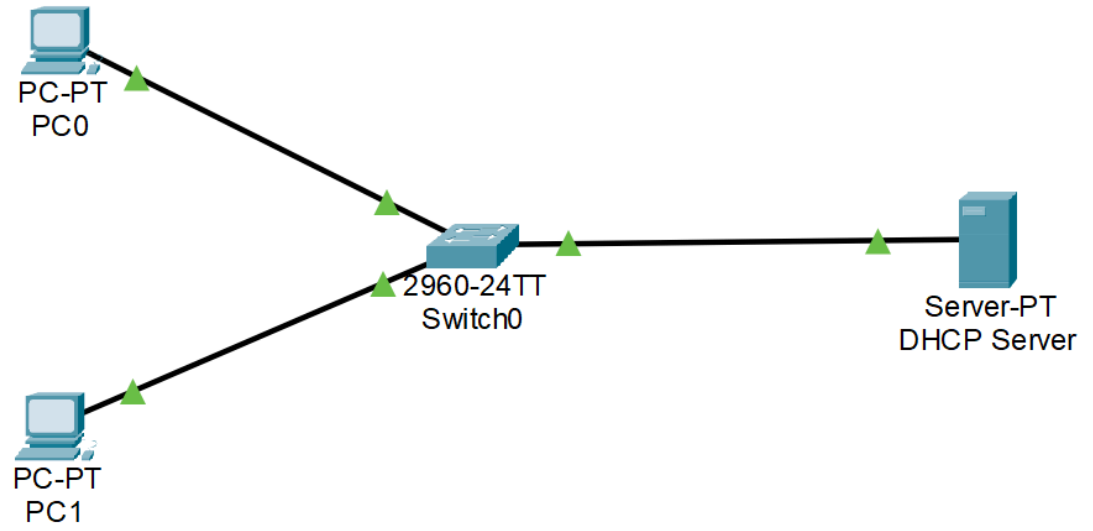
1. What is rogue DHCP server attack?

DHCP servers are responsible for assigning IP addresses to the client machine, where the aim of the attacker is to setup a fake DHCP server that will distribute the spoofed IP address, default gateway, and also the DNS servers to the victim machine, and it becomes the middle machine in between the client and the actual destination. Any traffic from the client machine will reach the default gateway, which is, in this case, the spoofed IP address of an attacker, and the attacker will forward the packets to the actual destination server and monitor all the packets.

The attacker has now become the man in the middle and can listen to all the sensitive information. Also, the DNS server is also spoofed here, so the client web request can take through malicious servers and websites, which is a bigger exploit for the victim (the client).

2. Demo

Step 1 : This is the basic setup for the DHCP server.



Step 2 : IP address for legitimate DHCP server

DHCP configuration for the server

The screenshot shows the 'Server0' configuration window with the 'Services' tab selected. The 'DHCP' service is enabled. The configuration details are as follows:

Interface	Service	Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Maximum Number of Users	TFTP Server	WLC Address
FastEthernet0	On	serverPool	192.168.5.10	0.0.0.0	192	255	15	0.0.0.0	0.0.0.0

Buttons: Add, Save, Remove

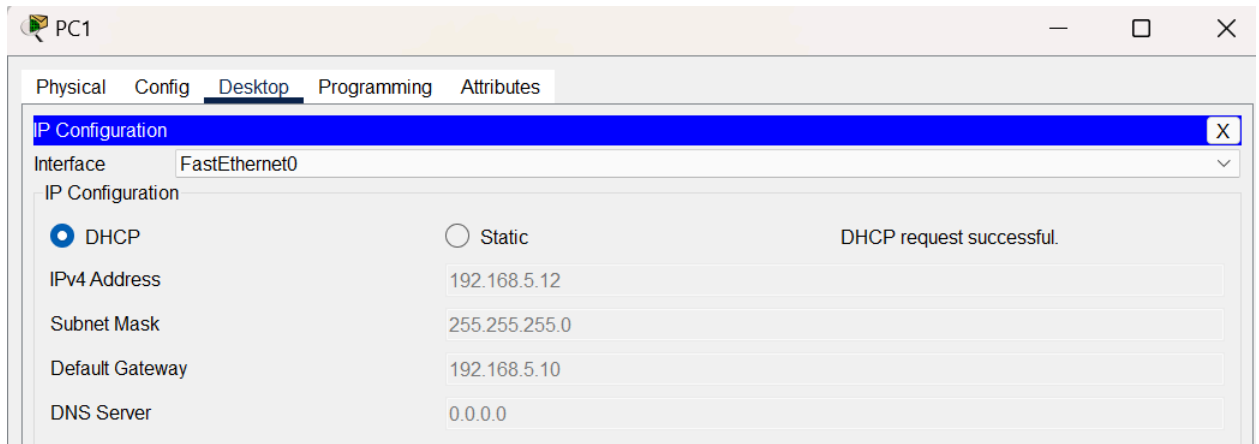
Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool	192.168.5.10	0.0.0.0	192.168.5.11	255.255.2...	15	0.0.0.0	0.0.0.0

Step 3 : From the PC, request the IP address through DHCP

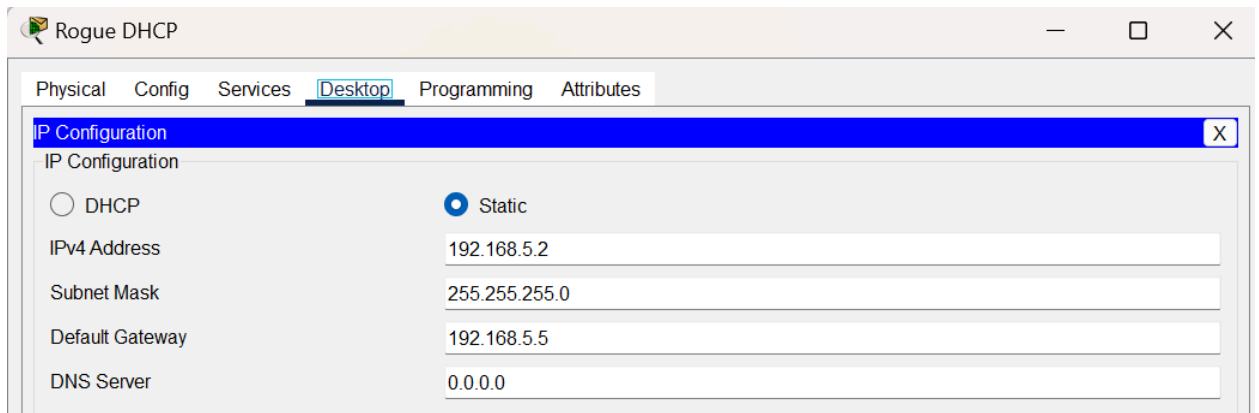
The screenshot shows the 'PC0' configuration window with the 'Desktop' tab selected. The 'IP Configuration' window is open, showing the following settings:

Interface	IP Configuration	IPv4 Address	Subnet Mask	Default Gateway	DNS Server
FastEthernet0	DHCP	192.168.5.11	255.255.255.0	192.168.5.10	0.0.0.0

Status: DHCP request successful.



Step 4 : Configuring and connecting the rogue DHCP server by attacker.



