Network Troubleshooting Admin Guide

Computer Networks Course Project

Berra SÖYLER

190315046

Date: 11.06.2025

Introduction

IT professionals working in system and network administration often face a variety of technical challenges in their daily operations. These issues may stem from configuration errors or infrastructure shortcomings, and resolving them can often be time-consuming and complex.

This report is intended to offer practical solutions to some of the common problems encountered in the field. While it does not aim to be a comprehensive documentation, it is designed to serve as a helpful guide for professionals who may be facing similar situations.

The main goal of this work is to provide clear and applicable solutions to challenges commonly encountered in system and network administration.

1. Insecure SSH Hardening

Application Layer (Layer 7)

Problem Description:

The default SSH configuration leaves the system vulnerable to several security risks, including brute-force attacks, unauthorized root access, and broad user access. Without proper hardening, attackers can exploit weak credentials, default settings, or open ports to gain unauthorized access to the server. This misconfiguration at the **Application Layer (Layer 7)** poses a serious threat to system integrity and confidentiality.

Resolution Steps

Step 1: Open the SSH Configuration File

```
sudo nano /etc/ssh/sshd_config
```

Step 2: Modify or Add the Following Lines

Make sure these lines are **uncommented** and correctly set(default settings are commented to show on screenshot):

```
PermitRootLogin no
PasswordAuthentication no
Port 2222
LoginGraceTime 30
MaxAuthTries 3
```

```
# Authentication:

#LoginGraceTime 2m
LoginGraceTime 30s
#PermitRootLogin prohibit-password
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
MaxAuthTries 3
#MaxSessions 10
MaxSessions 3
```

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
PasswordAuthentication no
#PermitEmptyPasswords no
```

Step 3: Restart the SSH Service

sudo systemctl restart ssh

💡 Explanation of Changes

- **PermitRootLogin no**
  ➤ Disables direct root login, reducing privilege escalation risks.
- **PasswordAuthentication no**
  ➤ Enforces key-based authentication, making brute-force attempts ineffective.
- **Port 2222**
  ➤ Changes the SSH port from the default 22 to a non-standard port for basic stealth (security through obscurity).
- **LoginGraceTime 30**
  ➤ Limits the time allowed to successfully log in before the connection is dropped.
- **MaxAuthTries 3**
  ➤ Restricts the number of authentication attempts, mitigating brute-force login attempts.
- **AllowUsers your_username**
  ➤ Limits SSH access to specific user(s), reducing the attack surface.

2. FTP (vsftpd) Hardening

Application Layer (Layer 7)

Problem Description:

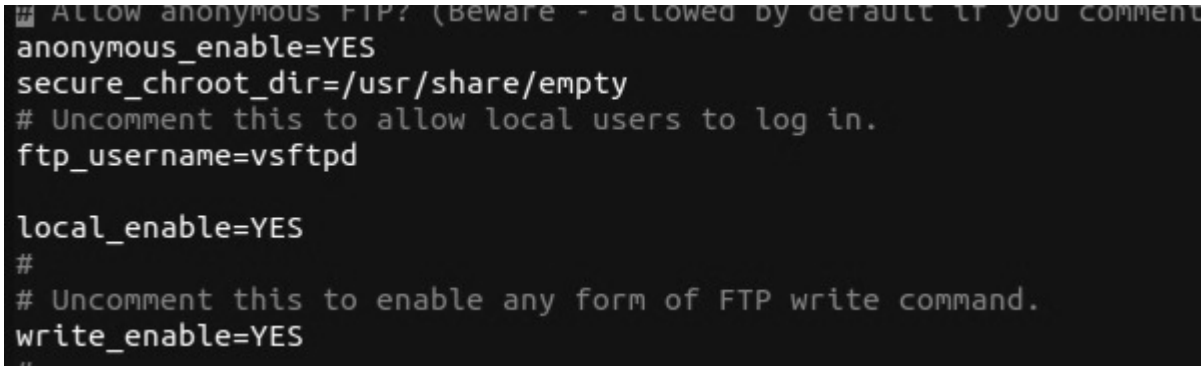
When FTP services run with default configurations, the following vulnerabilities may arise:

- The vsftpd (Very Secure FTP Daemon) may allow anonymous login by default.

- Once logged in, local users may have unrestricted access to the file system beyond their home directories.

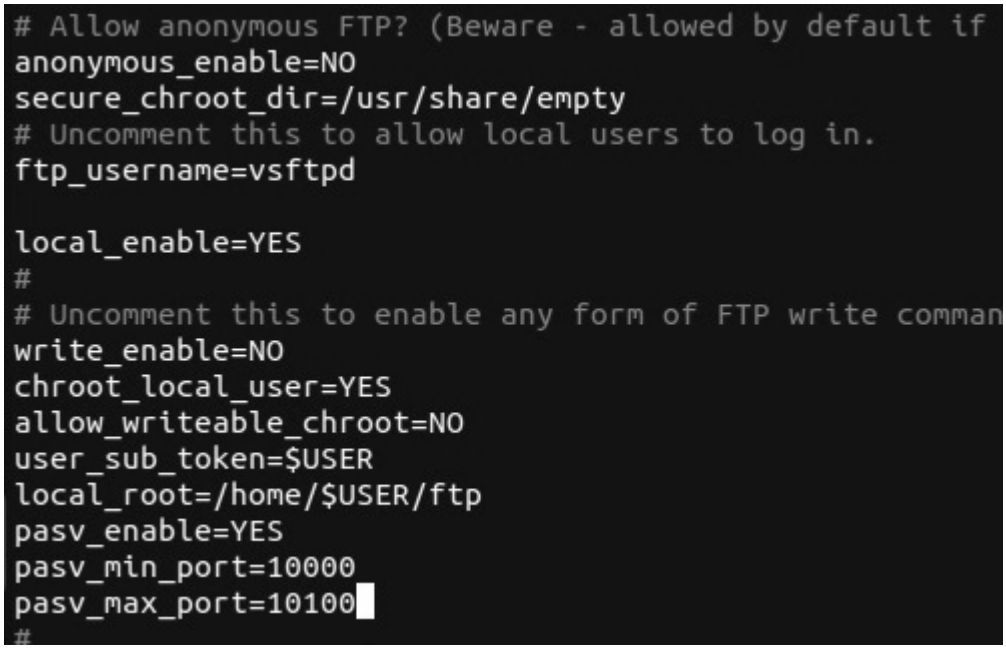
Resolution Steps:

Step 1: Open the vsftpd configuration file
sudo nano /etc/vsftpd.conf



Step 2: Update or add the following settings:
anonymous_enable=NO
local_enable=YES
write_enable=NO
chroot_local_user=YES
allow_writeable_chroot=NO
user_sub_token=$USER
local_root=/home/$USER/ftp
pasv_enable=YES
pasv_min_port=10000
pasv_max_port=10100

Step 3: Restart the vsftpd service
sudo systemctl restart vsftpd



Explanation:

- **Anonymous access was disabled.**
- **Local users were locked into their home directories.**
- **Passive mode was enabled with a defined port range.**
- **FTP access was secured system-wide.**

3. General Server Hardening

Application Layer (Layer 7)

Problem Description:

Servers with default configurations are highly vulnerable to common attack vectors such as brute-force login attempts, unauthorized remote access, and unfiltered network traffic. Without mechanisms like firewall rules or intrusion prevention tools, attackers can exploit open ports, insecure SSH/FTP configurations, or excessive login attempts. These weaknesses expose the server to both automated and targeted threats at the Application Layer (Layer 7) and Network Layer (Layer 3/4).

Step 1: Install and enable Fail2Ban

```
sudo apt install fail2ban -y
```

Step 2: Configure Fail2Ban to protect SSH

```
sudo nano /etc/fail2ban/jail.local
```

```
[sshd]
enabled = true
```

port = 2222
filter = sshd
logpath = /var/log/auth.log
maxretry = 3

---

sudo systemctl restart fail2ban

```
ssh: connect to host 192.168.84.137 port 22: Network is unreachable
berra@192 /etc % ssh helloiamattacker@192.168.84.137
ssh: connect to host 192.168.84.137 port 22: Network is unreachable
berra@192 /etc % ssh exploitme@192.168.84.137
ssh: connect to host 192.168.84.137 port 22: Network is unreachable
berra@192 /etc % ssh exploitme@192.168.84.137
ssh: connect to host 192.168.84.137 port 22: Network is unreachable
berra@192 /etc %
```

```
exploitme@exploitme-machine:/etc/fail2ban$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 0
   |- Total banned:     0
   `- Banned IP list:
```

Step 3: Install and configure UFW (Uncomplicated Firewall)
sudo apt install ufw -y
sudo ufw allow 2222/tcp
sudo ufw allow 21/tcp
sudo ufw allow 10000:10100/tcp
sudo ufw enable

```
exploitme@exploitme-machine:~$ sudo ufw allow 2222/tcp
Rules updated
Rules updated (v6)
exploitme@exploitme-machine:~$ sudo ufw allow 21/tcp
Rules updated
Rules updated (v6)
exploitme@exploitme-machine:~$ sudo ufw allow 10000:10100/tcp
Rules updated
Rules updated (v6)
exploitme@exploitme-machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Resolution Summary:
SSH and FTP services were hardened. Root login and password-based SSH access were disabled. Anonymous FTP login was blocked, and local users were jailed to their own

directories. Fail2Ban and UFW were enabled to provide essential network-level protection. As a result, the server was significantly hardened against common application-layer attacks.

4. Inter-VLAN Communication Partially Blocked – One-Sided Access

Network Layer (Layer 3)

Problem Description:

All devices (Laptop0, Laptop1, WebServer) were initially part of **VLAN 1**, the default VLAN. This meant all devices could ping each other. However, due to updated security policies from the IT department, **Laptop0 (Client1)** and **Laptop1 (Client2)** were not supposed to communicate directly. The only allowed shared resource was **WebServer**.

The goal was to segment the network using VLANs and allow selective inter-VLAN communication via a router.

