

# Lab 3: Detect Network Sniffing

## **Lab Scenario**

The previous labs demonstrated how an attacker carries out sniffing with different techniques and tools. This lab helps you understand possible defensive techniques used to defend a target network against sniffing attacks.

A professional ethical hacker or pen tester should be able to detect network sniffing in the network. A sniffer on a network only captures data and runs in promiscuous mode, so it is not easy to detect. Promiscuous mode allows a network device to intercept and read each network packet that arrives in its entirety. The sniffer leaves no trace, since it does not transmit data. Therefore, to detect sniffing attempts, you must use the various network sniffing detection techniques and tools discussed in this lab.

## **Lab Objectives**

- Detect ARP poisoning and promiscuous mode in a switch-based network

## **Overview of Detecting Network Sniffing**

Network sniffing involves using sniffer tools that enable the real-time monitoring and analysis of data packets flowing over computer networks. These network sniffers can be detected by using various techniques such as:

- **Ping Method:** Identifies if a system on the network is running in promiscuous mode
- **DNS Method:** Identifies sniffers in the network by analyzing the increase in network traffic
- **ARP Method:** Sends a non-broadcast ARP to all nodes in the network; a node on the network running in promiscuous mode will cache the local ARP address

## **Task 1: Detect ARP Poisoning and Promiscuous Mode in a Switch-Based Network**

ARP poisoning involves forging many ARP request and reply packets to overload a switch. ARP cache poisoning is the method of attacking a LAN network by updating the target computer's ARP cache with both forged ARP request and reply packets designed to change the Layer 2 Ethernet MAC address (that of the network card) to one that the attacker can monitor. Attackers use ARP poisoning to sniff on the target network. Attackers can thus steal sensitive information, prevent network and web access, and perform DoS and MITM attacks.

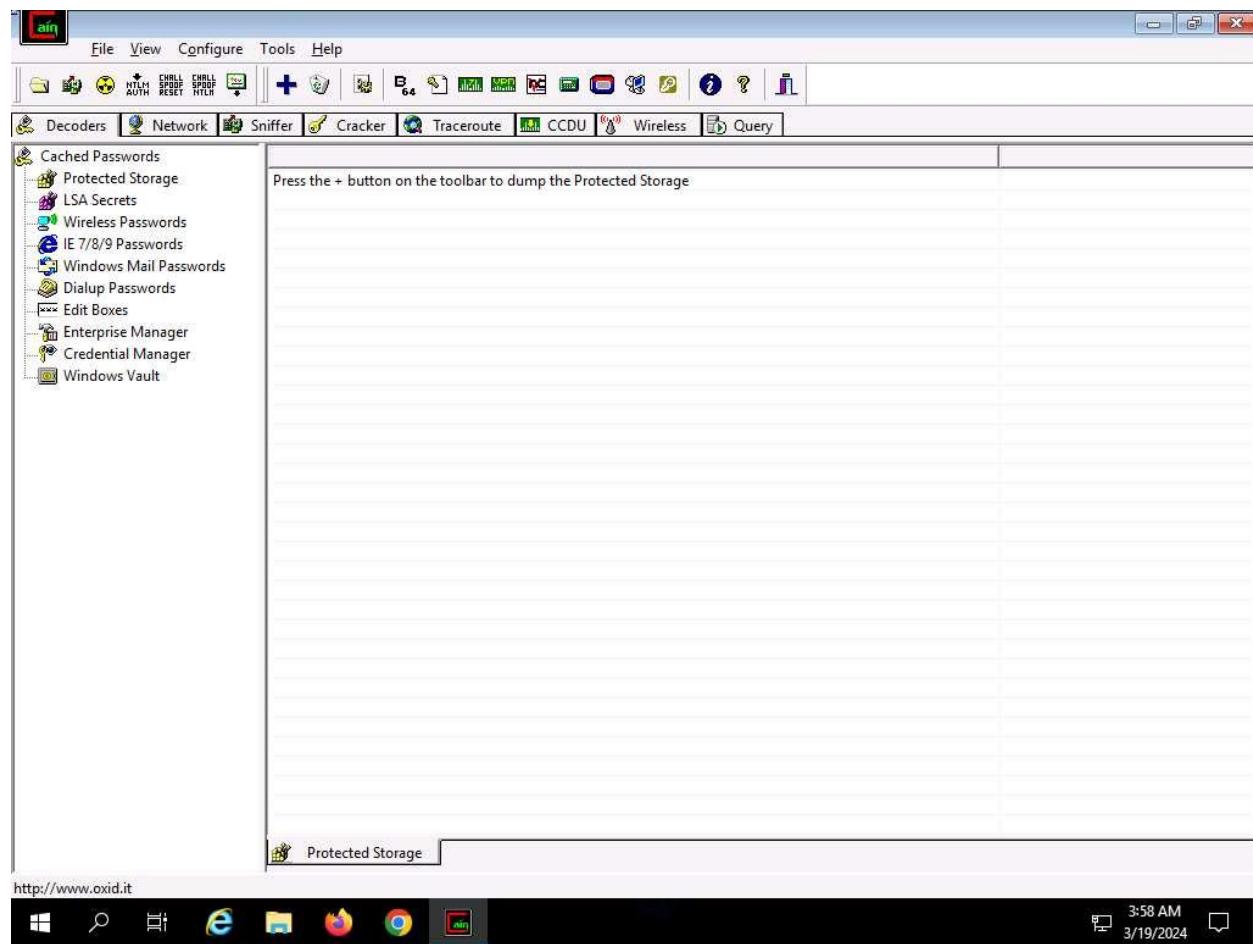
Promiscuous mode allows a network device to intercept and read each network packet that arrives in its entirety. The sniffer toggles the NIC of a system to promiscuous mode, so that it listens to all data transmitted on its segment. A sniffer can constantly monitor all network traffic to a computer through the NIC by decoding the information encapsulated in the data packet. Promiscuous mode in the network can be detected using various tools.

The ethical hacker and pen tester must assess the organization or target of evaluation for ARP poisoning vulnerabilities.