Default Gateway for the Rogue DHCP server :
192.168.5.5

Rogue DHCP

Physical Config **Services** Desktop Programming Attributes

SERVICES

- HTTP
- DHCP**
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

DHCP

Interface: FastEthernet0 Service: ☒ On ☐ Off

Pool Name: serverPool

Default Gateway: 192.168.5.5

DNS Server: 0.0.0.0

Start IP Address: 192 168 5 11

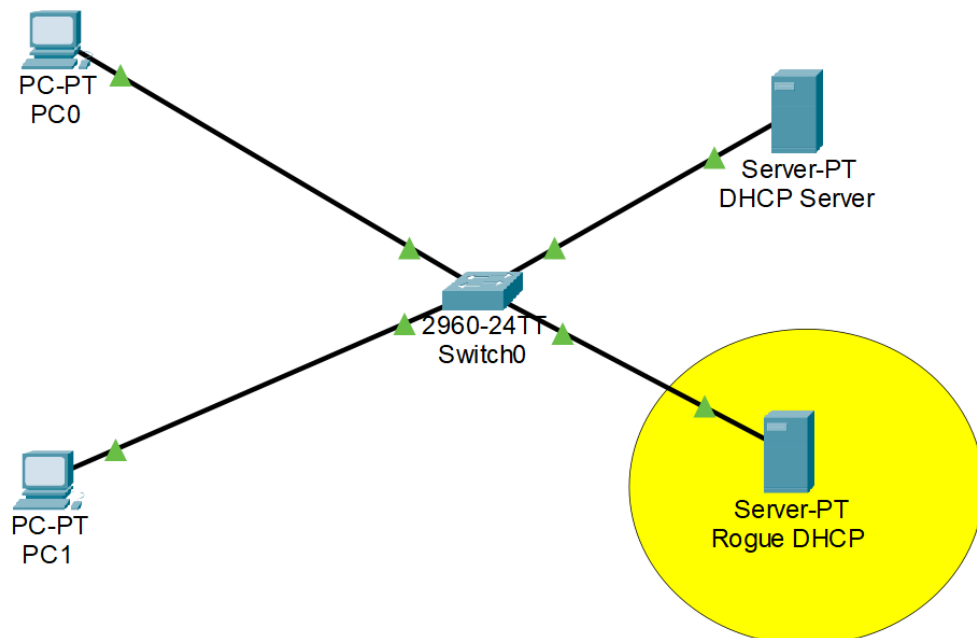
Subnet Mask: 255 255 255 0

Maximum Number of Users: 15

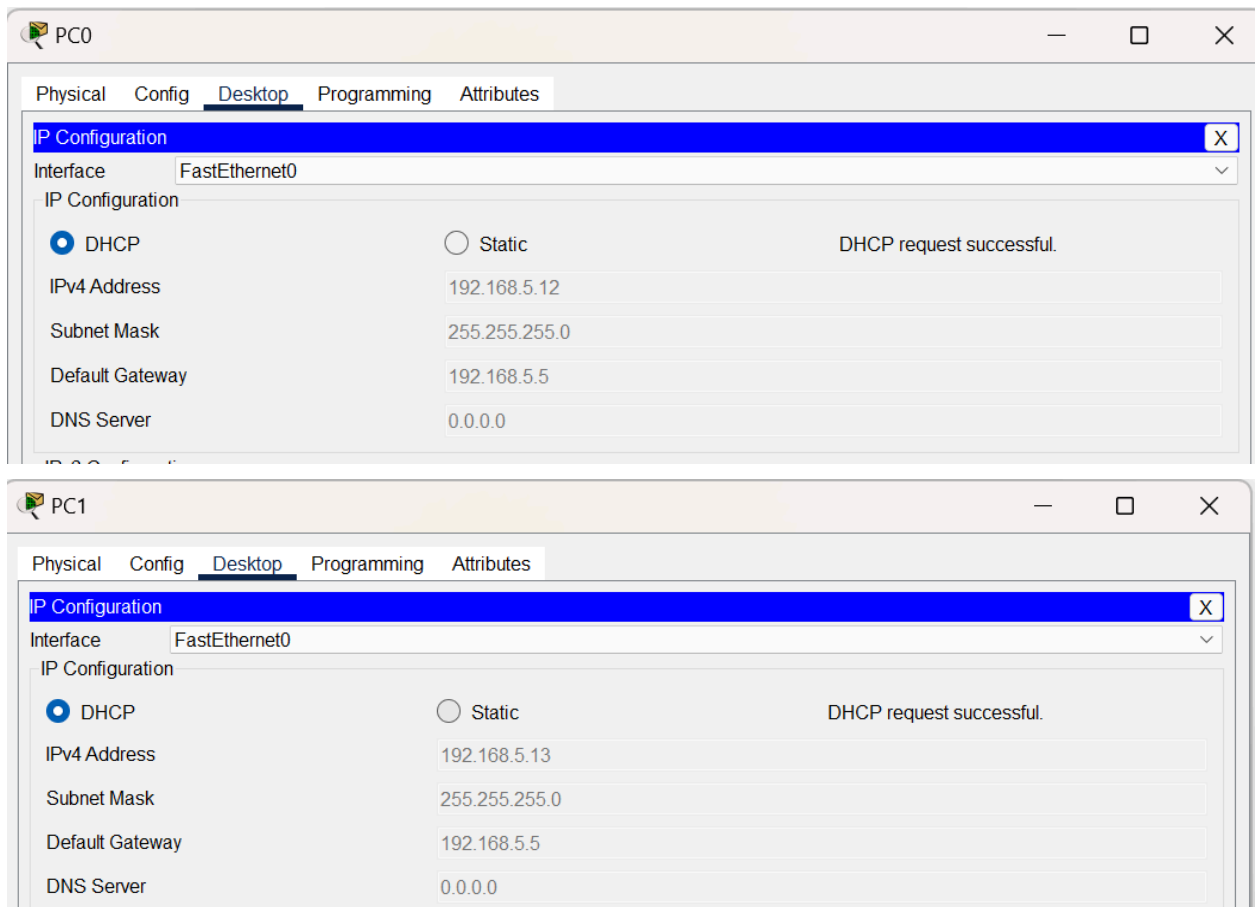
TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool	192.168.5.5	0.0.0.0	192.168.5.11	255.255.2... 0	15	0.0.0.0	0.0.0.0



Step 5 : Now requesting IP from PC0,1 again.



Now we can see that the default gateway of both the pc's are the default gateway configured by rogue server.

So the rogue DHCP has access to all the packets of PC's

3. Mitigation through DHCP snooping

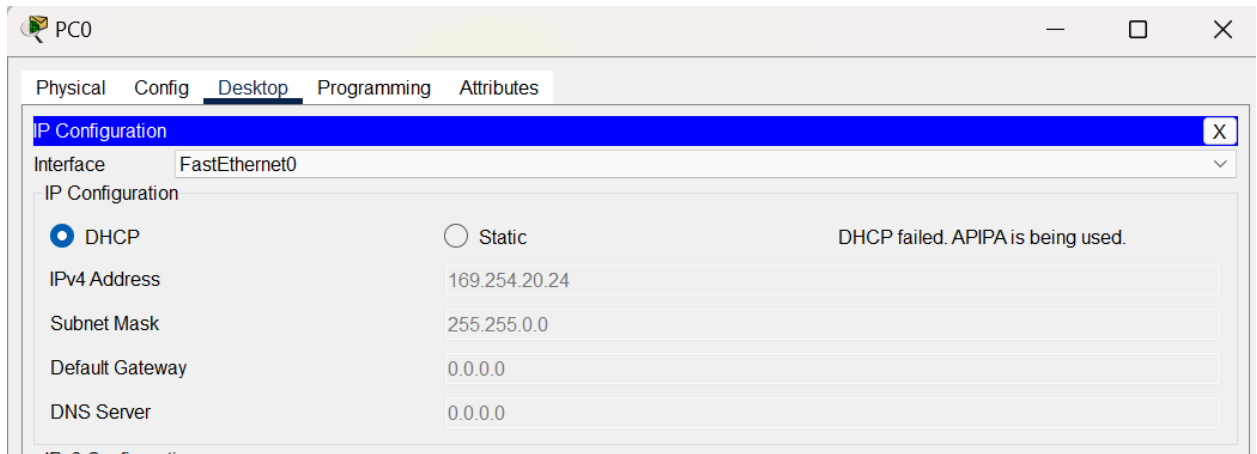
We can mitigate the Rogue DHCP through DHCP snooping which makes the ports in the system as trusted or untrusted, the client machine will only trust the response from DHCP server from trusted ports only, also DHCP snooping table will have all the ports information regarding it is trusted and the information of the device connected to that port. DHCP snooping table is very useful for other mitigations too. So it plays a vital role in layer 2 mitigations.

Demo for the mitigation :

Step 1 : First I need to enable dhcp snooping on the switch where the Pc's are connected :

```
Switch>
Switch>enable
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#ip dhcp snooping
Switch(config)#ip
```

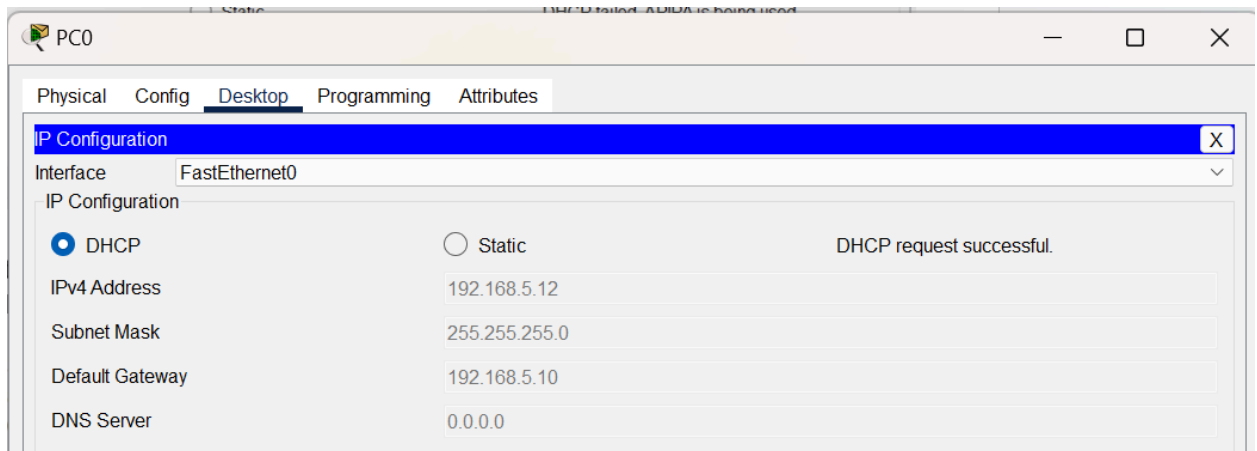
Step 2 : Now all the ports are untrusted by default, so if i try to request the dhcp request, it will fail inside the PC :