Desired Communication Matrix:

Source	Destination	Result
Laptop0	WebServer	Allowed (ping successful)
Laptop1	WebServer	Allowed (ping successful)
Laptop0	Laptop1	Blocked (ping fails)

■ Possible Causes:

- All devices in VLAN 1 → flat network, no isolation
- No VLAN segmentation on the switch
- Subinterfaces not correctly configured (Router-on-a-Stick issue)
- Trunk port not configured between Switch and Router
- IP assigned to physical interface (G0/0) instead of subinterfaces

---

🔧 Troubleshooting & Configuration Steps:

Step 1: Create VLANs on the Switch

```
Switch(config)# vlan 10
Switch(config-vlan)# name Client_VLAN_10
Switch(config)# vlan 20
Switch(config-vlan)# name Client_VLAN_20
```

Step 2: Assign Switch Ports to VLANs

Device	Switch Port	VLAN
Laptop0	Fa0/1	10
Laptop1	Fa0/2	20
WebServer	Fa0/3	10

```
Switch(config)# interface fa0/1
Switch(config-if)# switchport mode access
```

Switch(config-if)# switchport access vlan 10

Switch(config)# interface fa0/2

Switch(config-if)# switchport mode access

Switch(config-if)# switchport access vlan 20

Switch(config)# interface fa0/3

Switch(config-if)# switchport mode access

Switch(config-if)# switchport access vlan 10

```
10   Client_VLAN_10                     active    Fa0/1, Fa0/3
20   Client_VLAN_20                     active    Fa0/2
```

Step 3: Configure Trunk Between Switch and Router

Switch(config)# interface fa0/24

Switch(config-if)# switchport mode trunk

```
Switch#show interfaces fa0/24 switchport
Name: Fa0/24
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
 --More--
```

Step 4: Configure Router Subinterfaces (Router-on-a-Stick)

Critical Fix: Remove IP from physical interface (G0/0), assign to subinterfaces only!

Router(config)# interface g0/0

Router(config-if)# no ip address

Router(config)# interface g0/0.10

Router(config-subif)# encapsulation dot1Q 10

Router(config-subif)# ip address 192.168.10.1 255.255.255.0

Router(config)# interface g0/0.20

Router(config-subif)# encapsulation dot1Q 20

Router(config-subif)# ip address 192.168.20.1 255.255.255.0

Router(config)# interface g0/0

Router(config-if)# no shutdown

```
Router(config)#interface g0/0
Router(config-if)#no ip address
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface g0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface g0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES manual up                    up
GigabitEthernet0/0.10  192.168.10.1    YES manual up                    up
GigabitEthernet0/0.20  192.168.20.1    YES manual up                    up
GigabitEthernet0/1     unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
Router#
```

Step 5: Assign IP Settings on End Devices

Device	IP Address	Subnet Mask	Default Gateway
Laptop0	192.168.10.10	255.255.255.0	192.168.10.10
Laptop1	192.168.20.10	255.255.255.0	192.168.20.10
WebServer	192.168.10.20	255.255.255.0	192.168.10.20