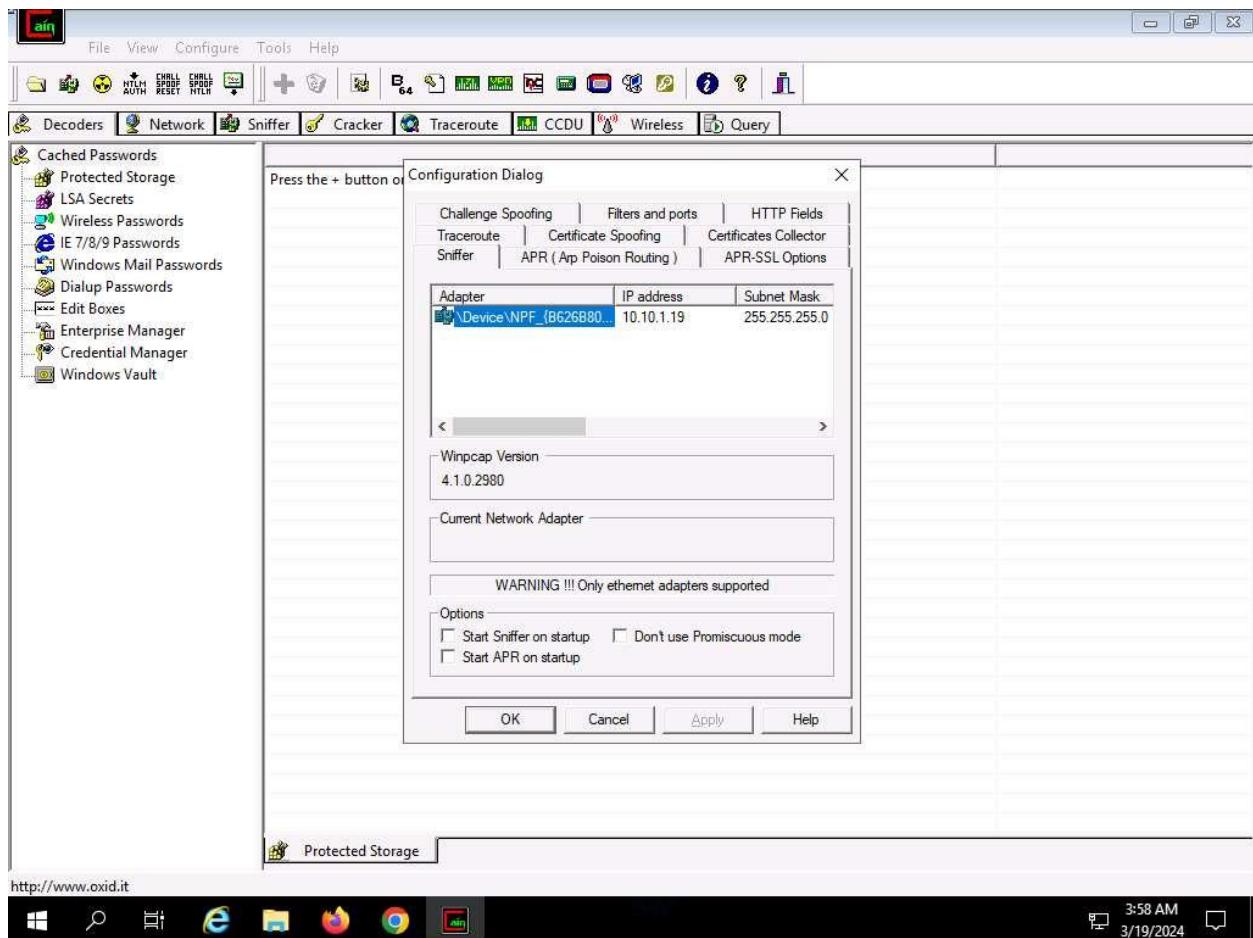
Here, we will detect ARP poisoning in a switch-based network using Wireshark and we will use the Nmap Scripting Engine (NSE) to check if a system on a local Ethernet has its network card in promiscuous mode.

In this task, we will use the **Windows Server 2019** machine as the host machine to perform ARP poisoning, and will sniff traffic flowing between the **Windows 11** and **Parrot Security** machines. We will use the same machine (**Windows Server 2019**) to detect ARP poisoning and use the **Windows 11** machine to detect promiscuous mode in the network.

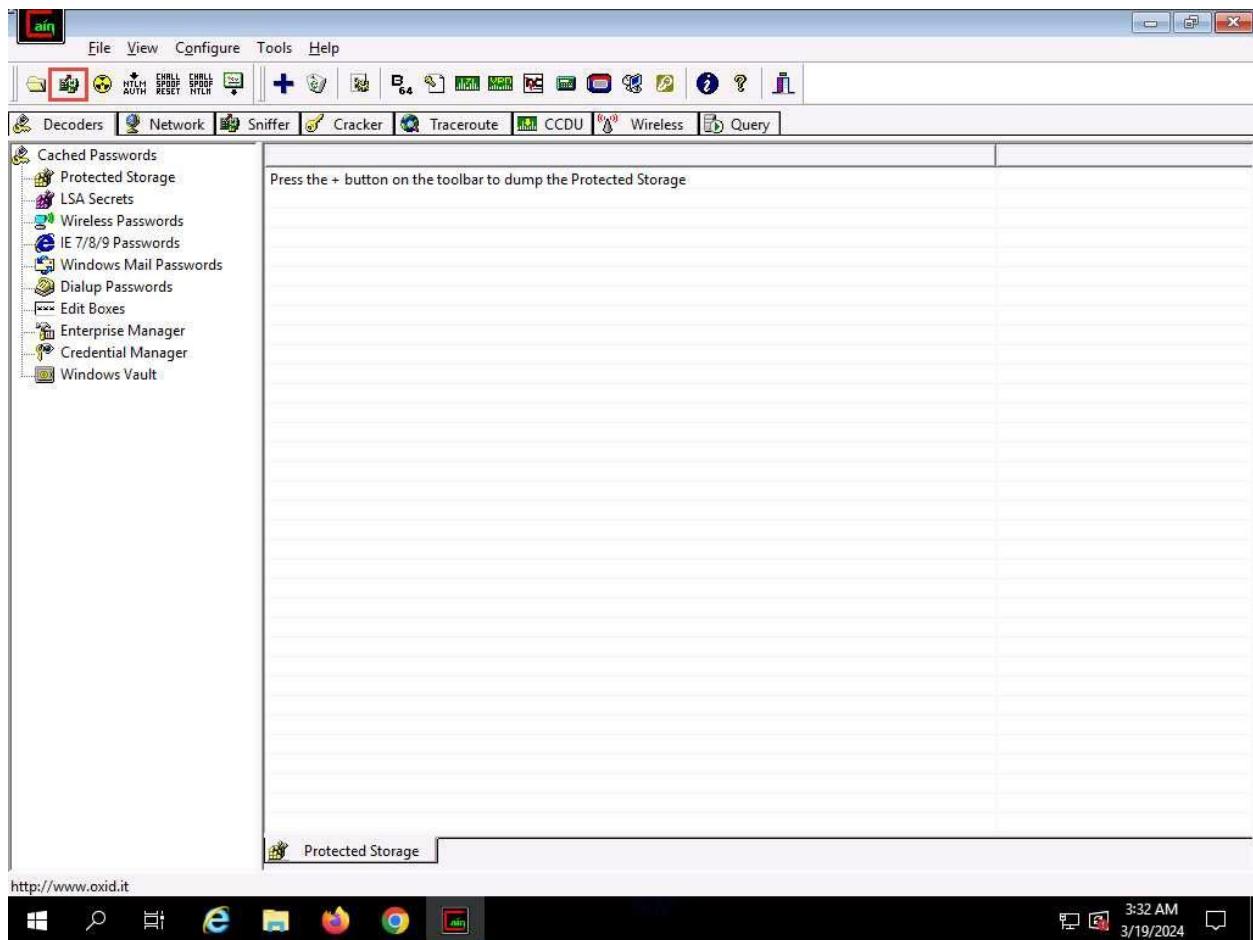
1. Click [Windows Server 2019](#) to switch to the **Windows Server 2019** machine.
2. In the **Desktop** window, click windows **Search** icon and search for **cain** in the search bar and launch it.
3. The **Cain & Abel** main window appears, click **Configure** from the menu bar to configure an ethernet card.