Step 3 : We have to now make the Legitimate DHCP server as trusted port in CLI of switch so that i will be able to assign IP for PCs

```
Switch#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
1
Insertion of option 82 is enabled
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface                Trusted      Rate limit (pps)
-----
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fa0/3
Switch(config-if)#ip dhcp snooping trust
Switch#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
1
Insertion of option 82 is enabled
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface                Trusted      Rate limit (pps)
-----
FastEthernet0/1          no          unlimited
FastEthernet0/3          yes         unlimited
FastEthernet0/4          no          unlimited
Switch#
```

Step 4 : Now lets try DHCP request from PC0, it should be successful



It is successful and also the default gateway is gateway of original DHCP server.

Step 5 : We can also see the binding table where it have records of mac address of PC's that requested the DHCP server from the trusted port.

```
Switch#show ip dhcp snooping binding
-----
MacAddress      IPAddress      Lease(sec)    Type           VLAN  Interface
-----
00:07:EC:83:14:18  192.168.5.12  86400        dhcp-snooping  1     FastEthernet0/1
00:60:47:77:2C:9B  192.168.5.13  86400        dhcp-snooping  1     FastEthernet0/2
Total number of bindings: 2
```

This is the mitigation for Rogue DHCP server using DHCP snooping.

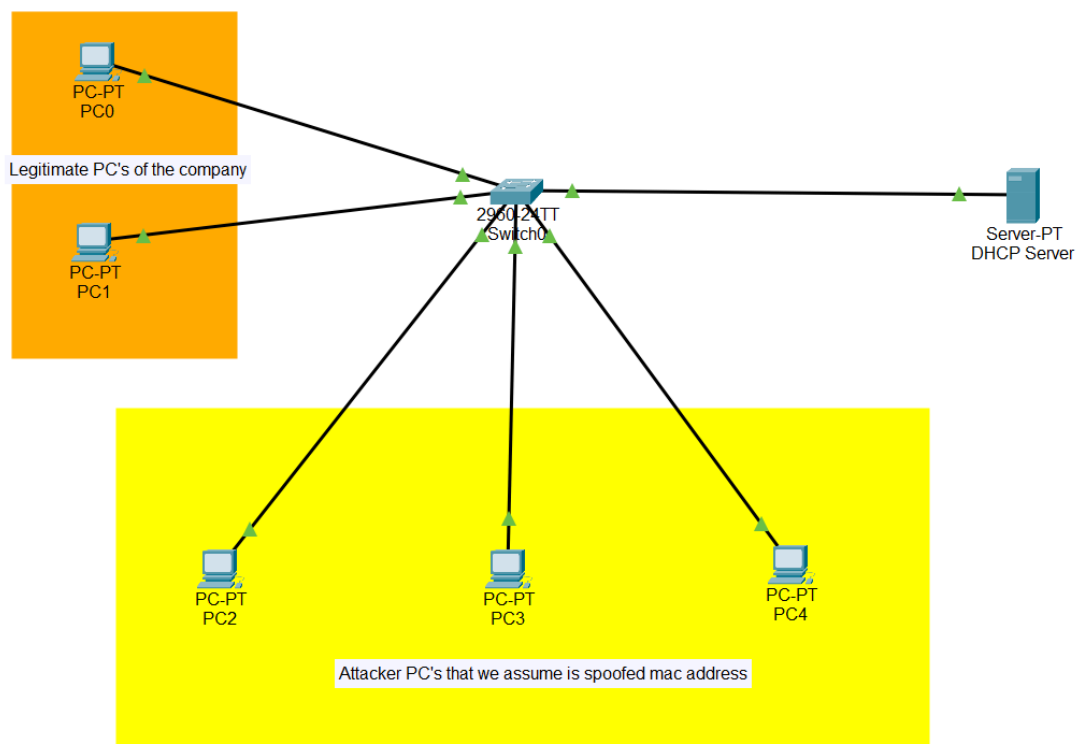
DHCP starvation attack

1. What is DHCP starvation attack ?

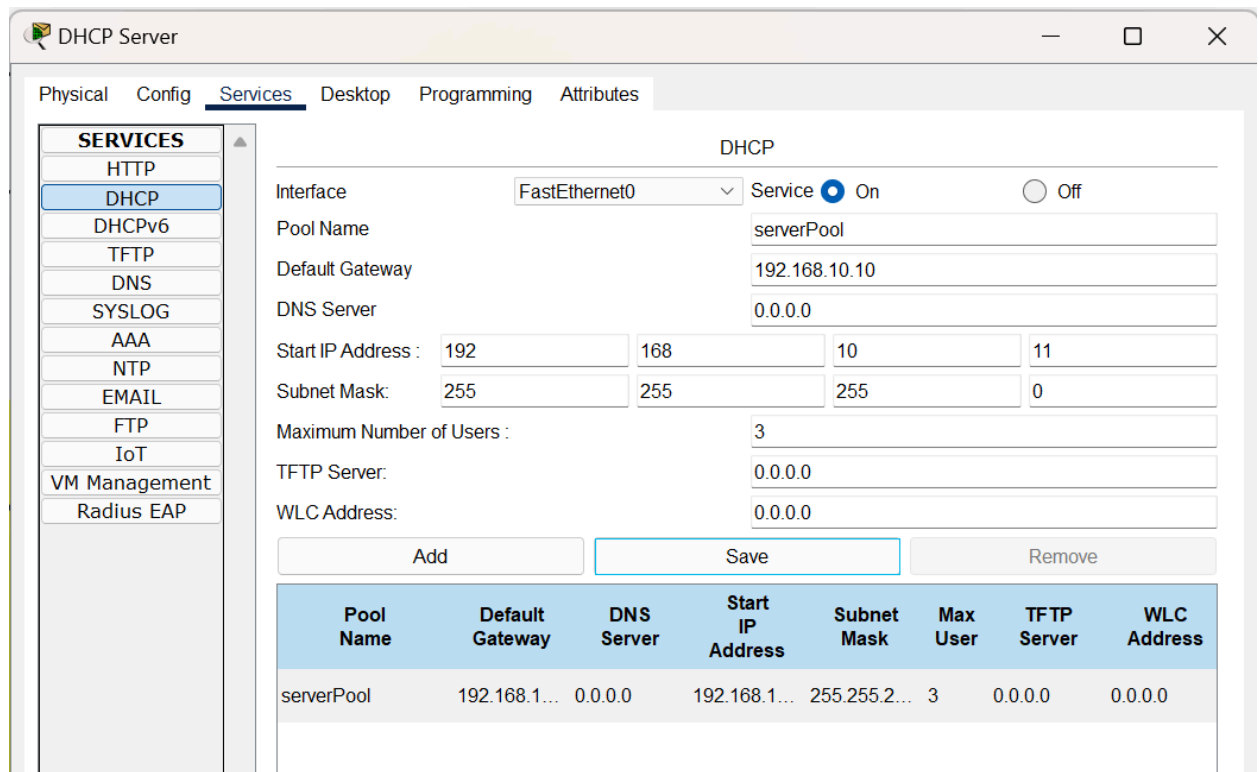
This is very interesting attack and as the name says, it makes DHCP servers to starve in providing the IP address, what really happens is attacker will find a way to communicate with DHCP server first and then he will have pool of spoofed mac address and lot of DHCP discover request is generated for request IP address for these fake Mac address that attacker has been generated, and eventually this discover request will flood the whole network and DHCP server will be out of IP address and finally it won't have any IP left that can be assigned to new device that is trying to connect to this network, so now it became a some kind of denial of service, and DHCP server no longer function properly as it can't assign a new IP address.

2. Demo

Step 1 : The overall outlook of the organization, here as I am using CISCO packet tracer, I am considering the 3 attacker PC's are spoofed MAC address.