Step 6: Test Ping Connectivity

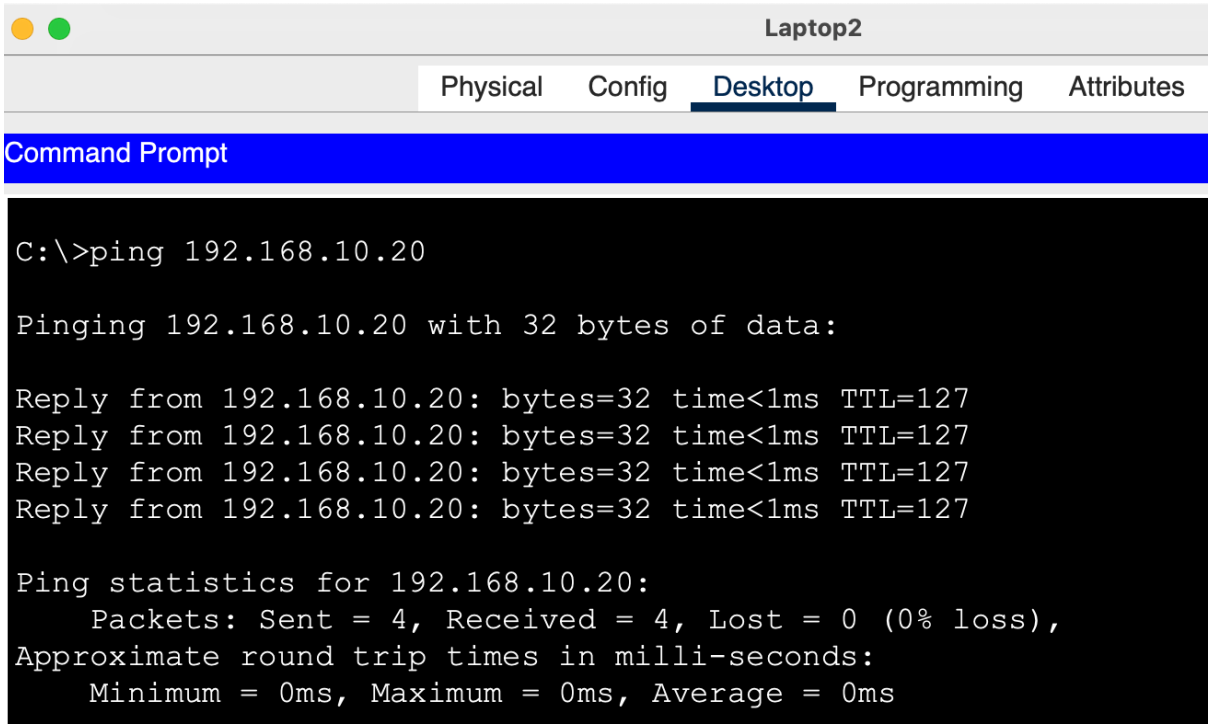
🔷 **Laptop0 → WebServer**

ping 192.168.10.20 → Successful

```
C:\>ping 192.168.10.20

Pinging 192.168.10.20 with 32 bytes of data:

Reply from 192.168.10.20: bytes=32 time=1ms TTL=128
Reply from 192.168.10.20: bytes=32 time<1ms TTL=128
Reply from 192.168.10.20: bytes=32 time<1ms TTL=128
Reply from 192.168.10.20: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

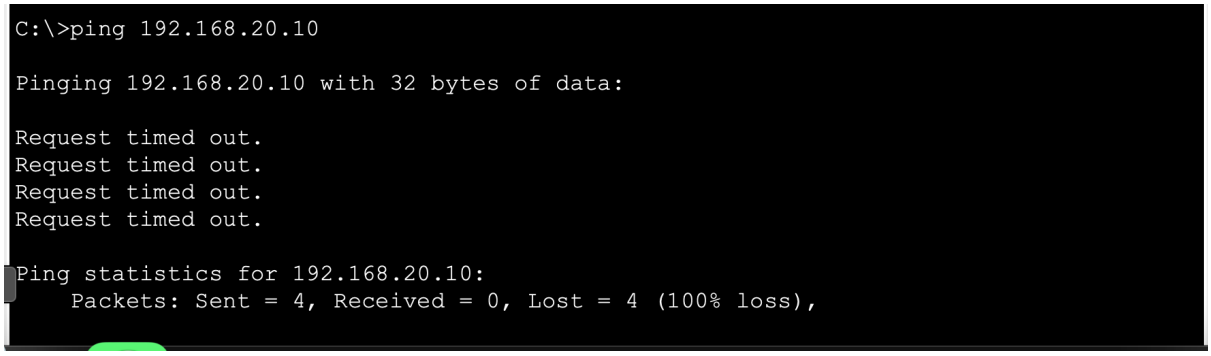
🔷 **Laptop1 → WebServer**

ping 192.168.10.20 → Successful



🔷 **Laptop0 → Laptop1**

ping 192.168.10.10 → Request timed out



Resolution Summary:

The network segmentation and inter-VLAN routing were not working correctly due to an **IP address assigned on the router's physical interface (G0/0)**. This prevented ARP from resolving correctly across VLANs. Once the IP was removed from the physical interface and assigned to the subinterfaces instead, full routing and isolation were restored.

No ACL was needed — the **VLAN-based segmentation combined with Router-on-a-Stick** provided natural communication control:

- Both clients can access WebServer
- Clients cannot communicate with each other

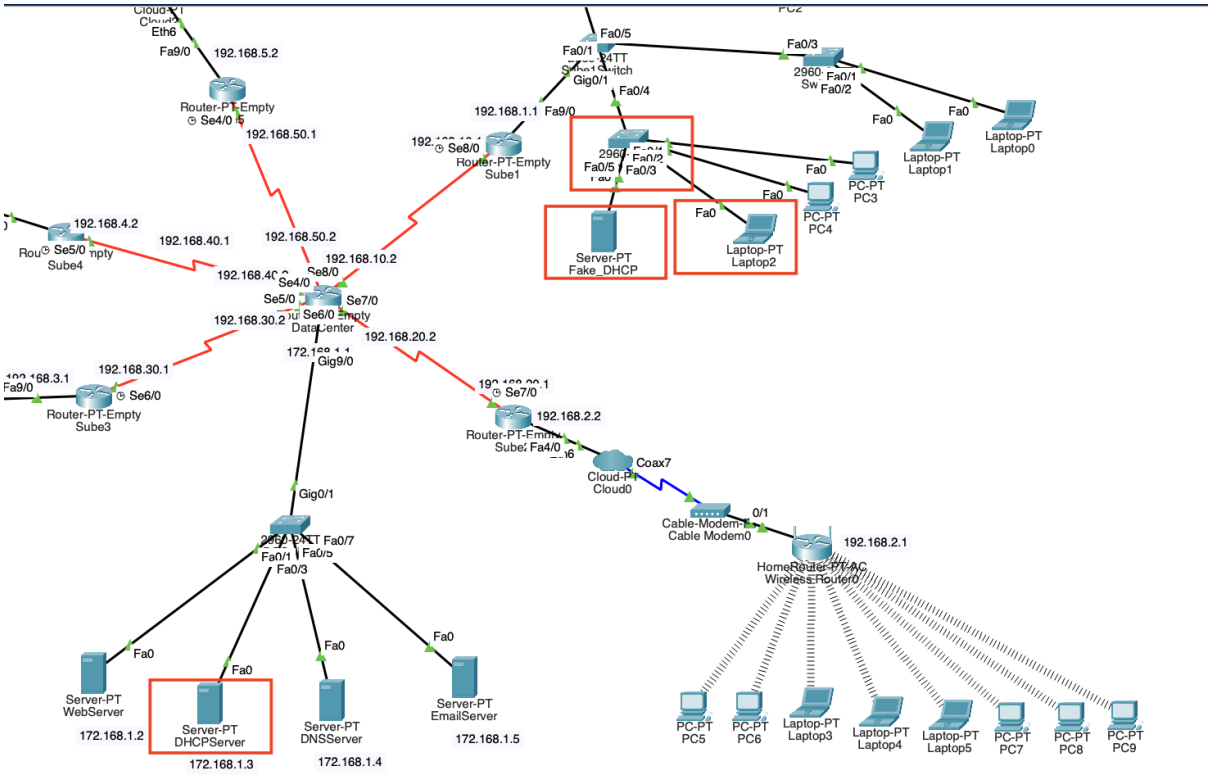
5. Unauthorized DHCP Server (DHCP Spoofing)

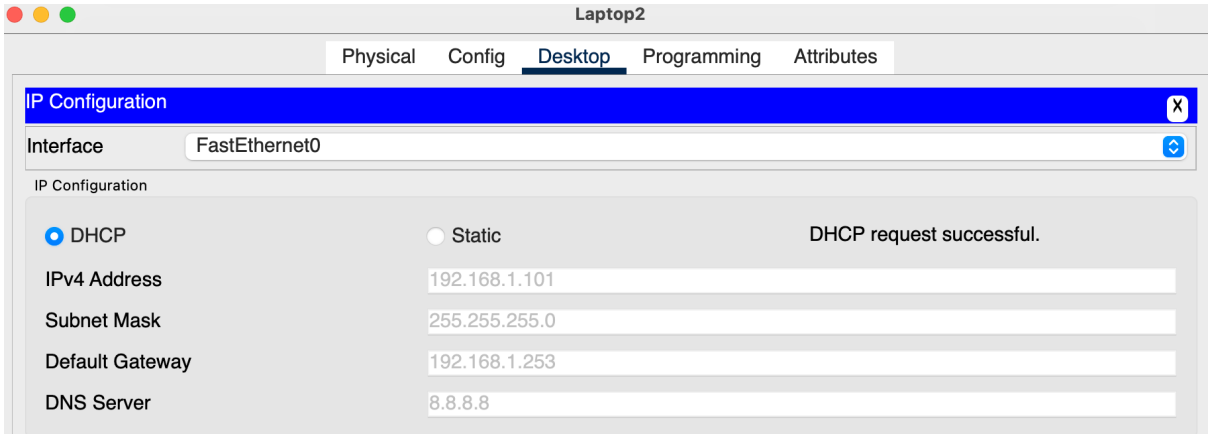
Network Layer (Layer 3)

■ Problem Description:

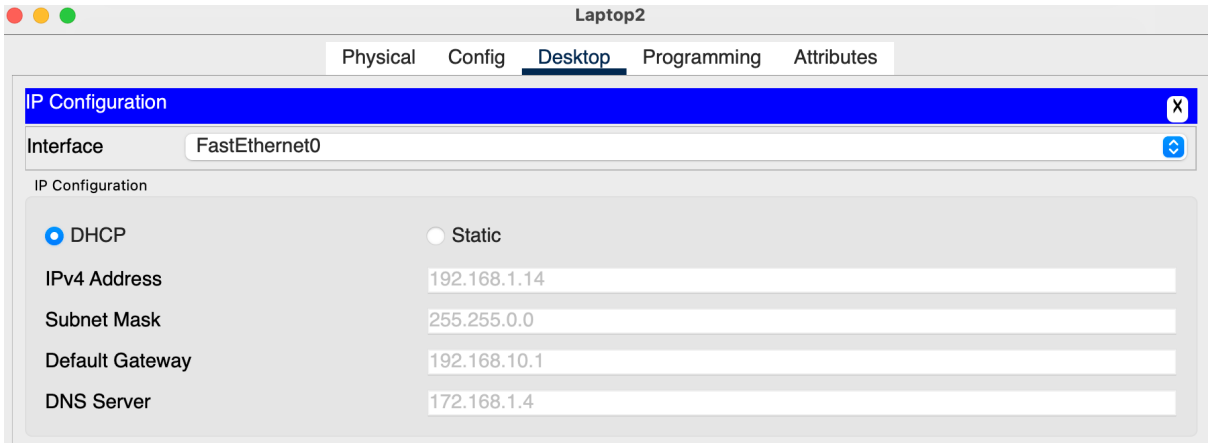
A rogue (unauthorized) DHCP server has been introduced into the network. As a result, end-user devices began receiving incorrect IP configurations—such as wrong default gateway or DNS server settings. This can redirect traffic, cause network disruptions, or pose serious security threats.

Note: Without the rogue DHCP server, clients receive proper IP addresses (e.g., 192.168.1.14) and correct DNS server information (e.g., 172.168.1.4) from the legitimate DHCP server.





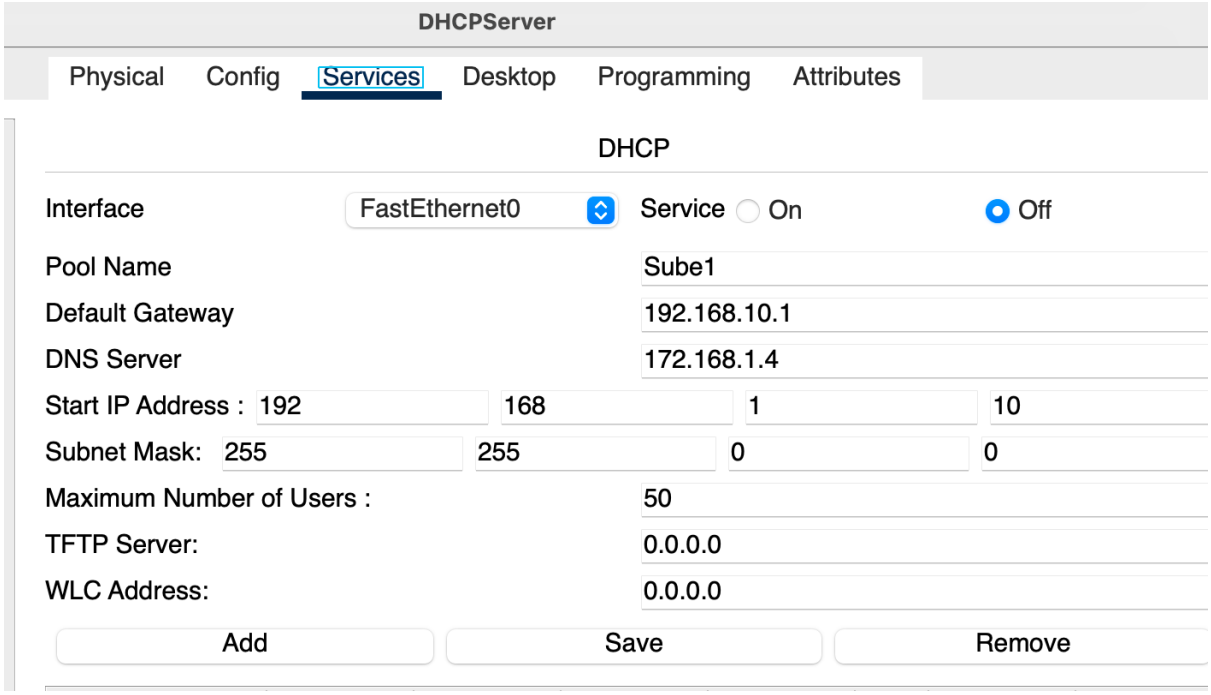
But if there is no fake DHCP, the IP address would be like 192.168.1.14 and DNS Server is 172.168.1.4 like that on original DHCP and its configuration.



Resolution Steps:

Step 1: Temporarily Disable the Legitimate DHCP Server

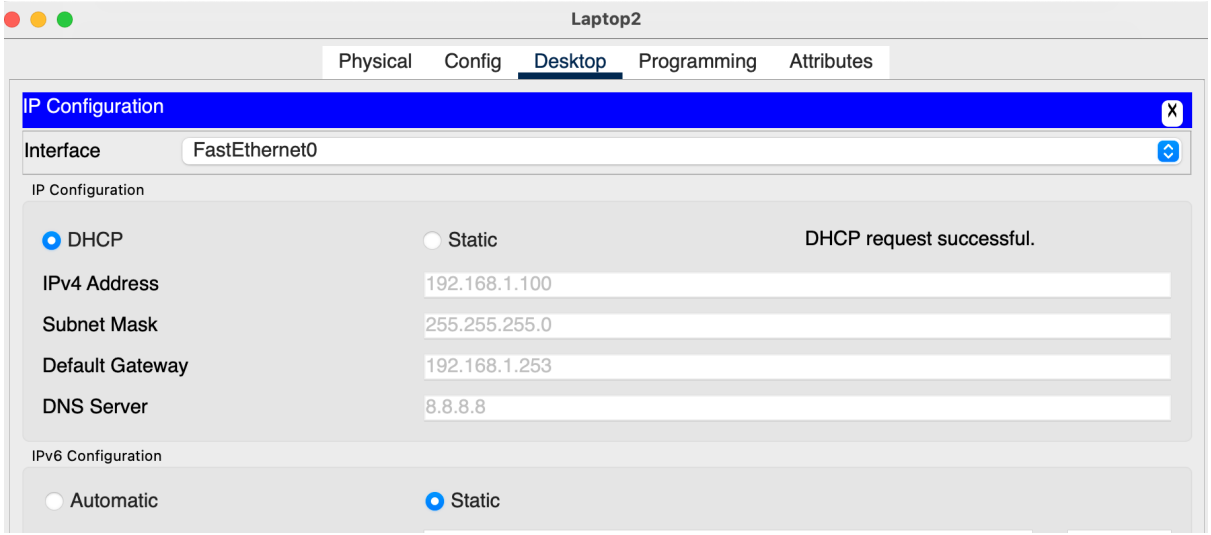
Turn off the DHCP service on the real DHCP server located in the Data Center (Config > DHCP Service: OFF).



Step 2: Renew DHCP Lease on Client Device

Go to Laptop2 → Desktop → IP Configuration → Click on DHCP to renew the IP address.

→ Check if the client obtains an IP address from the fake DHCP server instead of getting an APIPA address (169.254.x.x).



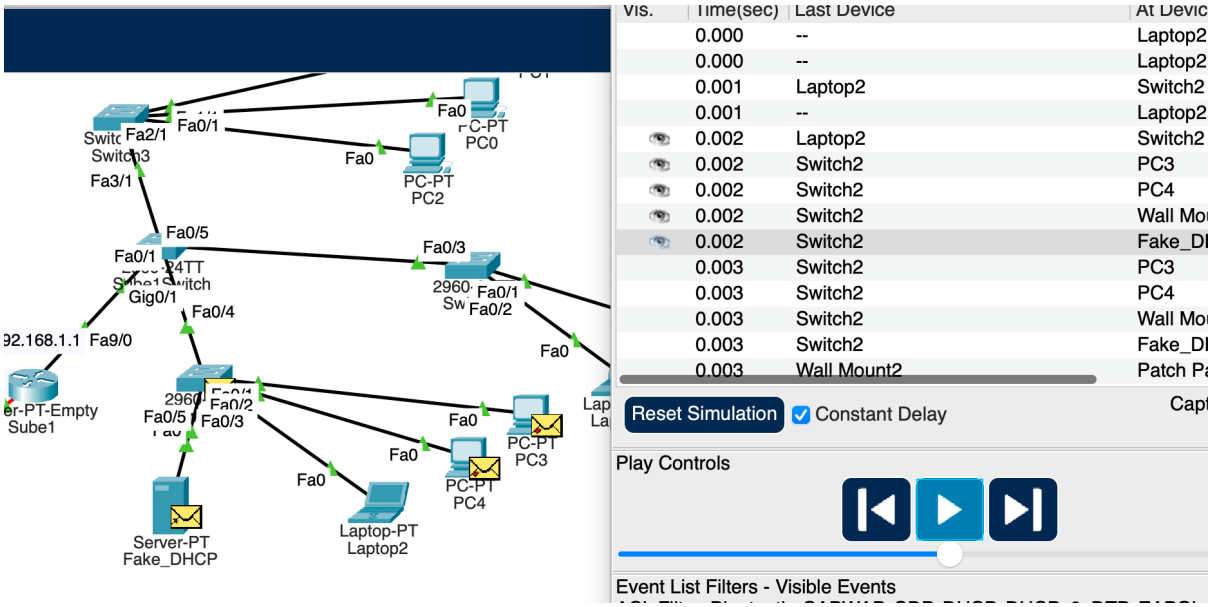
Step 3: Use Simulation Mode to Inspect DHCP OFFER Packet

Enable Packet Tracer's Simulation Mode.

From Laptop2, send a new DHCP Discover request.

→ Observe which device is sending the DHCP OFFER packet.

(This confirms the presence of the rogue DHCP server.)



Solution: Enable DHCP Snooping on the Switch

DHCP Snooping is a security feature that controls which switch ports are allowed to forward DHCP packets. It prevents rogue DHCP servers by filtering DHCP traffic on untrusted ports.

Step 4: Activate DHCP Snooping