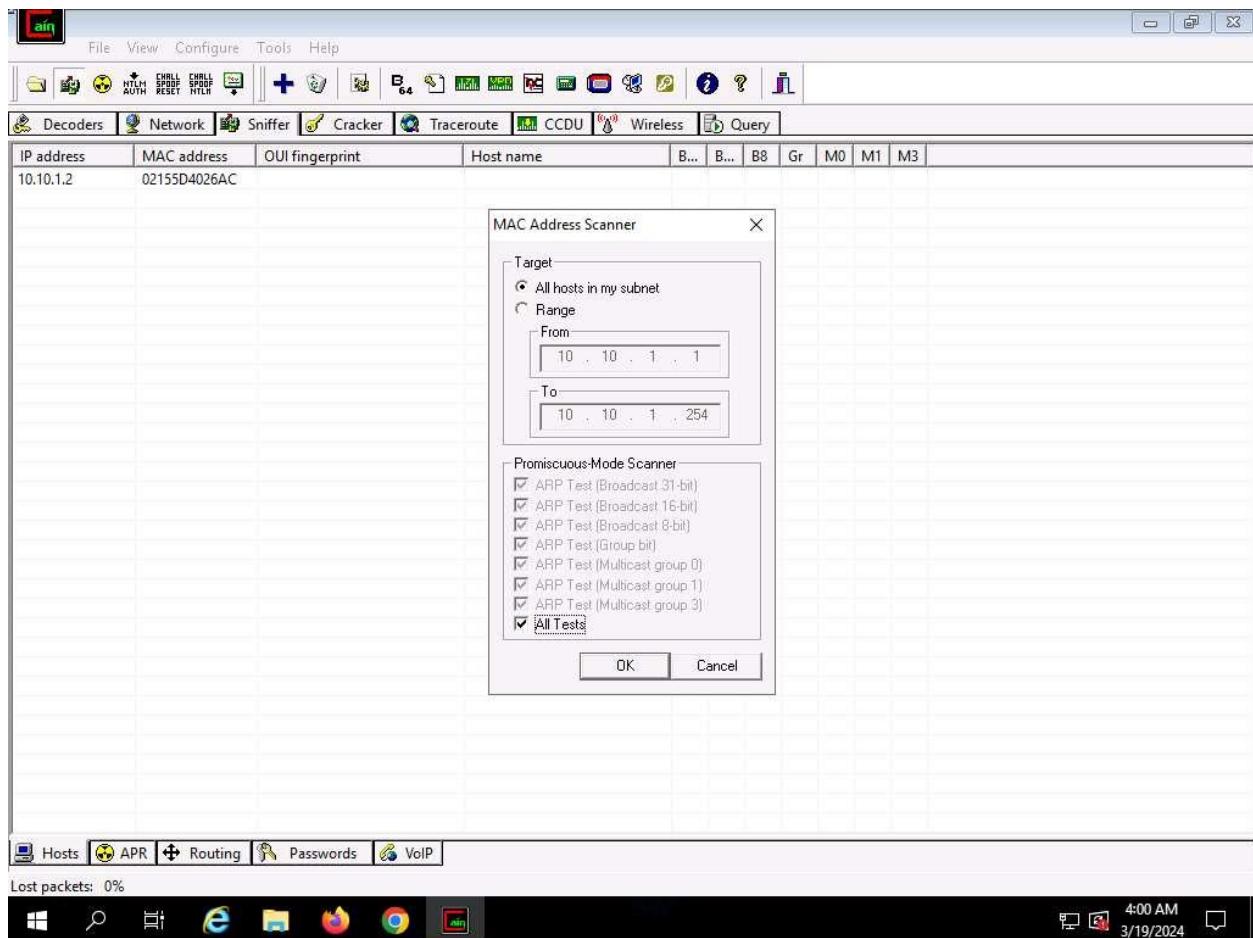
4. The **Configuration Dialog** window appears. The **Sniffer** tab is selected by default. Ensure that the **Adapter** associated with the **IP address** of the machine is selected and click **OK**.



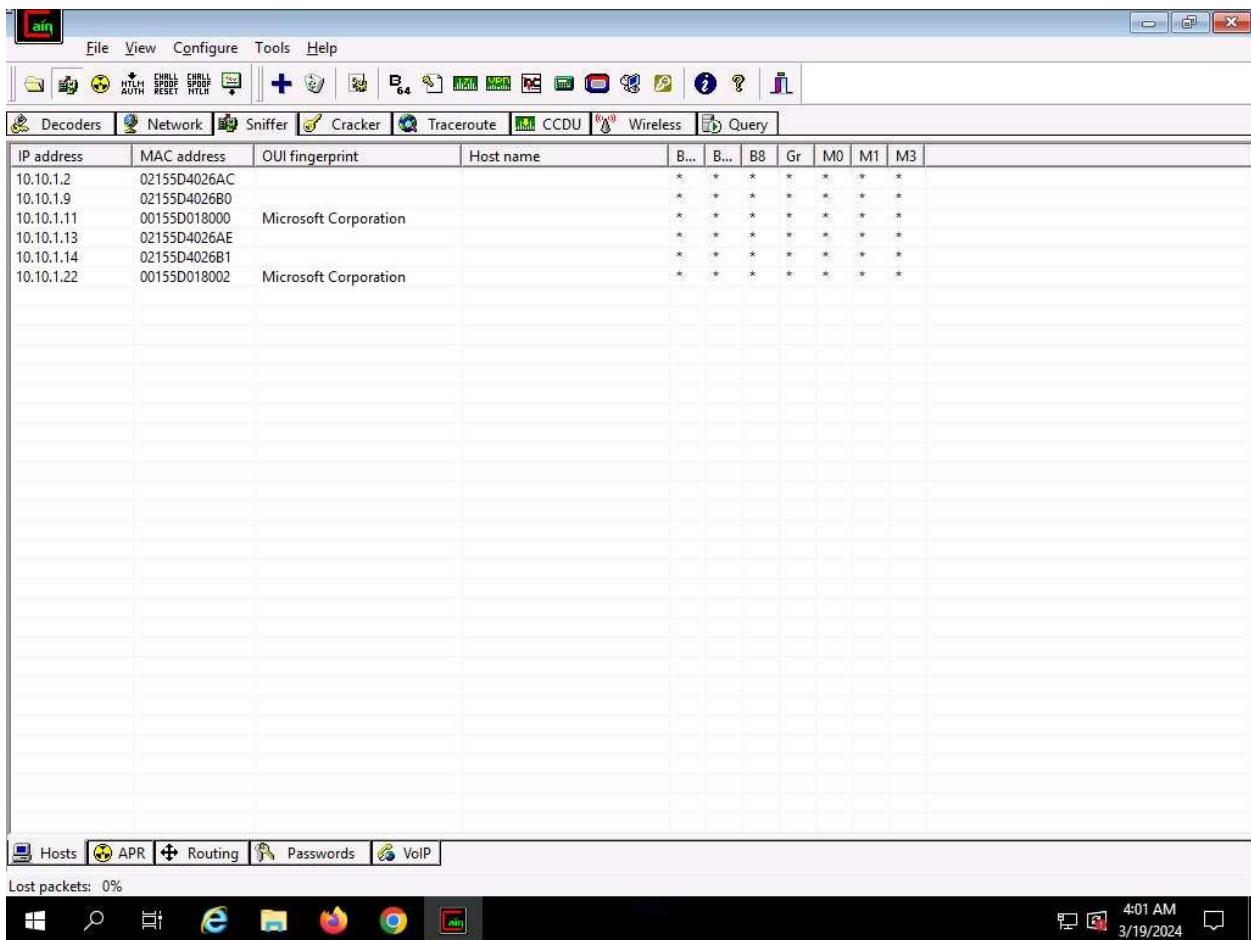
5. Click the **Start/Stop Sniffer** icon on the toolbar to begin sniffing.