Step 2 : Configure the DHCP server with pool of 3 IP addresses.



SERVICES

- HTTP
- DHCP**
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

DHCP

Interface: FastEthernet0 Service: ☒ On ☐ Off

Pool Name: serverPool

Default Gateway: 192.168.10.10

DNS Server: 0.0.0.0

Start IP Address: 192.168.10.10

Subnet Mask: 255.255.255.0

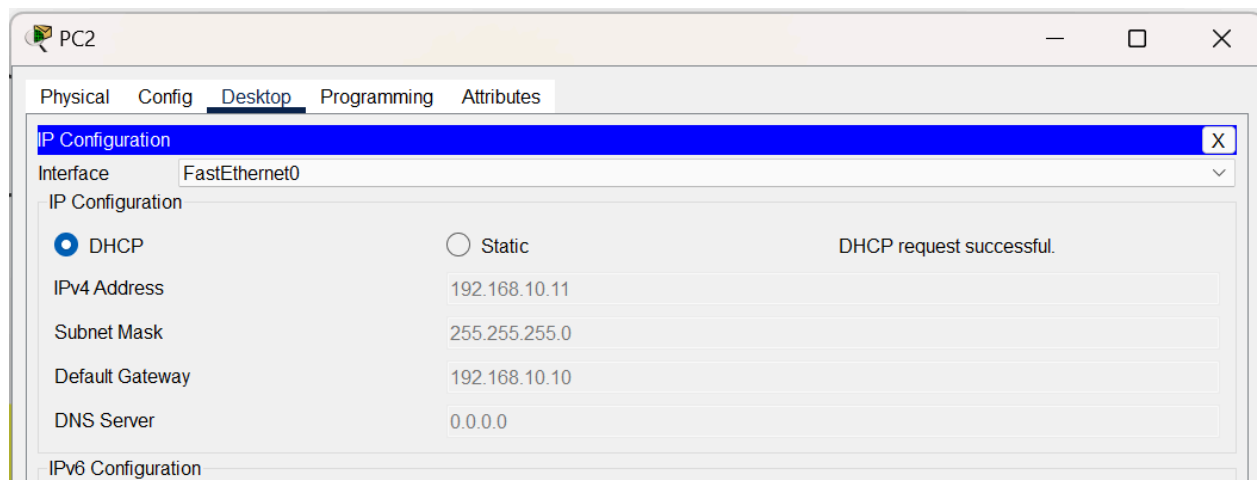
Maximum Number of Users: 3

TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool	192.168.10.10	0.0.0.0	192.168.10.10	255.255.255.0	3	0.0.0.0	0.0.0.0

Step 3 : Now attacker PC's will request DHCP for IP address :
Attacker PC1 request IP :



PC2

Physical Config **Desktop** Programming Attributes

IP Configuration [X]

Interface: FastEthernet0

IP Configuration

☒ DHCP ☐ Static DHCP request successful.

IPv4 Address: 192.168.10.11

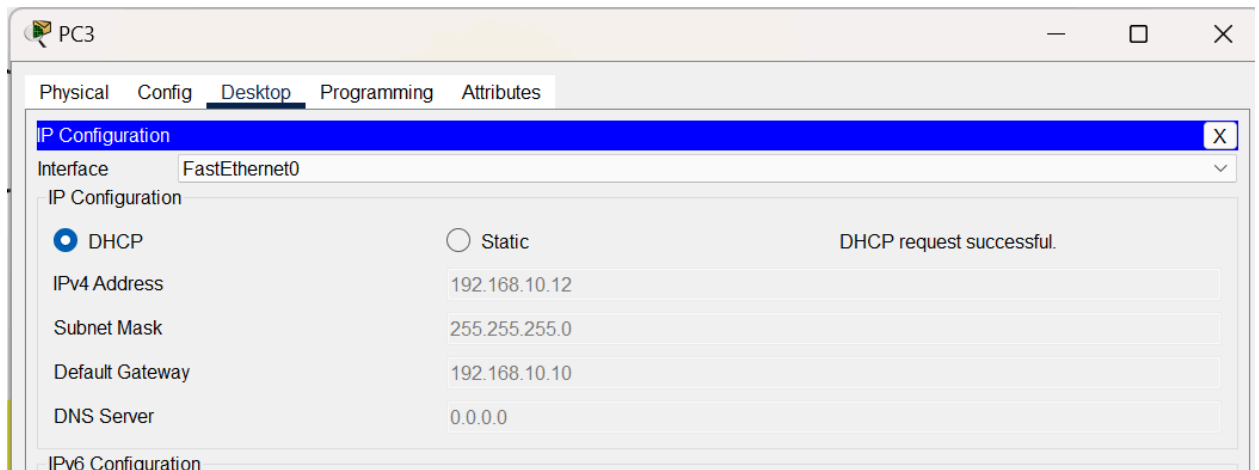
Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.10

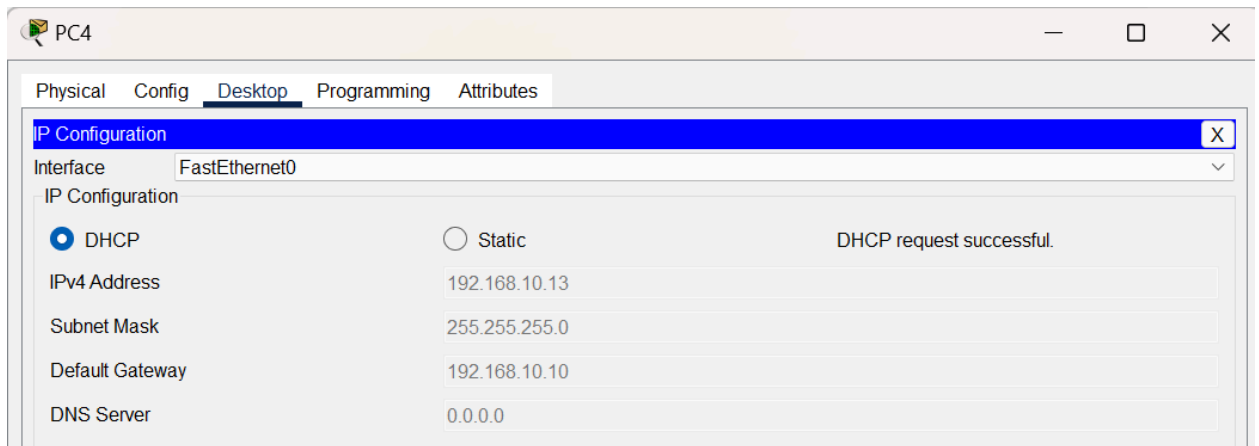
DNS Server: 0.0.0.0

IPv6 Configuration

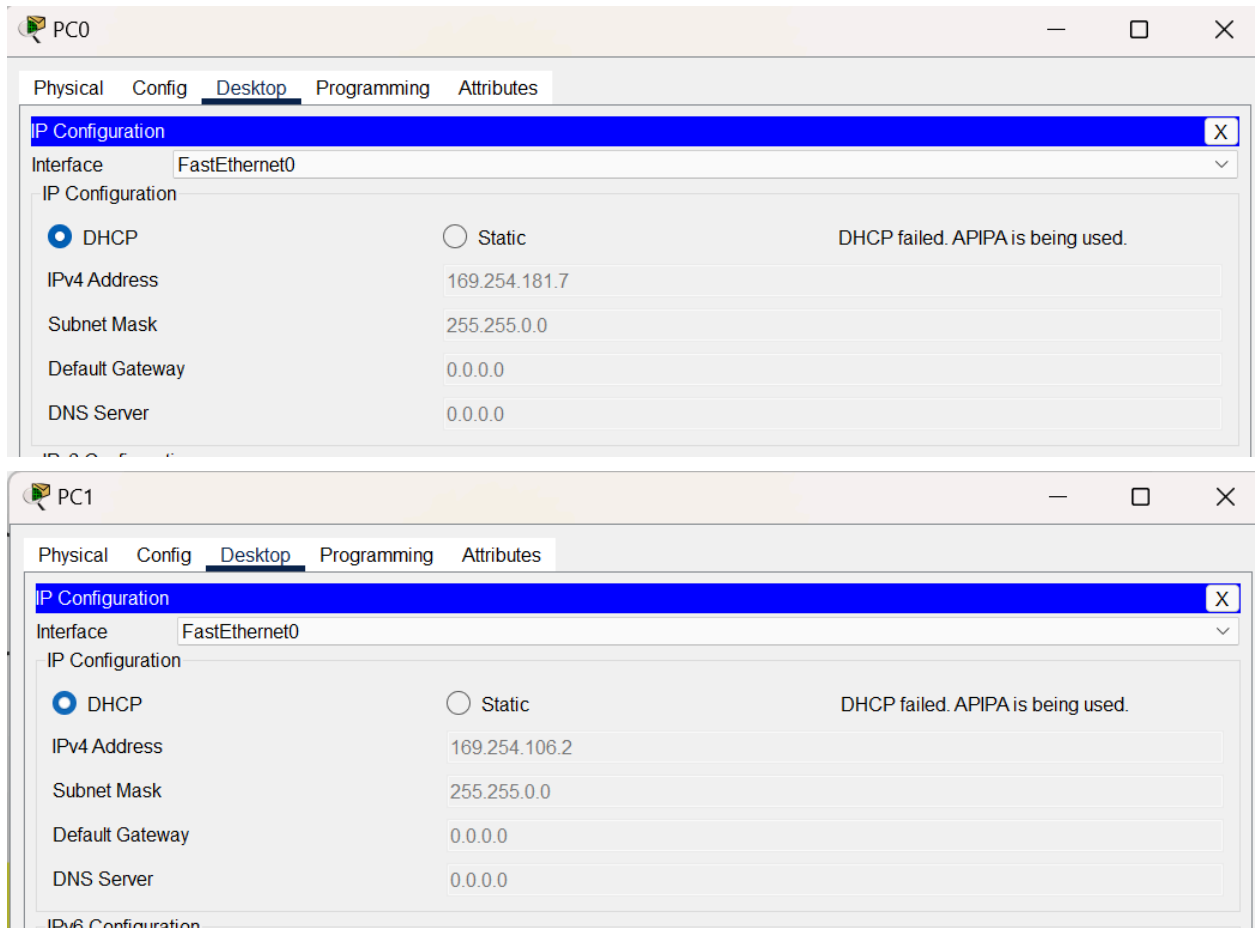
Attacker PC2 Request IP :