```
Switch(config)#ip dhcp snooping
Switch(config)#ip dhcp snooping vlan 1
Switch(config)#interface fastethernet0/1
Switch(config-if)#ip dhcp snooping trust
Switch(config-if)#
```

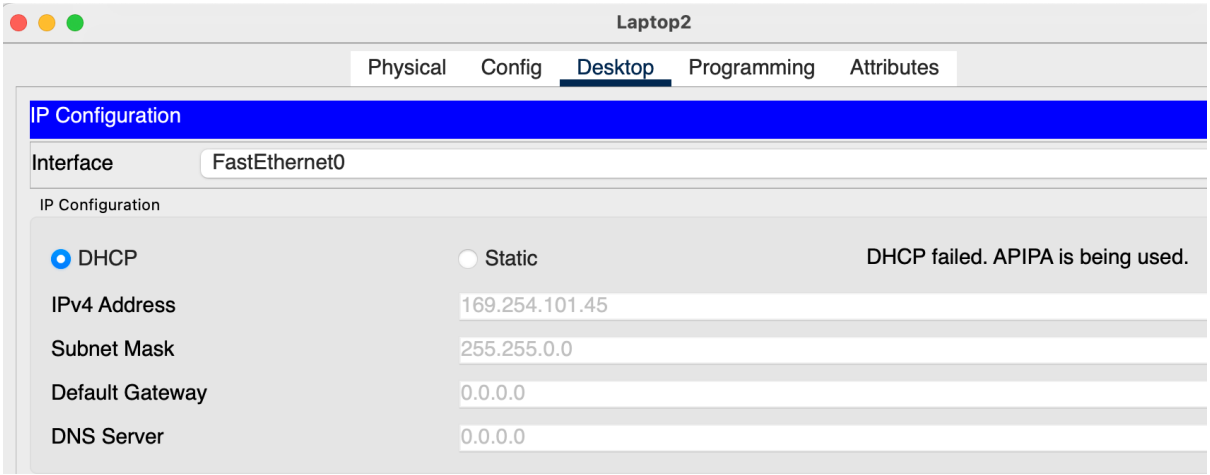
- Enable DHCP Snooping on VLAN 1.
- Mark the switch port connected to the **legitimate DHCP server** (e.g., FastEthernet0/1) as **trusted**.
- All other client ports are considered **untrusted**, blocking DHCP offers from rogue servers.

This setup ensures DHCP OFFER messages only come from trusted ports.

Step 5: Retest DHCP Assignment

- Turn the legitimate DHCP server back on.
- Even if the fake DHCP server remains active, clients will no longer accept its IP offers.
- On Laptop2, renew DHCP lease again.
  → Confirm that the IP address comes from the legitimate DHCP server.

If clients fail to get an IP from the rogue server, they may temporarily fall back to APIPA until the real DHCP responds.

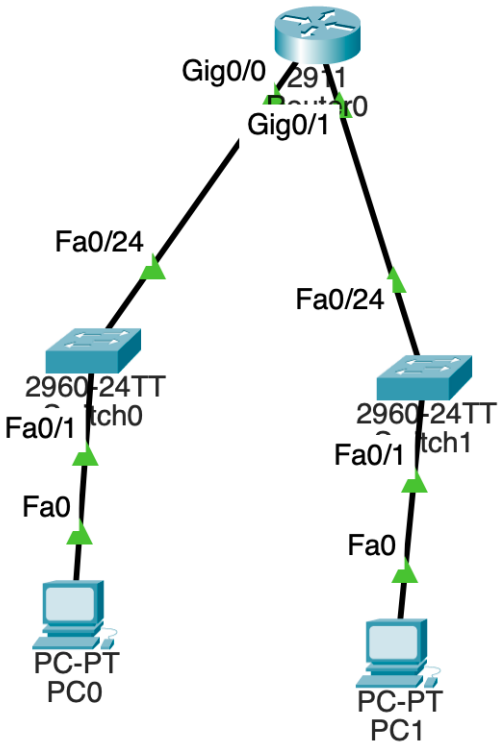


■ Resolution Summary:
An attacker introduced a rogue DHCP server trying to manipulate the network by handing out incorrect IP configurations. By enabling DHCP Snooping on the switch and designating only the real DHCP server's port as trusted, the rogue DHCP server's packets are blocked. This secures the network and ensures clients only receive valid IP information from the authorized DHCP server.

6. Classful vs CIDR Subnetting – Legacy Device Conflict

Network Layer (Layer 3)

■ Problem Description:
A legacy router was configured to use classful subnet assumptions, specifically assigning a 255.255.255.0 (Class C) subnet mask to an interface with a 192.168.x.x address, despite the network being designed with CIDR subnetting (/28). As a result, traffic from PC1 (192.168.40.18/28) to PC2 (192.168.40.173/28) could pass through the router, but the reverse direction failed due to mismatched subnet recognition.

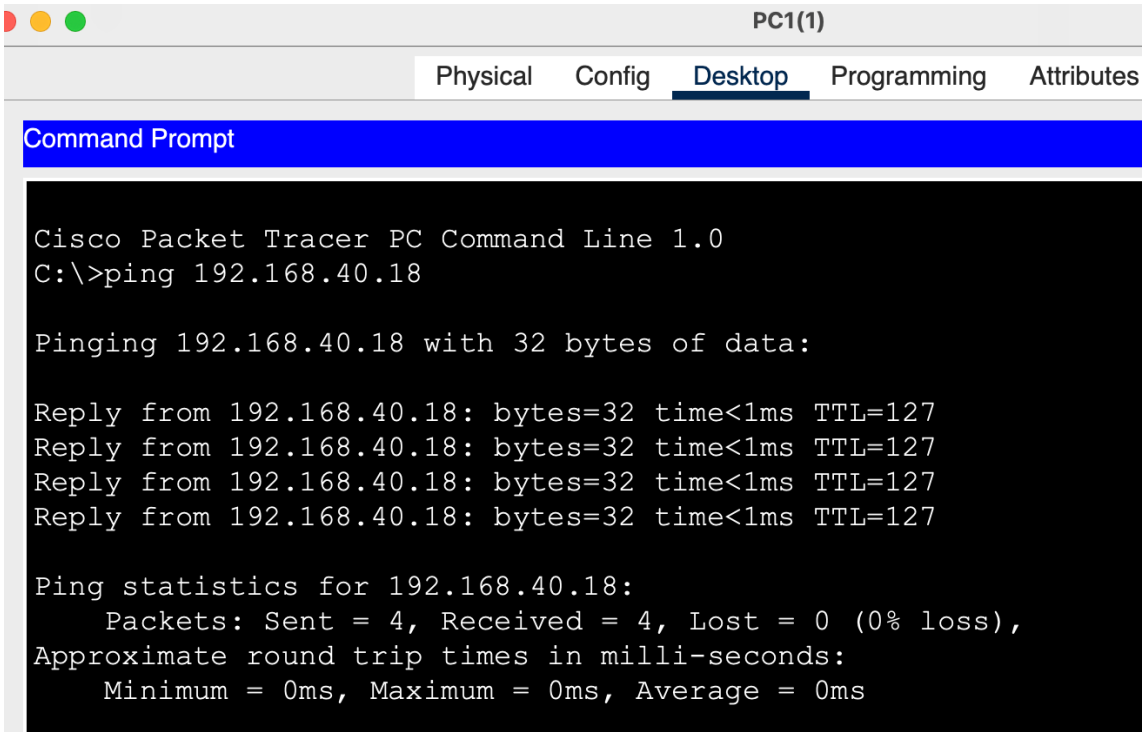


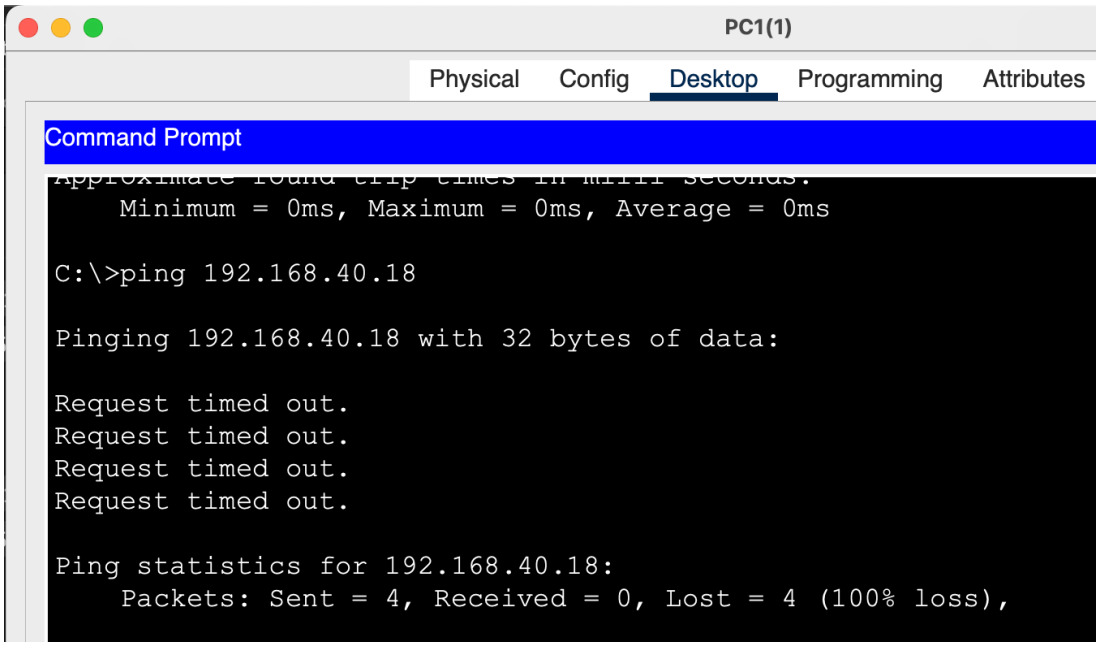
■ Troubleshooting Steps:

Step 1: Assign addresses and connect PCs via a router with two interfaces

- PC1: 192.168.40.18 /28, GW: 192.168.40.17
- PC2: 192.168.40.173 /28, GW: 192.168.40.161
- Router interfaces:
    - Gig0/0 → 192.168.40.17 255.255.255.240
    - Gig0/1 → 192.168.40.161 255.255.255.0 ← incorrect

Step 2: Test reachability from PC1 → PC2
→ ping fails due to subnet conflict
Step 3: Test reachability from PC2 → PC1
→ ping fails due to subnet conflict