6. The **Cain** pop-up appears with a **Warning** message, click **OK**.
7. Now, click the **Sniffer** tab.
8. Click the plus (+) icon or right-click in the window and select **Scan MAC Addresses** to scan the network for hosts.
9. The **MAC Address Scanner** window appears. Check the **All hosts in my subnet** radio button. Select the **All Tests** checkbox; then, click **OK**.

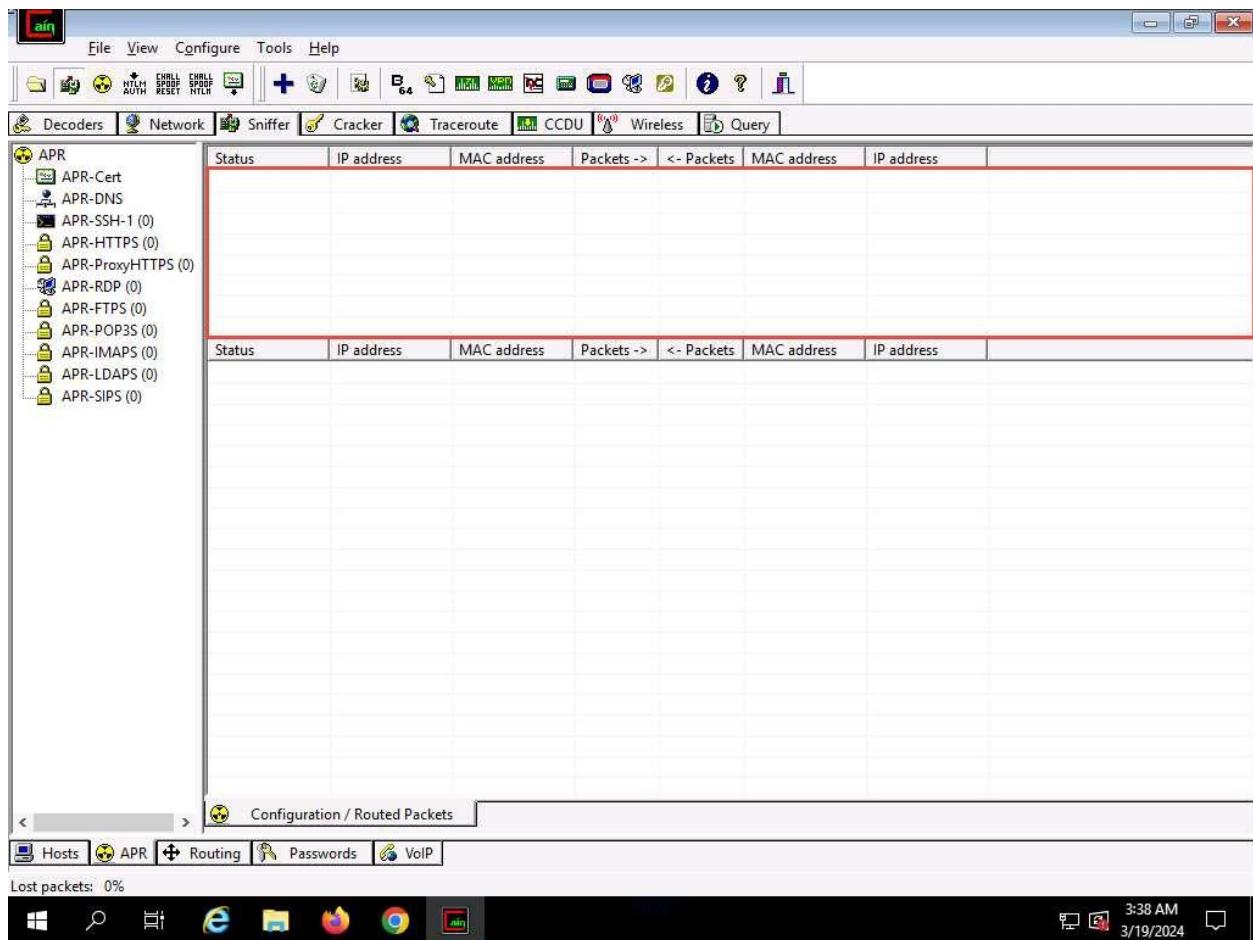


10. Cain & Abel starts scanning for MAC addresses and lists all those found.
11. After the completion of the scan, a list of all active IP addresses along with their corresponding MAC addresses is displayed, as shown in the screenshot.