Attacker PC3 Request IP :



Step 4 : When Legitimate PC try to request DHCP server it will fail, as the IP address pool of DHCP server is already depleted :



So this is how the attacker floods the server with spoofed mac and depletes the DHCP server pool and now there is no IP left for legitimate PC's to connect to or acquire IP. Kind of a denial of service attack.

3. a) Mitigation implemented with port security

We have a feature in port security to limit the number of mac address in the network, also we can specify the particular MAC address to be connected in the network. So through that we can mitigate the attack, so no longer attacker can request the new IP through spoofed Mac address.

Step 1 : Enable port security on the Switch and configure it with Mac address of Legitimate PC0 :

```
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security maximum 2
Switch(config-if)#switchport port-security mac-address 0060.5C9B.B507
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security
```

Step 2 : Enable Port security with the mac address of Legitimate PC1 :

```
Switch(config)#interface fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security maximum 2
Switch(config-if)#switchport port-security mac-address 0040.0B6B.6A02
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security
Switch(config-if)#exit
```

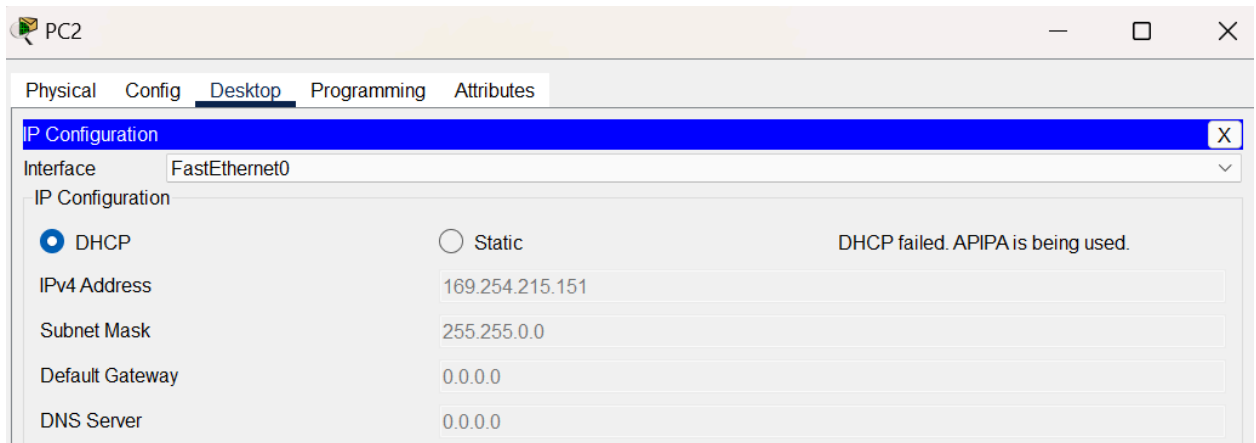
Step 3 : Enabling port security on the other three ports where attacker machines are connected :

```
Switch(config)#interface fa0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security maximum 0
^
% Invalid input detected at '^' marker.

Switch(config-if)#switchport port-security maximum 1
Switch(config-if)#switchport port-security violation shutdown
```

So now the attacker can't spoof more than 1 Mac address, if so the port will shutdown automatically.

Step 4 : Now port security is enabled, now lets try requesting the dhcp server again from attacker PC with different MAC address(spoofed Mac address) this time.



This is how we can use port security to mitigate the starvation attack.

b) Mitigation implemented through DHCP snooping

DHCP snooping has a database or table called binding that is maintained and basically it keeps the MAC address information that is associated with the particular port and also monitors the dhcp traffic. So now if the lot of fake discover request is coming from untrusted ports with different MAC address, it will drop the fake traffic.

Step 1 : Turn on DHCP snooping, all ports will become untrusted.

```
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#ip dhcp snooping
Switch(config)#ip dhcp snooping vlan 1
Switch(config)#show ip dhcp snooping
^
% Invalid input detected at '^' marker.

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

Switch#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
1
Insertion of option 82 is enabled
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface                Trusted      Rate limit (pps)
-----
-----
```

Step 2 : Trust the DHCP server first and other clients are untrusted.

Step 3 : After configuring the snooping, lets do some request to DHCP server and see the binding table.

```

Switch#show ip dhcp snooping binding

```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
00:60:5C:9B:B5:07	192.168.10.12	86400	dhcp-snooping	1	FastEthernet0/1
00:05:5E:16:D7:97	192.168.10.11	86400	dhcp-snooping	1	FastEthernet0/3
00:D0:BC:A1:CA:4B	192.168.10.14	86400	dhcp-snooping	1	FastEthernet0/4
00:01:42:31:7C:AA	192.168.10.15	86400	dhcp-snooping	1	FastEthernet0/5

```

Total number of bindings: 4

```

Step 4 : We can see the MAC address associated with the port. If the MAC address is spoofed and we get a request again from the same port, this time, the DHCP snooping will see the binding table and drop the packet.

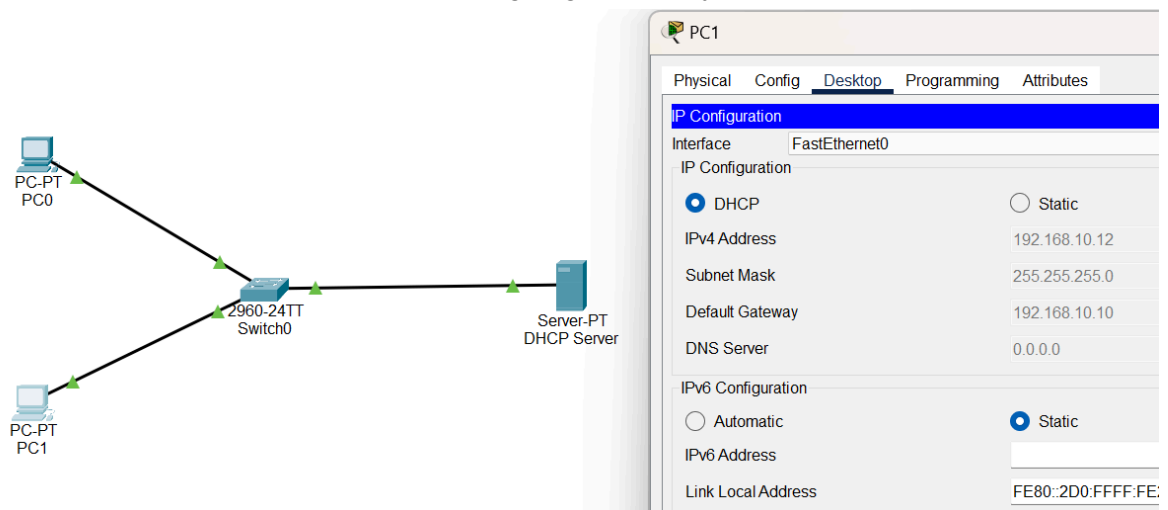
IP Spoofing

1. What is IP Spoofing ?

IP spoofing is a technique where the attacker tries to create communication between the sender and receiver, where the attacker spoofs the packet by forging the source IP address of the packet by the sender's IP address, then sends the packet to the receiver. The receiver will assume that the packet originated from the sender and start sending back the information that the attacker needs. As the spoofed packets have the IP address of the original sender, but attackers send them to the receiver, it makes the attacker anonymous too. It is a basic flaw in the internet that has existed for a long time.