```
Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.40.161  YES manual up                    up
GigabitEthernet0/1     unassigned      YES manual up                    up
GigabitEthernet0/2     unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
Router#
```

Step 4: Fix the router interface mask

Router(config)# interface g0/1

Router(config-if)# ip address 192.168.40.161 255.255.255.240

Step 5: Test bidirectional communication
→ ping successful from both sides

■ Resolution Summary:
The issue was caused by a legacy device defaulting to classful subnet assumptions (Class C), ignoring the CIDR /28 configuration. After correcting the subnet mask on the router interface, full bidirectional communication was restored between subnets.

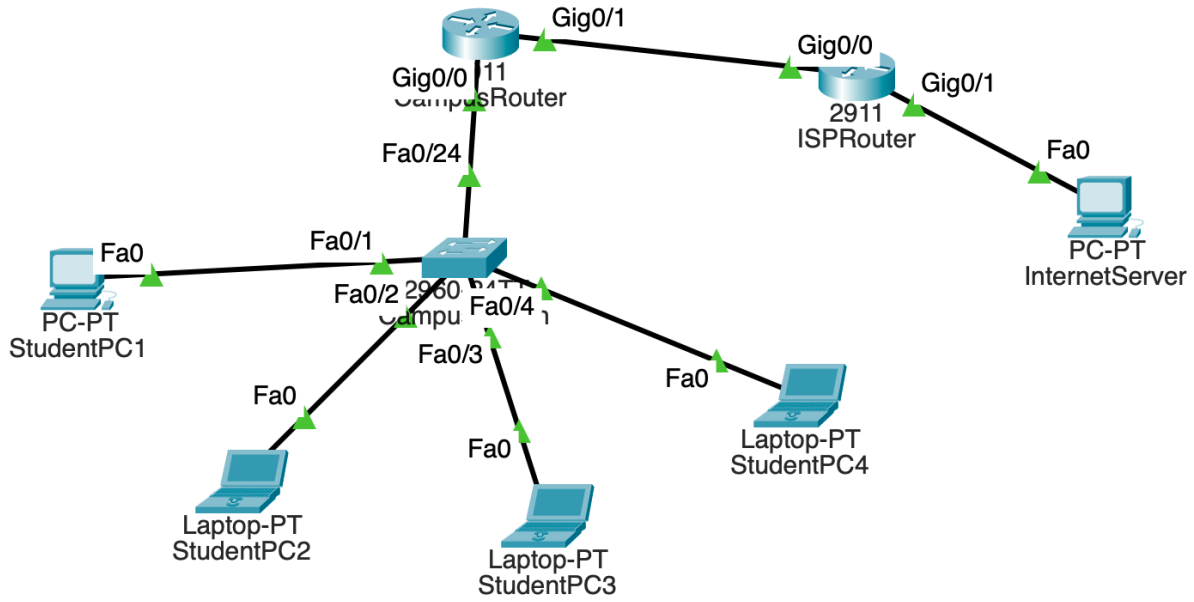
7. Missing NAT Configuration – Internet Access Failure in Campus Network

Network Layer (Layer 3)

■ Problem Description:

Although all devices in the campus network have correctly assigned IP addresses and default gateways, they are unable to access external servers (e.g., 8.8.8.8). While internal communication works properly, ping attempts to public IPs fail.

Upon inspection, it was discovered that the **Campus_Router** lacked proper NAT (Network Address Translation) configuration. Since private IP addresses (e.g., 192.168.10.11) were being sent to the public network without translation, they were not recognized by the upstream router (ISP) and were consequently dropped.



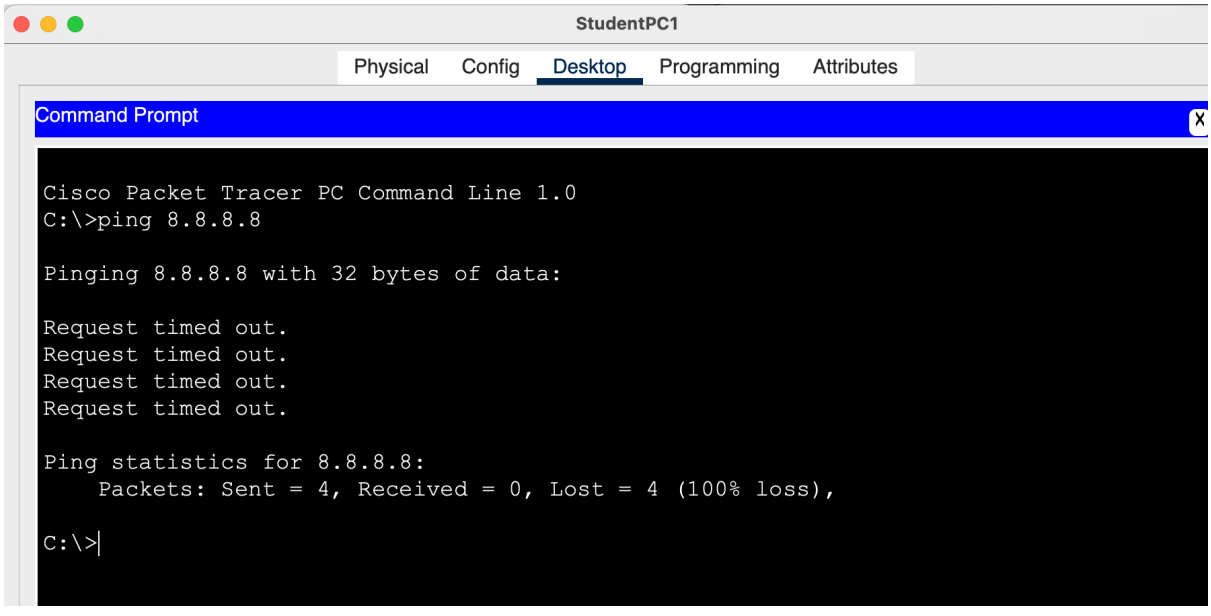
■ Troubleshooting Steps:

✅ Step 1: Initial Ping Test

From Student1_PC, the following command was executed:

ping 8.8.8.8

→ The result was unsuccessful: Request timed out.



✅ **Step 2: NAT Translation Table Checked**

On **Campus_Router**, the following command was run:

show ip nat translations

→ The NAT translation table was **empty**.