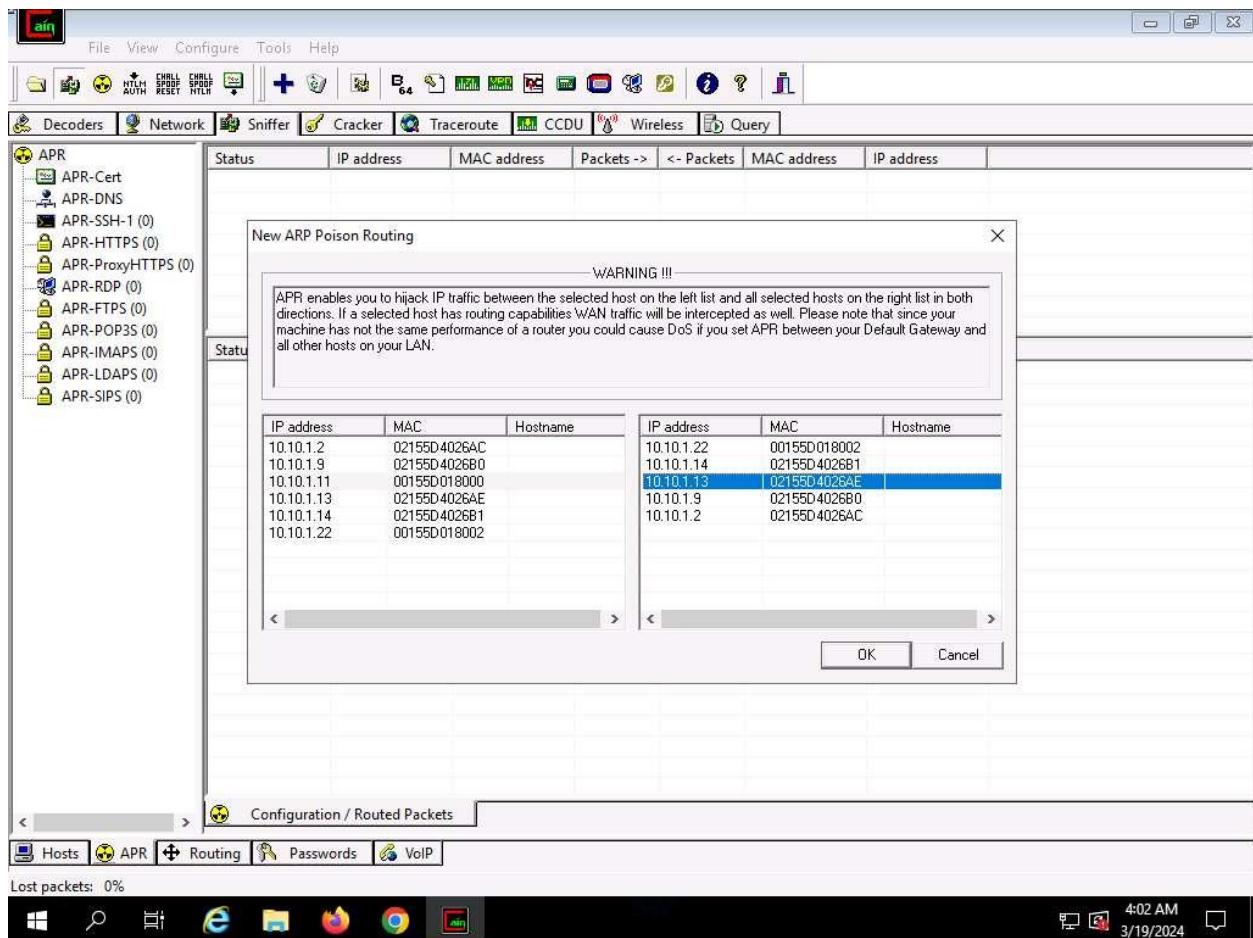


12. Now, click the **APR** tab at the bottom of the window.

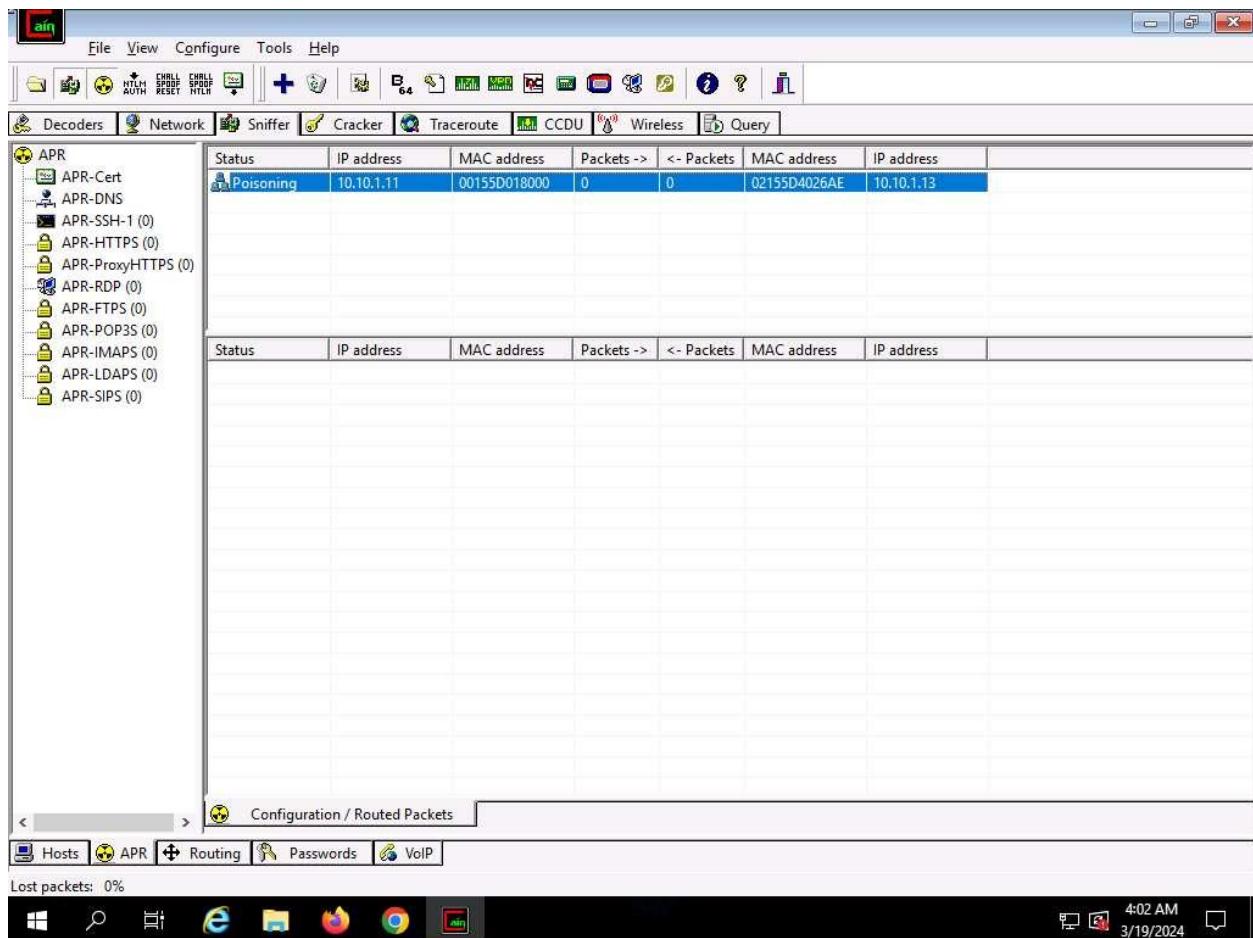
13. APR options appear in the left-hand pane. Click anywhere on the topmost section in the right-hand pane to activate the plus (+) icon.