2. Demo : I am using CISCO packet tracer, so I am trying to do the demo identical as possible :

Step 1 : Setup two pc's with DHCP server assigning them a dynamic address.



Step 2 : Enable DHCP snooping on switch and make the dhcp server as trusted port.

```
Switch0/
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#ip dhcp snooping
Switch(config)#ip dhcp snooping vlan 1
Switch(config)#interface fa0/3
Switch(config-if)#ip dhcp snooping trust
Switch(config-if)#exit
Switch(config)#
```


Step 3 : Attacker PC with same IP address as PC0 which is 192.168.10.11

Threat

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

☐ DHCP ☒ Static

IPv4 Address 192.168.10.11

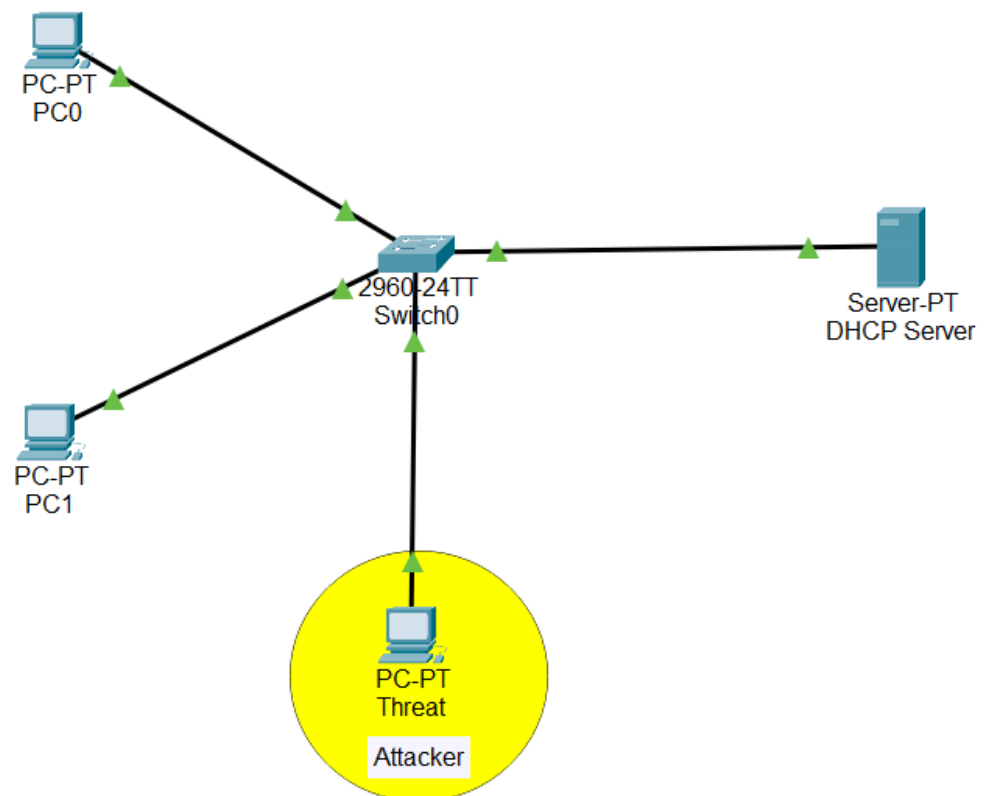
Subnet Mask 255.255.255.0

Default Gateway 0.0.0.0

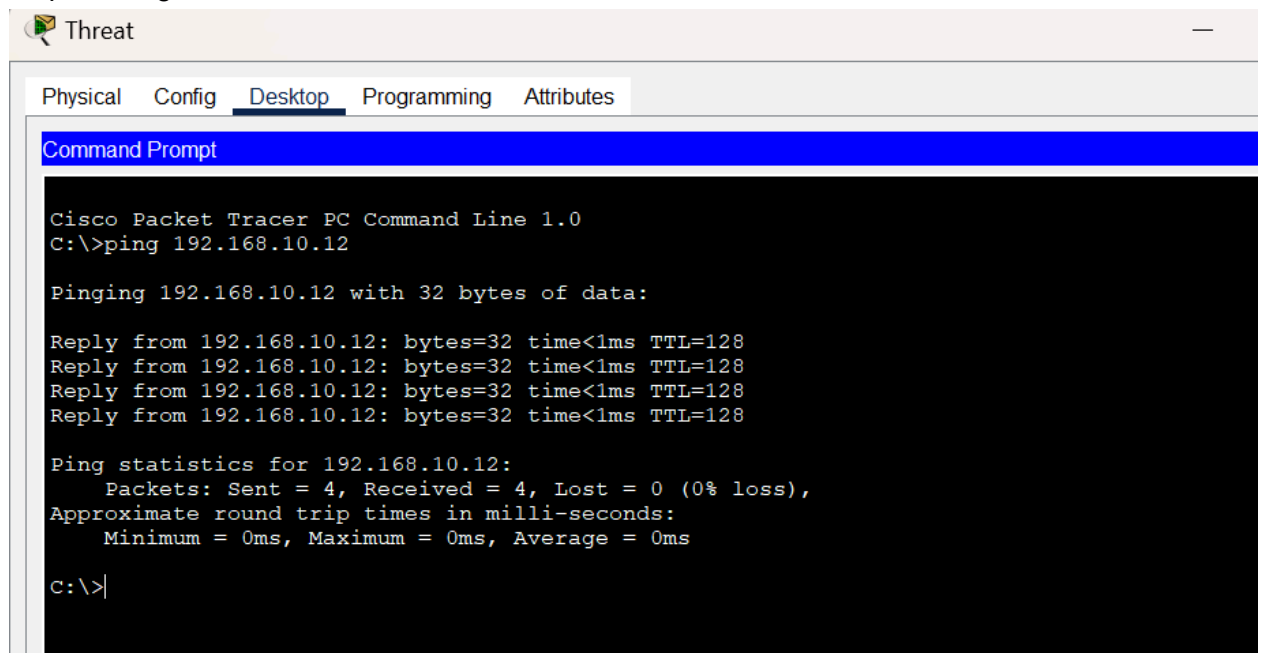
DNS Server 0.0.0.0

IPv6 Configuration

This is how it looks (SETUP)



Step 4 : Ping other PC's on the network :



The screenshot shows a Cisco Packet Tracer PC Command Line window. The window has tabs for Physical, Config, Desktop, Programming, and Attributes, with Desktop selected. The Command Prompt shows the following output:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.10.12

Pinging 192.168.10.12 with 32 bytes of data:

Reply from 192.168.10.12: bytes=32 time<1ms TTL=128
Reply from 192.168.10.12: bytes=32 time<1ms TTL=128
Reply from 192.168.10.12: bytes=32 time<1ms TTL=128
Reply from 192.168.10.12: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

I am able to successfully ping to other PCs on the network.

This is how we can get access to the other PC through the spoofed IP that is legitimate in the network.

3. Mitigation implemented with IP source guard ?

The IP source guard works with the help of the DHCP snooping table because they have records of the IP address on particular ports and they keep these records saved to the Dynamic Port ACL (Access Control List), which will be updated dynamically. Also, the bindings will be maintained for checking the IP source and if the IP address is the same as the legitimate IP that originated from a different port, they will be dropped as they won't match with the records. So the IP source guard will mitigate the IP spoofing by the following: Port security is an additional layer of security for the IP source guard to function effectively.