```
Router#show ip nat translations
Router#
```

✅ **Step 3: NAT Configuration Verified**

show run | section nat

→ No NAT-related configuration was found.

✅ **Step 4: NAT Configuration Applied**

On **Campus_Router**, the following configuration commands were entered:

configure terminal
access-list 1 permit 192.168.10.0 0.0.0.255
ip nat inside source list 1 interface gigabitEthernet 0/1 overload

interface gig0/0
ip nat inside
exit

interface gig0/1
ip nat outside
exit

end

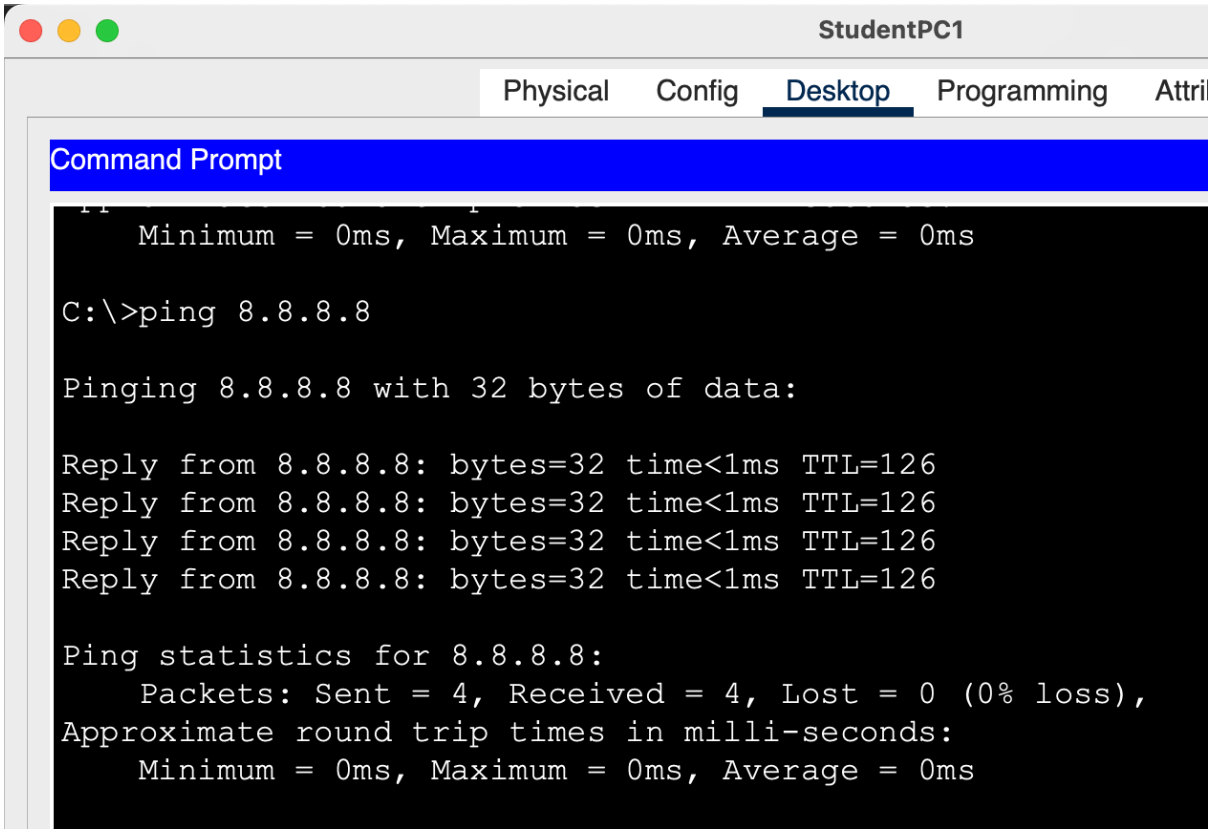
```
Router#show ip nat translations
Router#show run | section nat
 ip nat inside
 ip nat outside
ip nat inside source list 1 interface GigabitEthernet0/1
overload
```

✅ **Step 6: Final Ping Test**

Again on **Student1_PC**:

bash
Kodu kopyala
ping 8.8.8.8

→ The result was successful: replies received from the external server.



■ **Resolution Summary:**

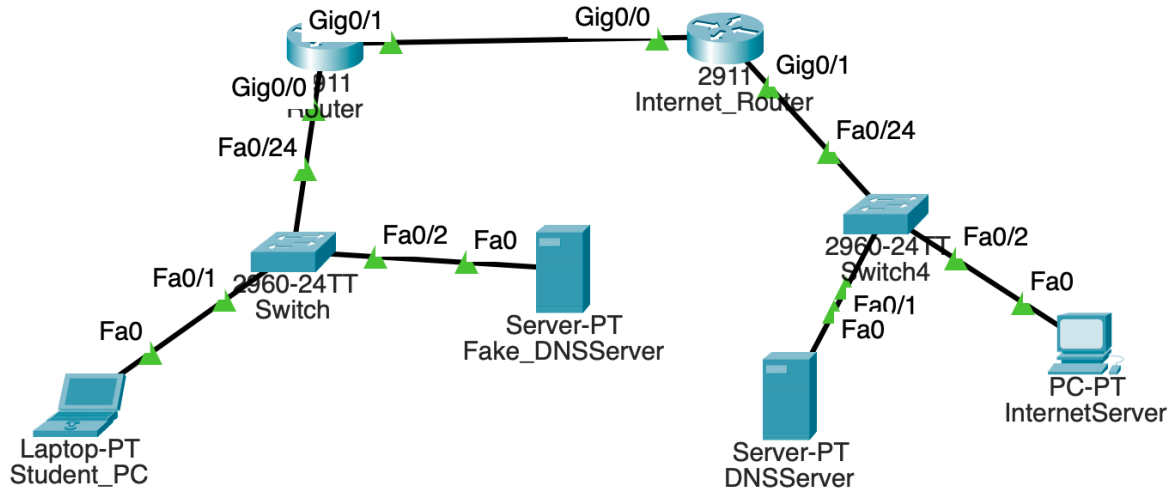
The issue stemmed from a missing NAT configuration on the Campus_Router. Private IP addresses were being sent to the internet without translation and were thus rejected by external routers. After applying NAT overload using an access list and configuring inside/outside interfaces properly, internet access was successfully restored for all internal devices.

8.# DNS Conflict – Internal Device Blocking Public DNS Resolution

Application Layer (Layer 7)

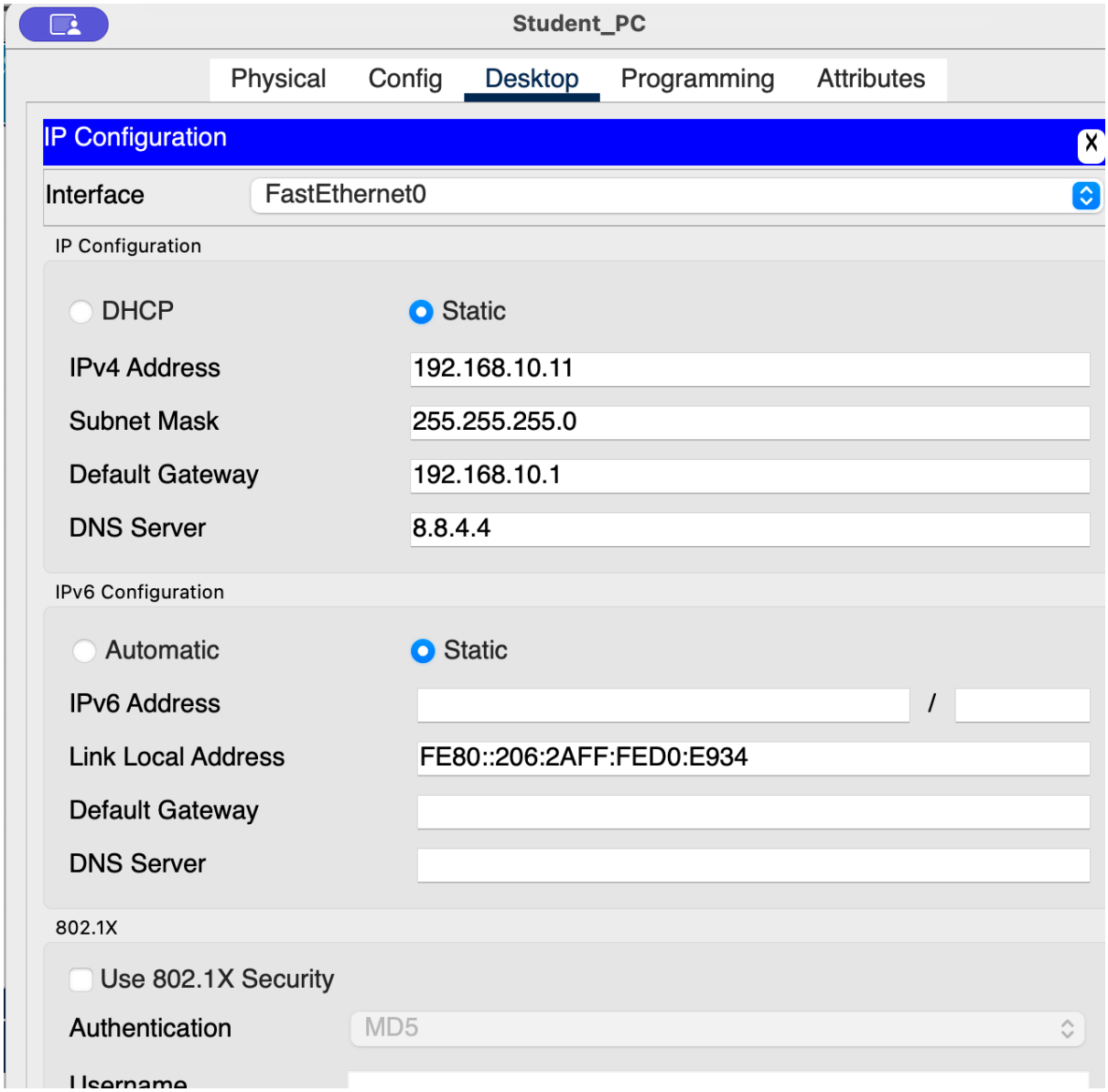
■ Problem Description:
A device within the internal network was misconfigured with the public DNS server address `8.8.8.8`. This caused name resolution to fail across the entire campus network, as client devices unknowingly sent their DNS queries to the rogue internal device instead of the legitimate public DNS.

Even after correcting the DNS IP address on client devices to `8.8.4.4`, resolution failed because the DNS server service on the real DNS device was turned off. Therefore, multiple layers of misconfiguration led to failed name resolution.

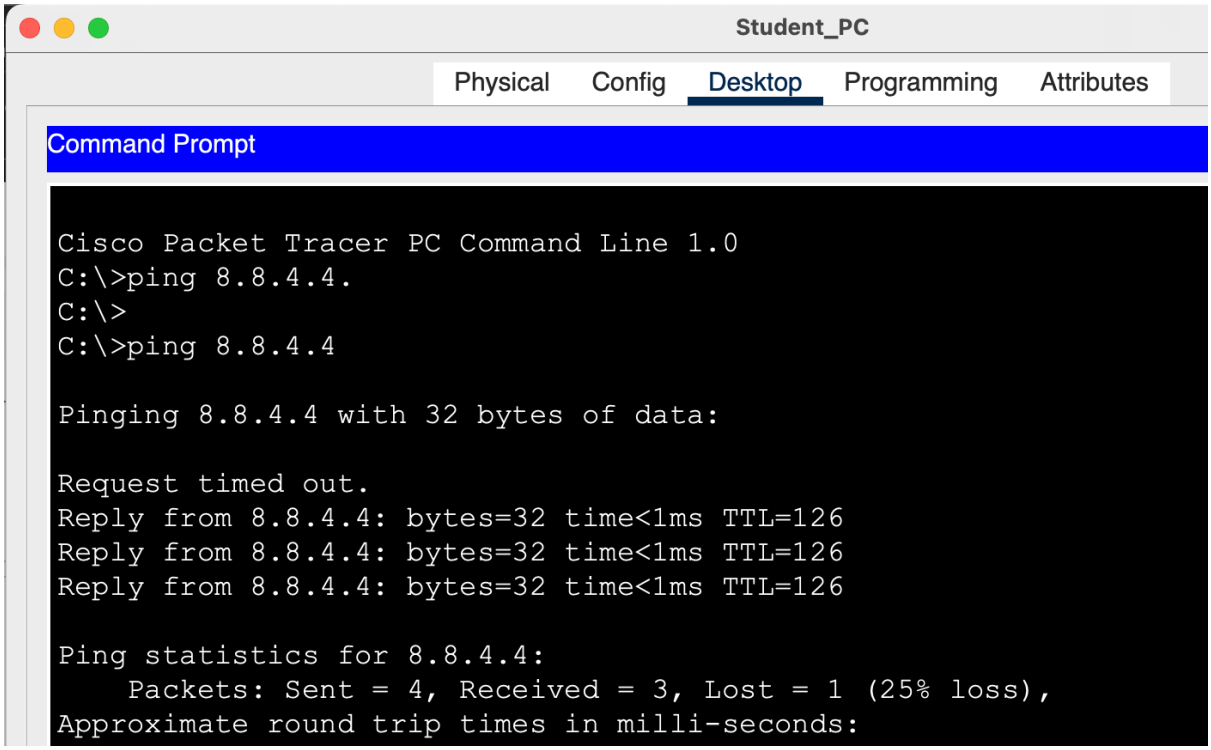


■ Troubleshooting Steps:

✅ **Step 1: Check IP and DNS settings on Student_PC**
IP: `192.168.10.11`, Mask: `255.255.255.0`, Gateway: `192.168.10.1`, DNS: `8.8.4.4`
8.8.8.8 conflicts with internal rogue DNS, so 8.8.4.4 was chosen as a clean alternative.



✅ **Step 2: Ping test to DNS server**
```bash
ping 8.8.4.4
```
→ Successful reply confirms Layer 3 connectivity.



✅ **Step 3: Attempt DNS resolution**

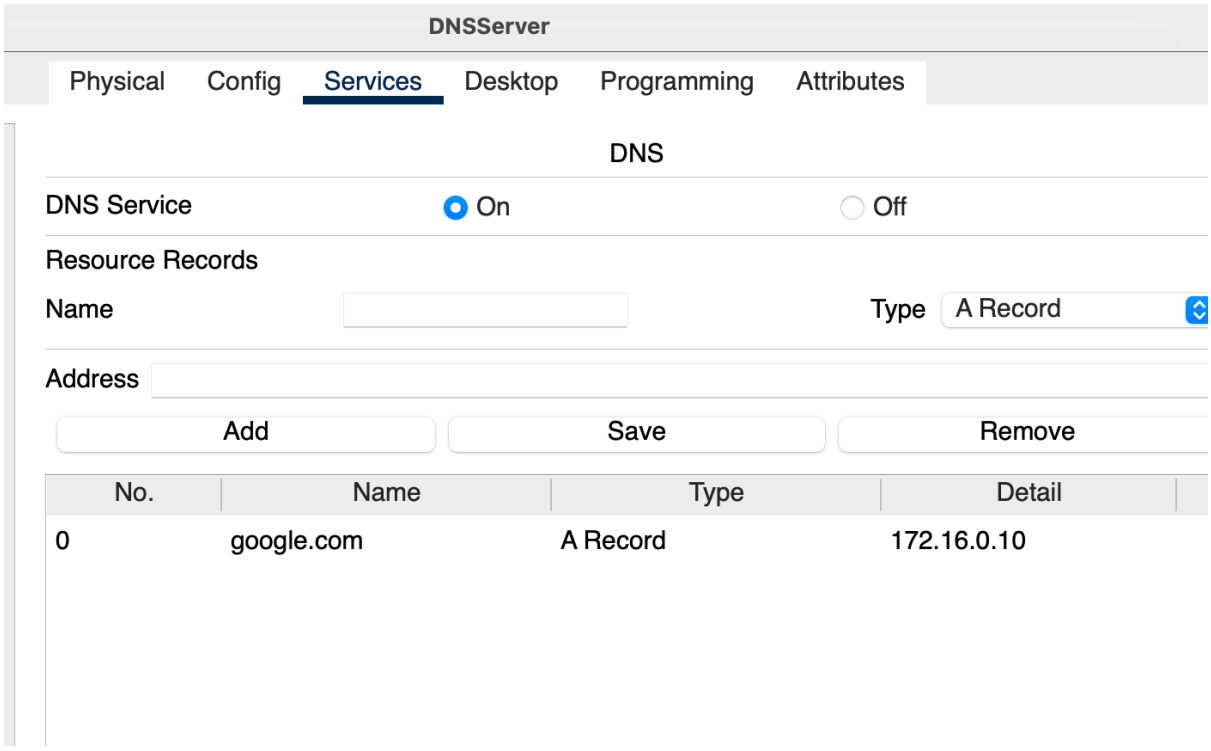
```bash
nslookup google.com
```

→ Initially failed due to DNS service being off

Step 4: Verify and activate DNS service

- On the device configured as DNS Server (8.8.4.4):
- Navigate to `Services > DNS`
- Toggle **Service ON**
- Add DNS Record:

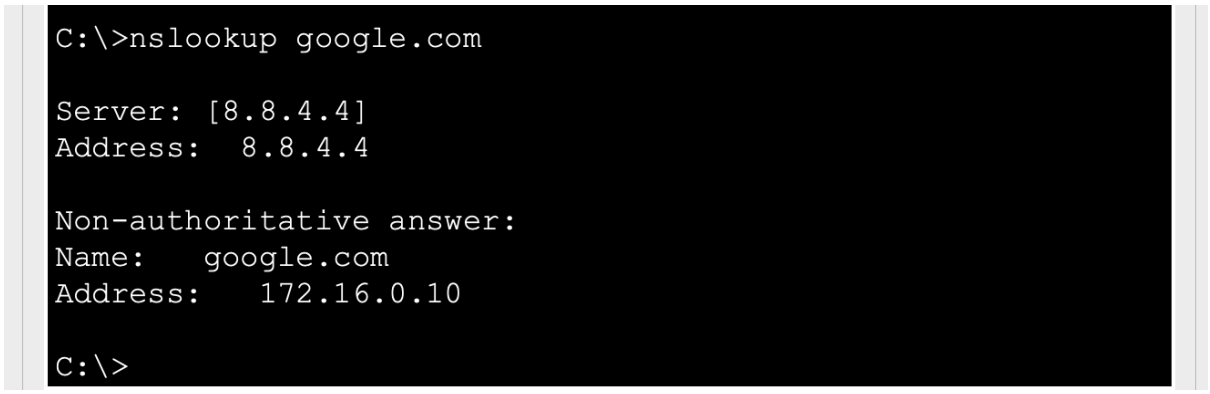
Name	IP Address
`google.com`	`172.16.0.10`



✅ **Step 5: Retest DNS resolution**