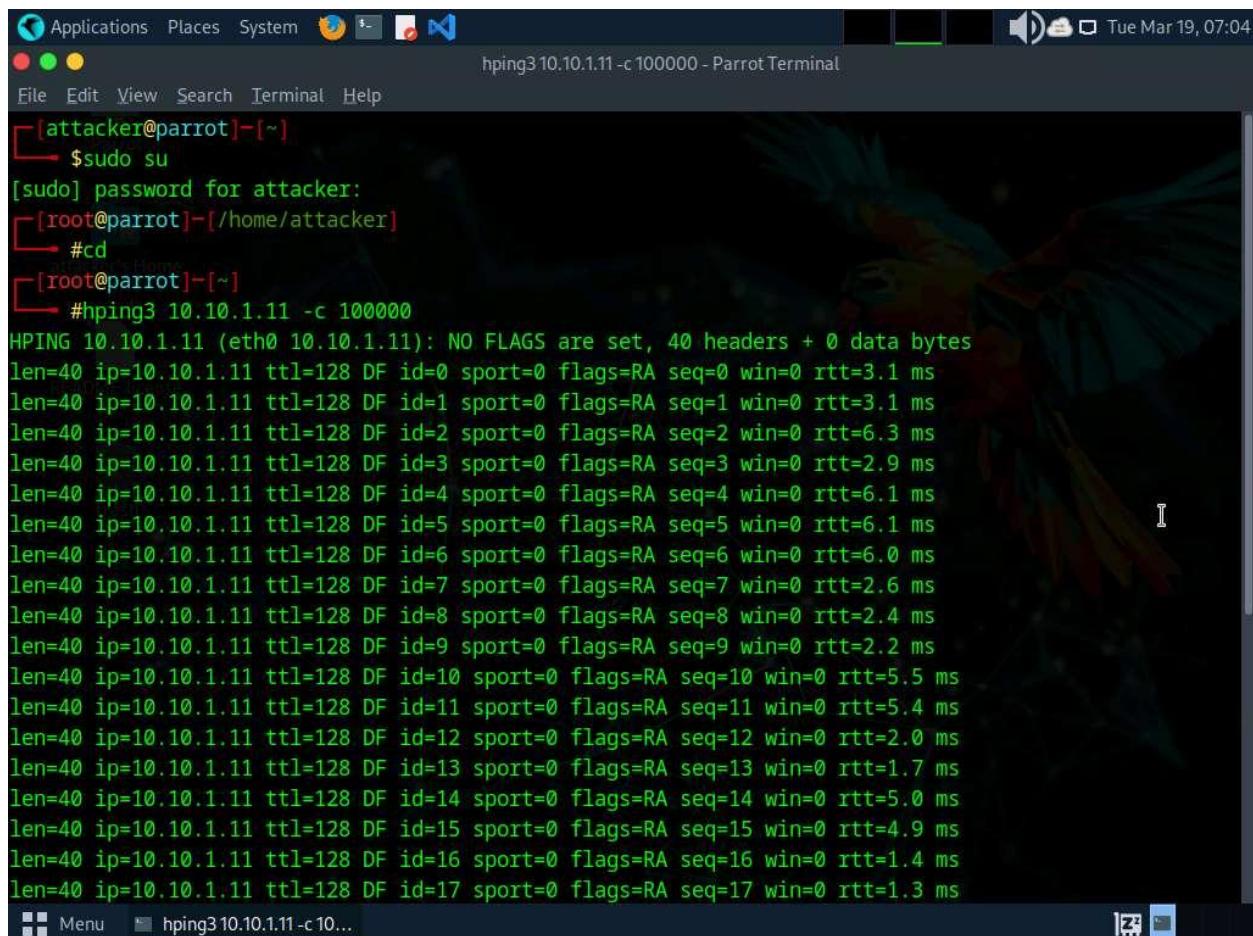
14. Click the plus (+) icon; a **New ARP Poison Routing** window appears; from which we can add IPs to listen to traffic.
15. To monitor the traffic between two systems (here, **Windows 11** and **Parrot Security**), from the left-hand pane, click to select **10.10.1.11 (Windows 11)** and from the right-hand pane, click **10.10.1.13 (Parrot Security)**; click **OK**. By doing so, you are setting Cain to perform ARP poisoning between the first and second targets.



16. Click to select the created target IP address scan that is displayed in the **Configuration / Routed Packets** tab.
17. Click on the **Start/Stop APR** icon to start capturing ARP packets.
18. After clicking on the **Start/Stop APR** icon, Cain & Abel starts ARP **poisoning** and the status of the scan changes to Poisoning, as shown in the screenshot.



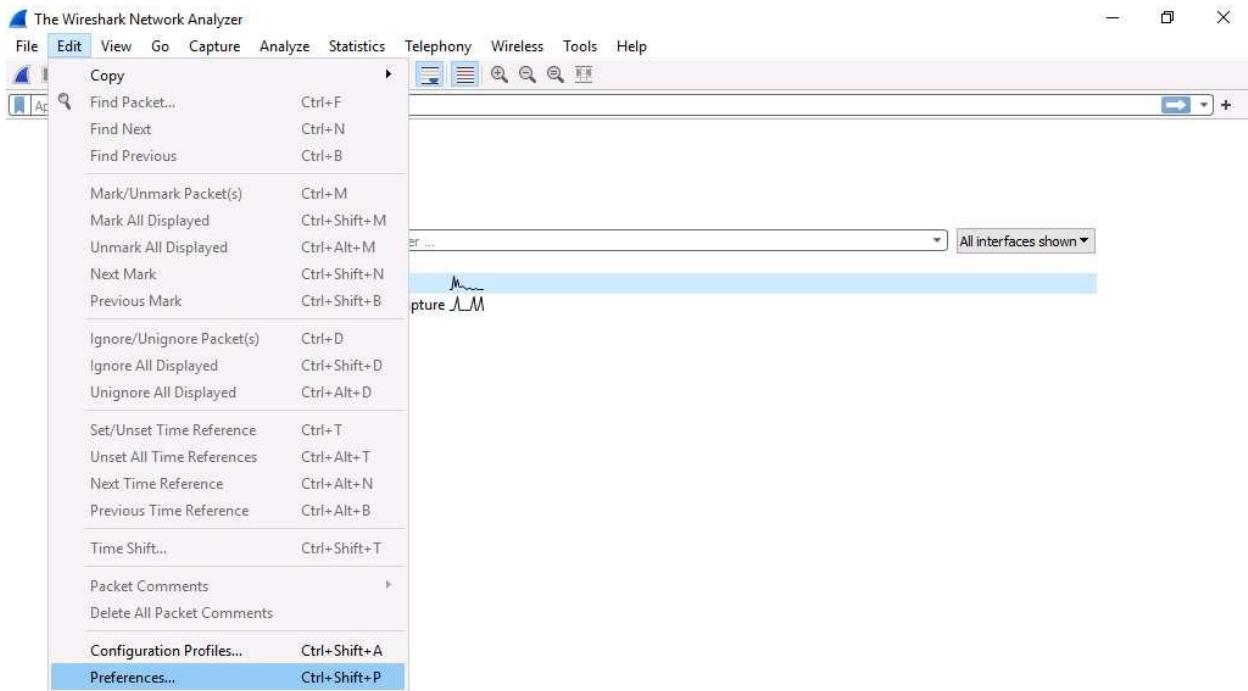
19. Cain & Abel intercepts the traffic traversing between these two machines.
  20. To generate traffic between the machines, you need to ping one target machine using the other.
  21. Click [Parrot Security](#) to switch to the **Parrot Security** machine.
  22. Open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**). Run **cd** command to jump to root directory.
  23. Run **hping3 [Target IP Address] -c 100000** command (here, target IP address is **10.10.1.11** [**Windows 11**]).
- c: specifies the packet count.
24. This command will start pinging the target machine (**Windows 11**) with 100,000 packets.



The screenshot shows a terminal window titled "hping3 10.10.1.11 -c 100000 - Parrot Terminal". The terminal session starts with the user "attacker" at the root prompt, entering "sudo su" to become root. Then, the user runs the command "#hping3 10.10.1.11 -c 100000", which performs a series of 100,000 TCP SYN scans on the target IP address. The output shows the details of each packet sent, including source and destination IP addresses, port numbers, sequence numbers, and round-trip times.