Below is the screenshot of how to setup IP source guard and how to bind the MAC address with the IP address. It didn't work in packet tracer, but for configuring real switch we can use this commands.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/0/1
Switch(config-if)# ip verify source
Switch(config-if)# exit
Switch(config)# ip source binding 0100.0022.0010 vlan 10 10.0.0.2 interface gigabitethernet 1/0/1
Switch(config)# ip source binding 0100.0230.0002 vlan 11 10.0.0.4 interface gigabitethernet 1/0/1
Switch(config)# end
```

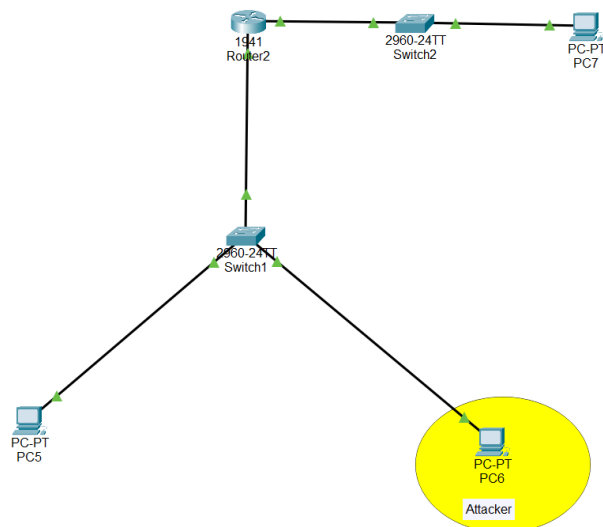
ARP cache poisoning attack

1. What is ARP cache poisoning attack ?

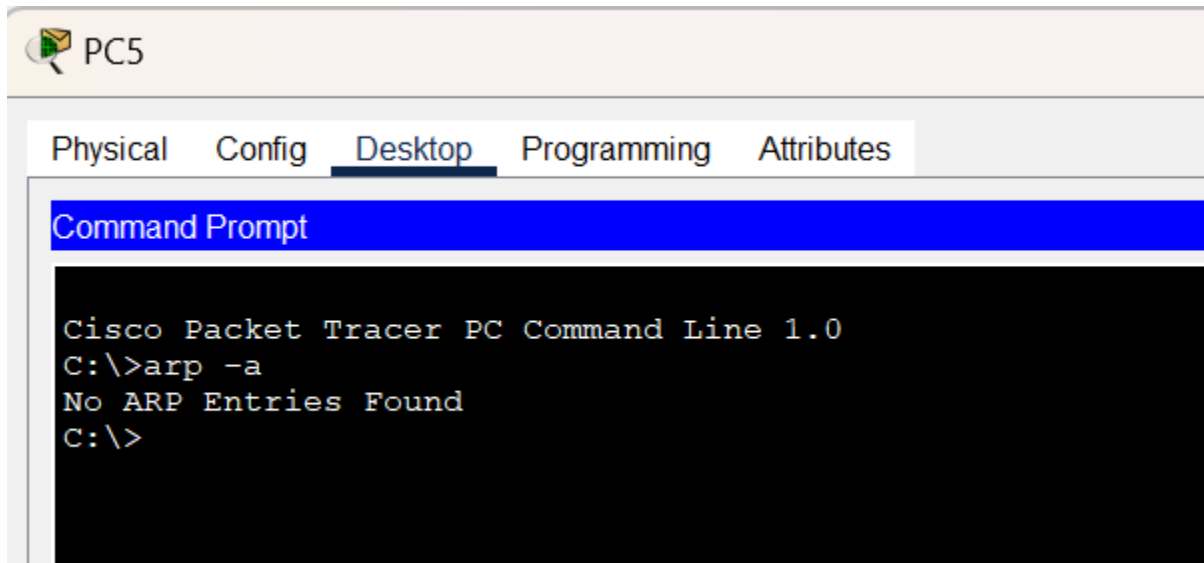
This is an interesting attack, also called ARP spoofing. The intention of the attacker is to poison the ARP values or cache present on the table, where the attacker tries to redirect the traffic to his machine as the man in the middle or collapse the network as a DDoS attack. The attacker can spoof the mac address of the default gateway and will try to redirect all the traffic to his system instead of the gateway. This poisoning is similar to IP spoofing, where we attack ARP tables specifically and their values.

2. Demo :

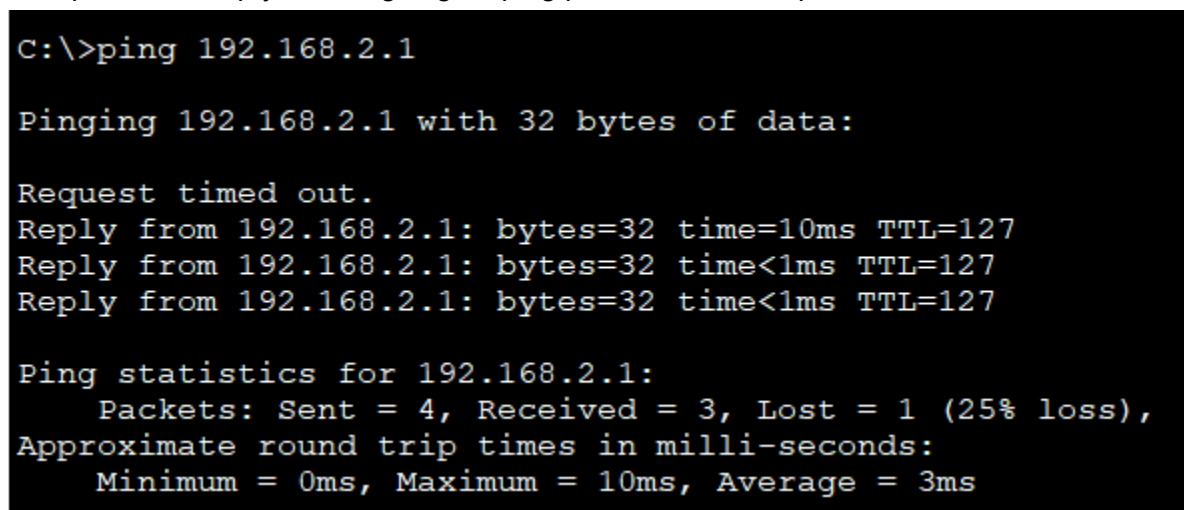
Step 1 : setting up the infrastructure



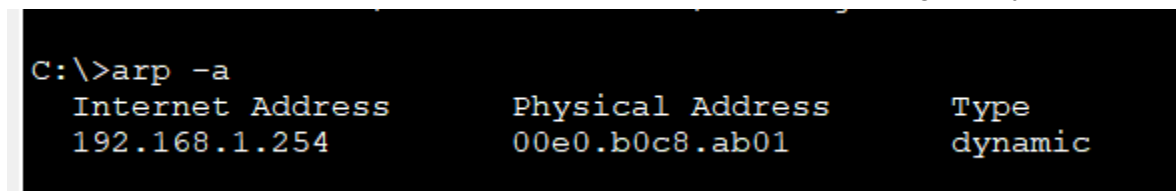
Step 2 : Initially no entries are found on PC5 which is legitimate user :



As arp table is empty we are going to ping pc7 to build the arp table



Step 3 : Now we can see the arp table and mac address of the default gateway :



The MAC address of the default gateway is 00e0.b0c8.ab01. We are going to spoof that address.

We are going to copy the mac address of the router (default gateway)
00E0.B0C8.AB01

Router2

PhysicalConfigCLIAttributes

GLOBAL

Settings

Algorithm Settings

ROUTING

Static

RIP

SWITCHING

VLAN Database

INTERFACE

GigabitEthernet0/0

GigabitEthernet0/1

GigabitEthernet0/0

Port Status

☐ 1000 Mbps

☒ 100 Mbps

☐ 10 Mbps

☒ On

Bandwidth

☐ Half Duplex

☒ Full Duplex

☒ Auto

Duplex

MAC Address00E0.B0C8.AB01

IP Configuration

IPv4 Address192.168.1.254

Subnet Mask255.255.255.0

Tx Ring Limit10

Step 4 : Paste the Mac address of the default gateway to attacker PC and ping from attacker pc to pc 1 (legitimate PC)

```
C:\>ping -t 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time=40ms TTL=128
Reply from 192.168.1.1: bytes=32 time=21ms TTL=128
Reply from 192.168.1.1: bytes=32 time=1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time=25ms TTL=128
Reply from 192.168.1.1: bytes=32 time=2ms TTL=128
Reply from 192.168.1.1: bytes=32 time=44ms TTL=128

Ping statistics for 192.168.1.1:
    Packets: Sent = 9, Received = 9, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 44ms, Average = 14ms