```
nslookup google.com
```

→ Successfully resolved



■ Resolution Summary:

The issue originated from an IP conflict with a public DNS address used internally. Additionally, the DNS server was configured correctly in terms of IP but had its service turned off. Once the correct DNS address was set on the client and the service was enabled on the server, DNS resolution began functioning properly.

9. Access Control List (ACL) Misconfiguration – Unintended External Access

Layer: Network (Layer 3)

🔍 Problem Description:

A critical web server hosted inside the network was unintentionally exposed to external access due to a missing Access Control List (ACL) on the Campus_Router. The server was reachable from the Internet_Router, which violated the security policy that mandated internal-only access.

Troubleshooting Steps:

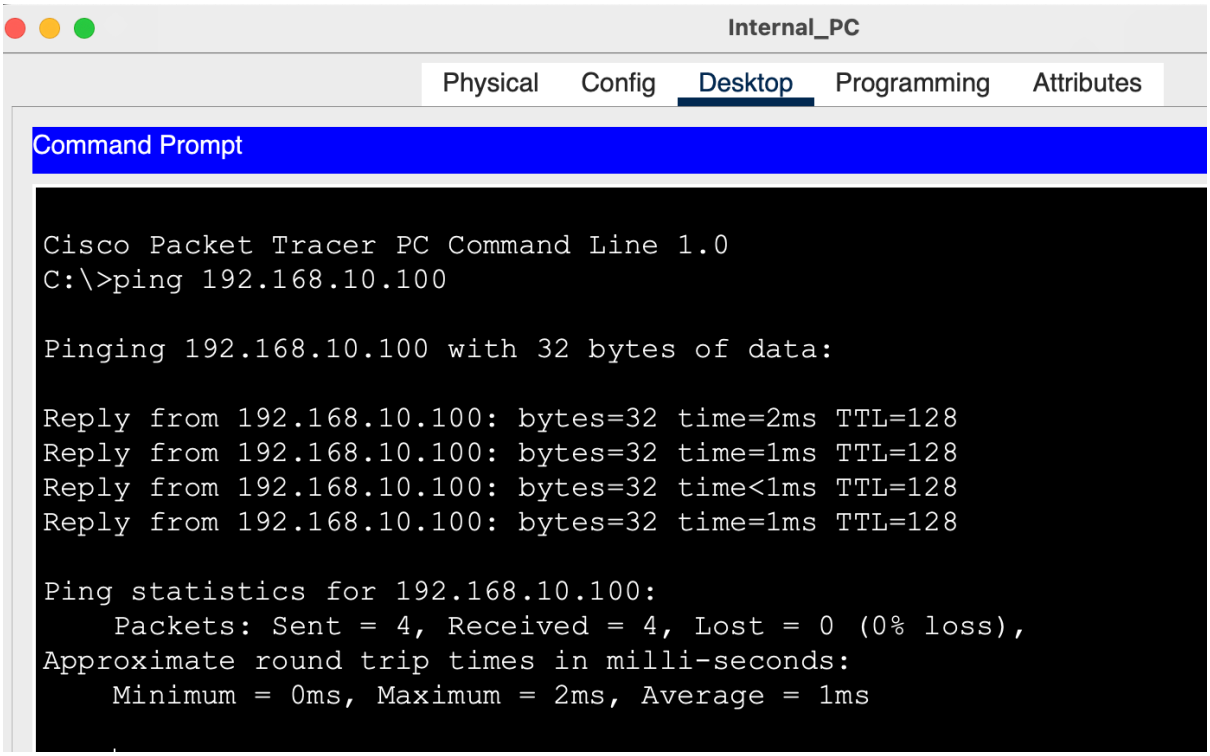
Step 1: Configure network topology

- Internet_Router: 200.0.0.1/30
- Campus_Router g0/0: 200.0.0.2/30 (WAN)
- Campus_Router g0/1: 192.168.10.1/24 (LAN)
- Web_Server: 192.168.10.100
- Internal_PC: 192.168.10.10

Step 2: Test connectivity

- Internal_PC → Web_Server → Success

- Internet_Router → Web_Server → ❌ Unexpected Success