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
#cd
[root@parrot] ~
#hping3 10.10.1.11 -c 100000
HPING 10.10.1.11 (eth0 10.10.1.11): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=10.10.1.11 ttl=128 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=3.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=1 sport=0 flags=RA seq=1 win=0 rtt=3.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=2 sport=0 flags=RA seq=2 win=0 rtt=6.3 ms
len=40 ip=10.10.1.11 ttl=128 DF id=3 sport=0 flags=RA seq=3 win=0 rtt=2.9 ms
len=40 ip=10.10.1.11 ttl=128 DF id=4 sport=0 flags=RA seq=4 win=0 rtt=6.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=5 sport=0 flags=RA seq=5 win=0 rtt=6.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=6 sport=0 flags=RA seq=6 win=0 rtt=6.0 ms
len=40 ip=10.10.1.11 ttl=128 DF id=7 sport=0 flags=RA seq=7 win=0 rtt=2.6 ms
len=40 ip=10.10.1.11 ttl=128 DF id=8 sport=0 flags=RA seq=8 win=0 rtt=2.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=9 sport=0 flags=RA seq=9 win=0 rtt=2.2 ms
len=40 ip=10.10.1.11 ttl=128 DF id=10 sport=0 flags=RA seq=10 win=0 rtt=5.5 ms
len=40 ip=10.10.1.11 ttl=128 DF id=11 sport=0 flags=RA seq=11 win=0 rtt=5.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=12 sport=0 flags=RA seq=12 win=0 rtt=2.0 ms
len=40 ip=10.10.1.11 ttl=128 DF id=13 sport=0 flags=RA seq=13 win=0 rtt=1.7 ms