Control-C
^C
```

Step 5 : Now lets look into the arp table of the legitimate PC :

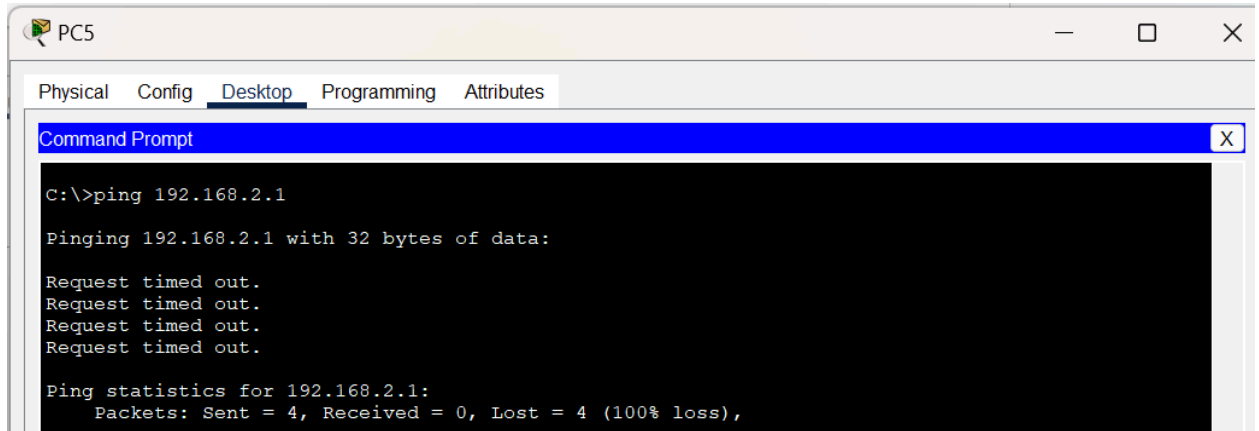
```
C:\>arp -a

Internet Address      Physical Address      Type
192.168.1.2           00e0.b0c8.ab01       dynamic
192.168.1.254         00e0.b0c8.ab01       dynamic
```

That's it game over, the attacker got the MAC address of the default gateway and now all the traffic that the PC (legitimate) user asks for will be redirected to the attacker.

```
1      00e0.b0c8.ab01      DYNAMIC      Fa0/3
Switch#show mac address-table dynamic
      Mac Address Table
-----
Vlan    Mac Address      Type      Ports
----    -
1       0090.2192.20be    DYNAMIC    Fa0/1
1       00e0.b0c8.ab01    DYNAMIC    Fa0/2
Switch#
```

From the switch we can see on port fa0/2 mac address belongs to attacker. Now when we try to ping to the other pc on another network from legitimate PC, the traffic direct to default gateway will fail as we can see below we got request timed out and we will get the information from the attacker sniffer, we can see that destination IP as 192.168.2.1.



The screenshot shows a Windows Command Prompt window titled 'PC5'. The command prompt displays the following text:

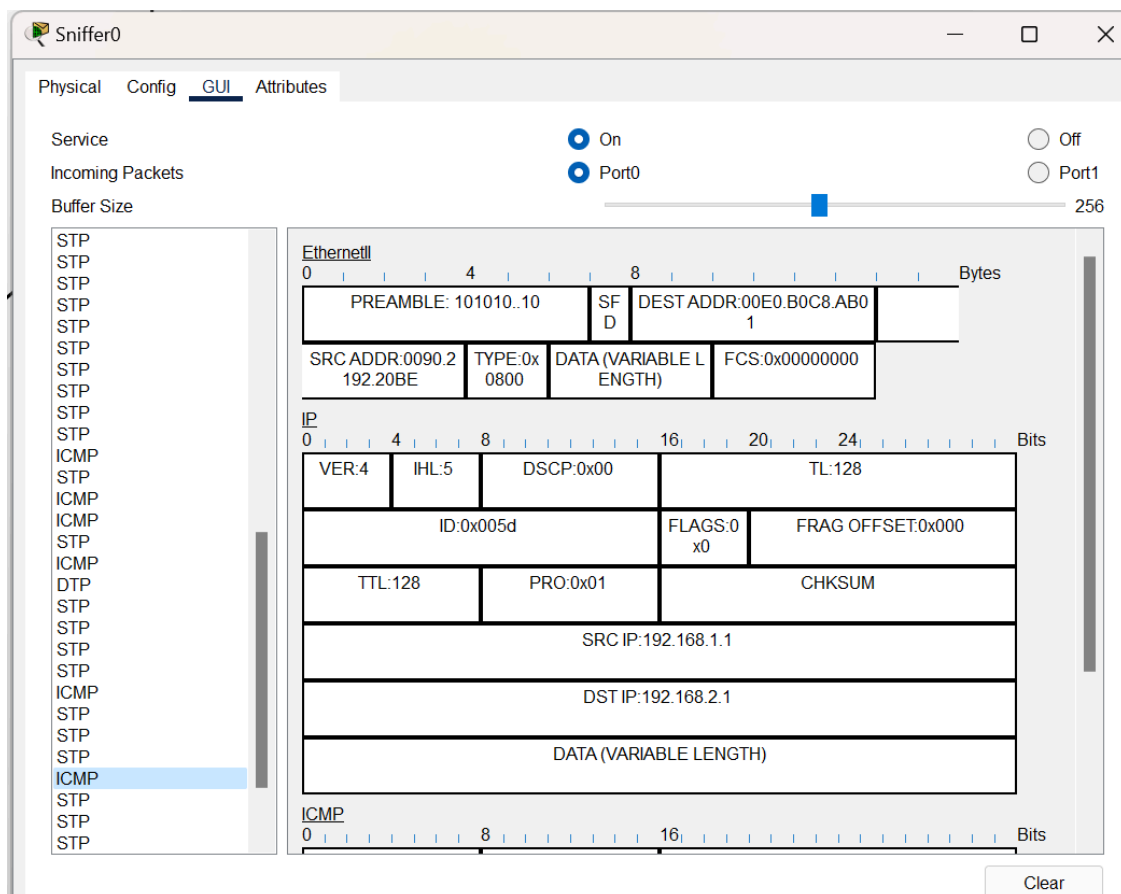
```
C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

We successfully became the man in the middle and redirected the traffic to view the content.



The screenshot shows the Sniffer0 GUI with the 'GUI' tab selected. The 'Service' is set to 'On', 'Incoming Packets' is set to 'Port0', and 'Buffer Size' is set to 256. The packet list on the left shows several 'STP' packets and one 'ICMP' packet, which is highlighted. The details pane on the right shows the structure of the captured packet:

EthernetII

0		4		8		Bytes	
PREAMBLE: 101010..10				SF D	DEST ADDR: 00E0.B0C8.AB01		
SRC ADDR: 0090.2192.20BE		TYPE: 0x0800		DATA (VARIABLE LENGTH)		FCS: 0x00000000	

IP

0		4		8		16		20		24		Bits	
VER: 4		IHL: 5		DSCP: 0x00		TL: 128							
ID: 0x005d				FLAGS: 0x0		FRAG OFFSET: 0x000							
TTL: 128				PRO: 0x01		CHKSUM							
SRC IP: 192.168.1.1													
DST IP: 192.168.2.1													
DATA (VARIABLE LENGTH)													

ICMP

0		8		16		Bits	

Clear

3. a) Mitigation implemented with Dynamic ARP inspection

For proper functioning of the Dynamic ARP inspection, it need to be synchronized with DHCP snooping table again, and the ARP access control lists, which will help to inspect the ARP packets that broadcasted in the network.

Step 1 : We need to enable arp inspection on the switch :

```
Switch>
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#ip arp inspection vlan 1
Switch(config)#
```

Step 2 : Delete the entries of arp table of the attacker :

```
Minimum = 0ms, Maximum = 35ms, Average = 12ms

C:\>arp -a
    Internet Address      Physical Address      Type
192.168.1.1              0090.2192.20be       dynamic
192.168.1.254           00e0.b0c8.ab01       dynamic

C:\>arp -d
C:\>arp -a
No ARP Entries Found
C:\>
```

Step 3 : We unable to ping inside the network, as arp inspection will prevent us from doing that.

```
C:\>arp -d
C:\>ping 192.168.1.254

Pinging 192.168.1.254 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.254:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```


b) Mitigation implemented with private VLANs

The private VLAN isolates the devices so it will prevent outside devices from communicating with each other and eventually stop the falsified ARP request, which will not enter the private VLAN. So no untrusted device can be connected to the devices inside the private VLAN.

I am using Cisco Packet Tracer, but we can configure private VLANs on the real gear.

Step 1: Overall setup with PCs, router, and switch.

Step 2: In the switch, go into enable mode and global configuration mode, where we can configure the Primary and secondary VLANs

Step 3: Create a primary VLAN and make it the private VLAN first.

Step 4: We have to create the secondary VLANs and then bind those secondary VLANs to the primary ones.

Step 5: Assign subnets for secondary VLANs and scale it to the clients or customers.

Step 6: It is difficult for an attacker to send an ARP cache inside the private VLAN because each secondary VLAN will have a separate ARP table and outside machines will be unable to access it.

By doing so, we can mitigate the ARP cache poisoning attack.