```
Router#ping 192.168.10.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.100, timeout is 2
seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/3/12 ms

Router#ping 192.168.10.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.100, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms

Router#
```

Step 3: Configure ACL on Campus_Router

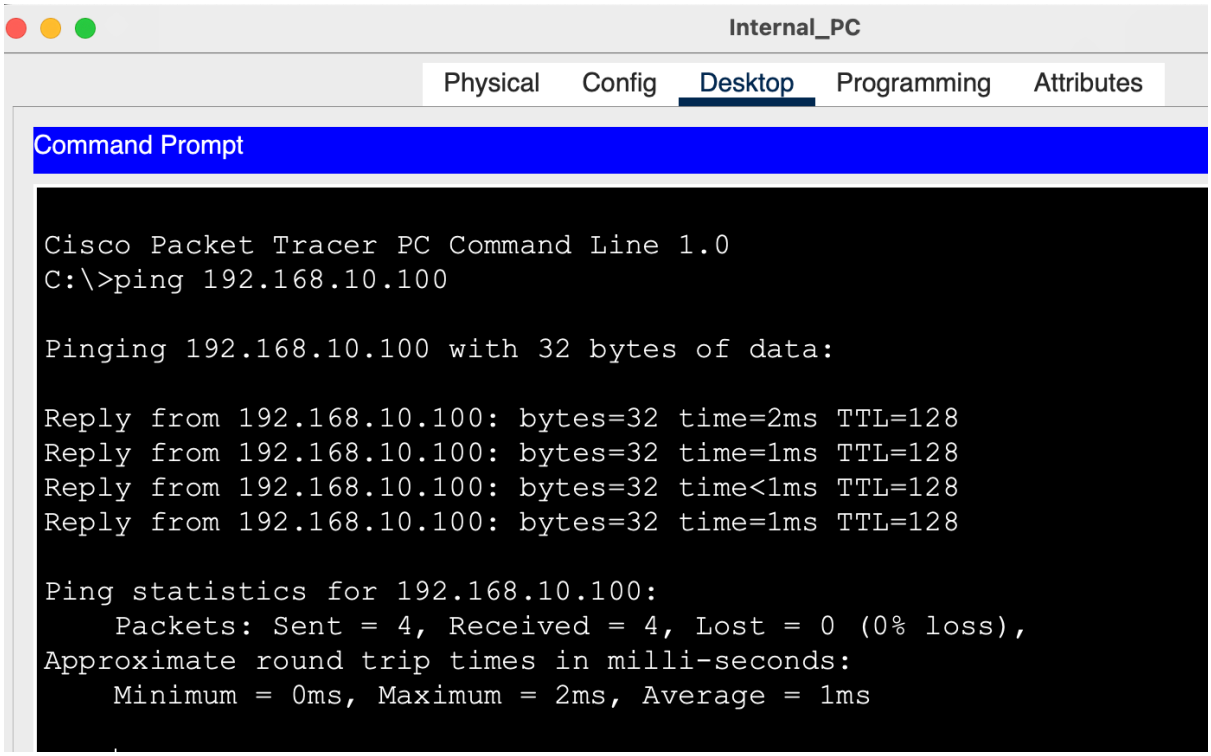
```
access-list 100 permit ip 192.168.10.0 0.0.0.255 any
access-list 100 deny ip any 192.168.10.100 0.0.0.0
interface g0/0
ip access-group 100 in
```

```
Router(config)#access-list 100 permit ip 192.168.10.0 0.0.0.255
any
Router(config)#access-list 100 deny ip any 192.168.10.100
0.0.0.0
Router(config)#interface g0/0
Router(config-if)#ip access-group 100 in
Router(config-if)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

✅ Step 4: Retest connectivity

- Internet_Router → Web_Server → ❌ Now blocked
- Internal_PC → Web_Server → ✅ Still successful

```
Router#ping 192.168.10.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.100, timeout is 2 seconds:
.UUUU
Success rate is 0 percent (0/5)

Router#ping 192.168.10.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.100, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
```



Resolution Summary:

The issue stemmed from the absence of an inbound access control list (ACL) on the external-facing interface of the Campus_Router. After deploying ACL 100 to allow internal traffic while denying external access to the web server, proper segmentation and access restrictions were restored.

---

10. Port Security Violation – Rogue Device Access
OSI Layer: Data Link Layer (Layer 2)

Problem Description:

In this scenario, an unauthorized (rogue) device was able to connect to the switch and gain access to the internal network by physically connecting to an unused port. This is a critical security risk that can lead to data theft, unauthorized network access, and device impersonation.

The objective is to prevent unknown MAC addresses from accessing the network by enforcing port security on the switch, thereby allowing only authorized devices to use a particular port.

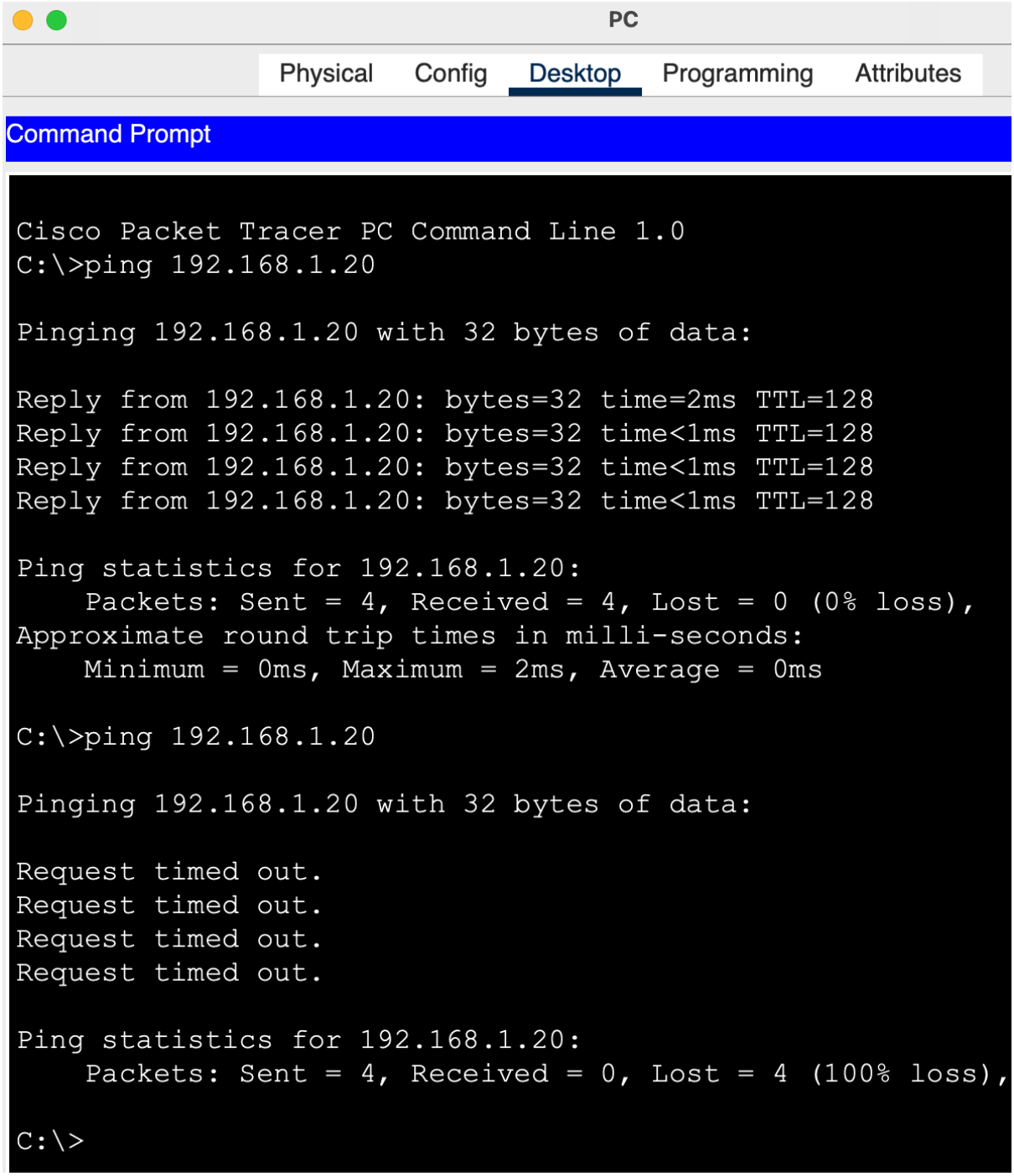
—

Troubleshooting & Attack Simulation Steps:

Step 1: Normal Working State (Before Security Configuration)

- PC was connected to Switch0 via FastEthernet0/1.
- Laptop was connected via FastEthernet0/2.
- PC and Laptop could successfully ping each other.
- No port-security configured yet.

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.20

Pinging 192.168.1.20 with 32 bytes of data:

Reply from 192.168.1.20: bytes=32 time=2ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>
```




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

Step 2: Port Security Applied to Fa0/1 (PC)

Switch0 > CLI:

interface FastEthernet0/1
switchport mode access
switchport port-security
switchport port-security maximum 1
switchport port-security mac-address sticky
switchport port-security violation shutdown

→ This ensures only one MAC (PC) can connect to Fa0/1.

Step 3: Violation Test – Rogue Device Attempt

- PC was unplugged from Fa0/1.
- Laptop was connected to the same port (Fa0/1).
- Laptop tried to ping → failed.
- The port was automatically disabled due to MAC address mismatch (violation detected).

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

```
-----------------------------------------------------------------------
Switch#show interface fa0/1
FastEthernet0/1 is down, line protocol is down (err-disabled)
  Hardware is Lance, address is 0009.7cdb.3501 (bia 0009.7cdb.3501)
 BW 100000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
```

Step 4: Recovery

- Switch CLI used to shut and no shut the disabled port.

Switch0 > CLI:

interface FastEthernet0/1
shutdown
no shutdown

- PC0 was reconnected.
- Port returned to secure-up status.

```
Switch#show port-security interface 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   : 0003.E488.745B:1
Security Violation Count   : 0
```

show port-security interface Fa0/1 → secure-up, 1 MAC learned

—

Resolution Summary:

A rogue device connected to a secured port triggered a violation, and the port was disabled to prevent unauthorized access. By enforcing port-security with sticky MAC and configuring violation mode as shutdown, unauthorized access was effectively prevented. After re-enabling the port and reconnecting the trusted device, normal operation resumed.