len=40 ip=10.10.1.11 ttl=128 DF id=14 sport=0 flags=RA seq=14 win=0 rtt=5.0 ms
len=40 ip=10.10.1.11 ttl=128 DF id=15 sport=0 flags=RA seq=15 win=0 rtt=4.9 ms
len=40 ip=10.10.1.11 ttl=128 DF id=16 sport=0 flags=RA seq=16 win=0 rtt=1.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=17 sport=0 flags=RA seq=17 win=0 rtt=1.3 ms
```

25. Leave the command running and immediately click [Windows Server 2019](#) to switch to the **Windows Server 2019** machine.
26. In the **Desktop** window, click windows **Search** icon and search for **wireshark** in the search bar and launch it.
27. The **Wireshark Network Analyzer** window appears; click **Edit** in the menu bar and select **Preferences....**



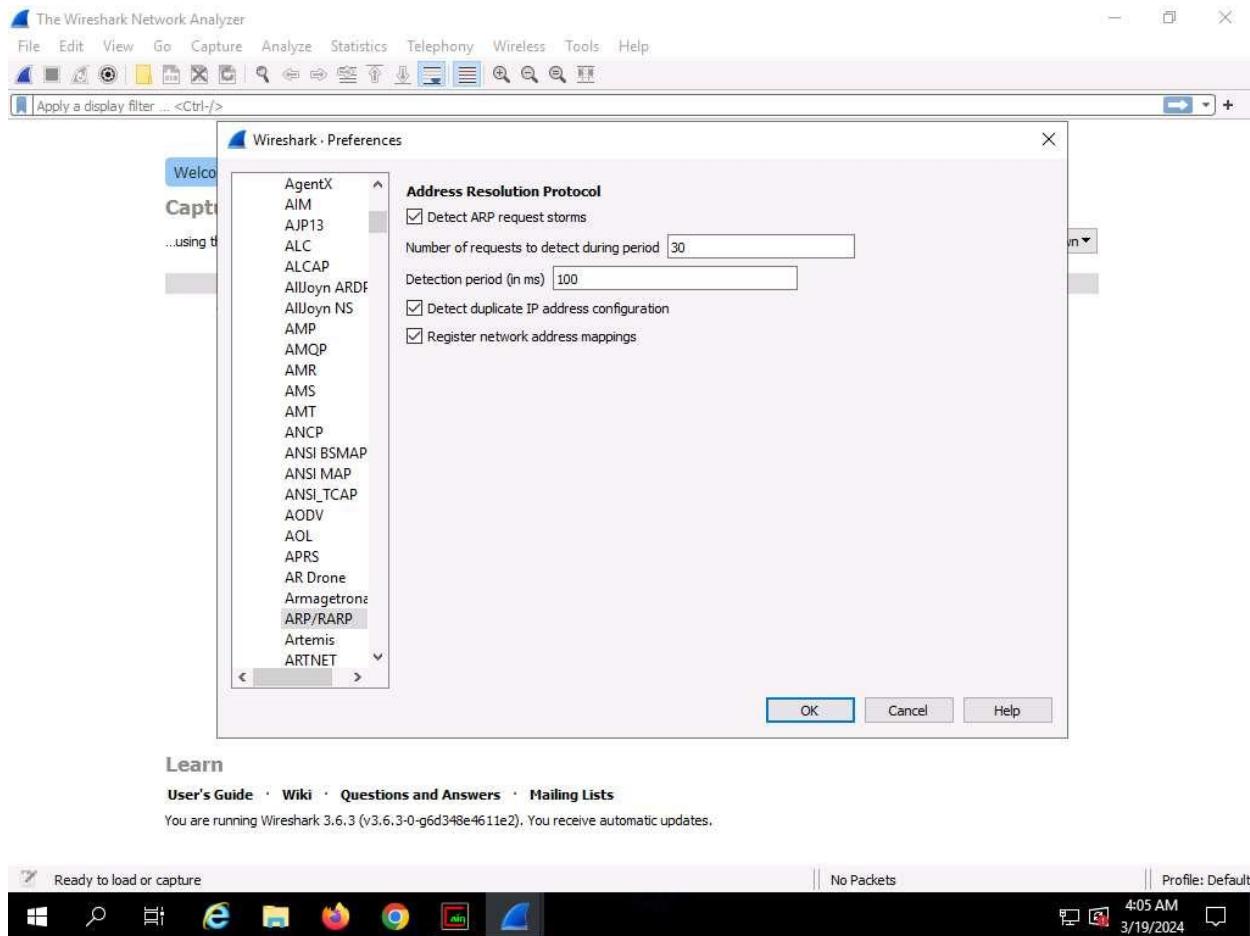
### Learn

[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#)

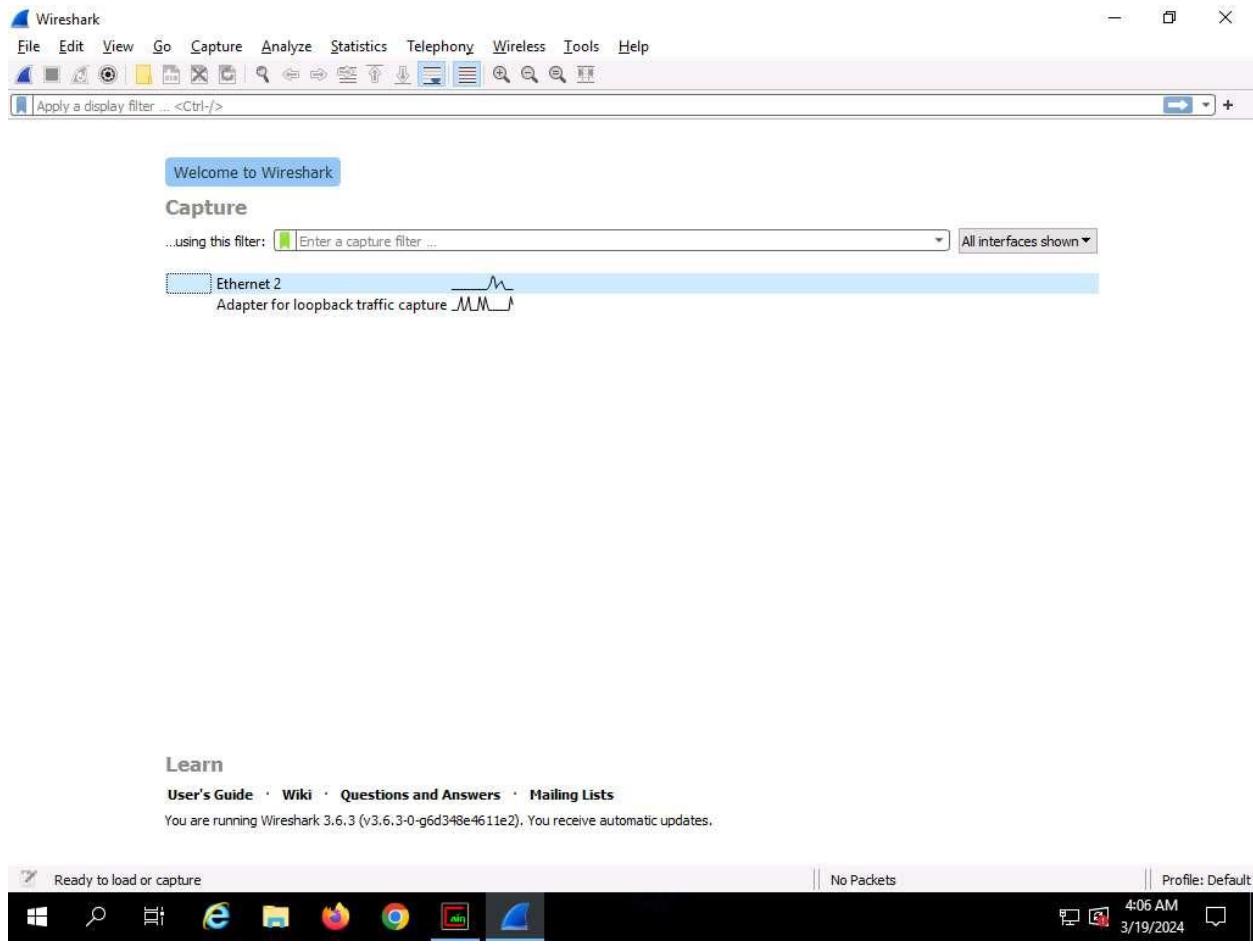
You are running Wireshark 3.6.3 (v3.6.3-0-g6d348e4611e2). You receive automatic updates.



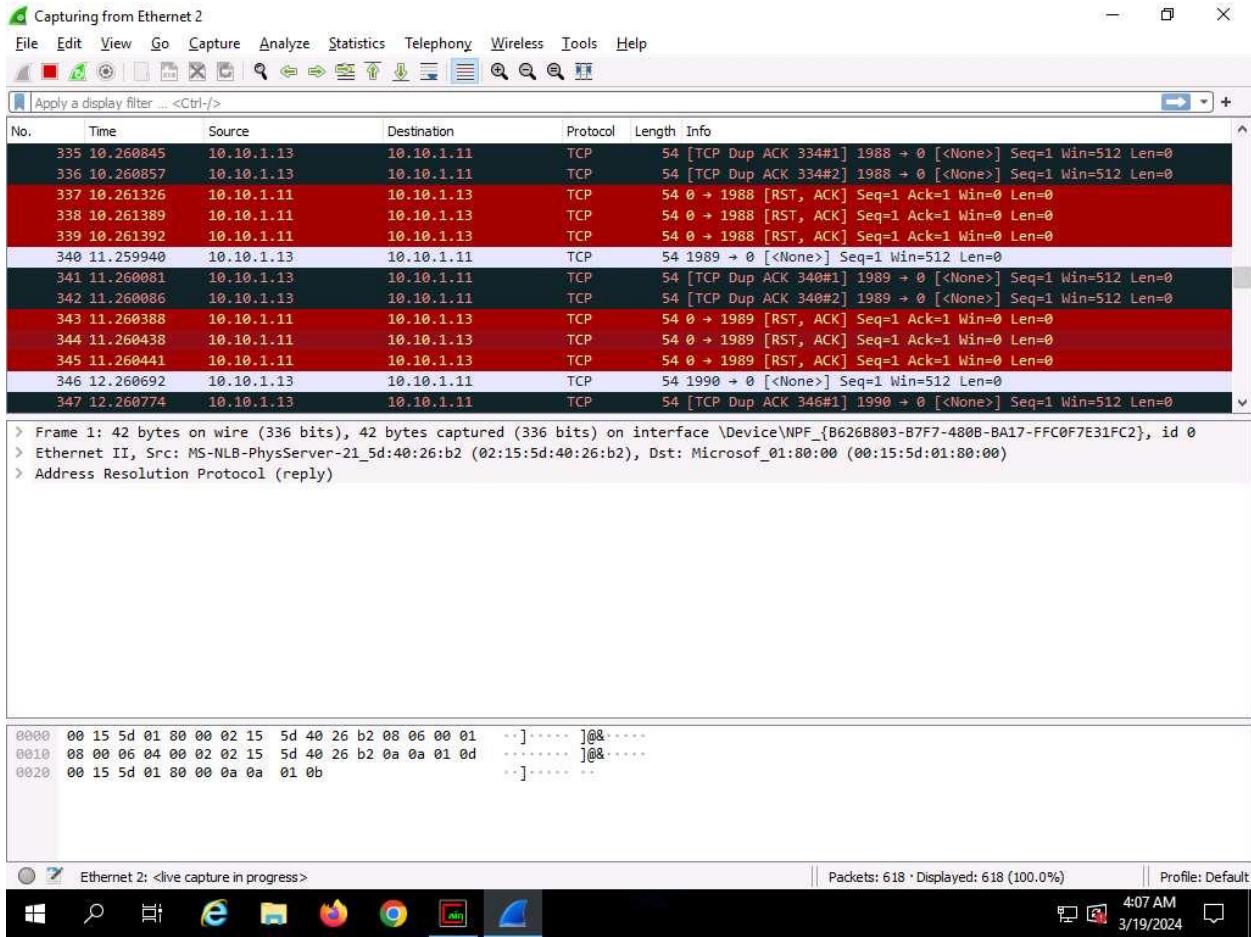
28. The **Wireshark . Preferences** window appears; expand the **Protocols** node.
29. Scroll-down in the **Protocols** node and select the **ARP/RARP** option.
30. From the right-hand pane, click the **Detect ARP request storms** checkbox and ensure that the **Detect duplicate IP address configuration** checkbox is checked; click **OK**.



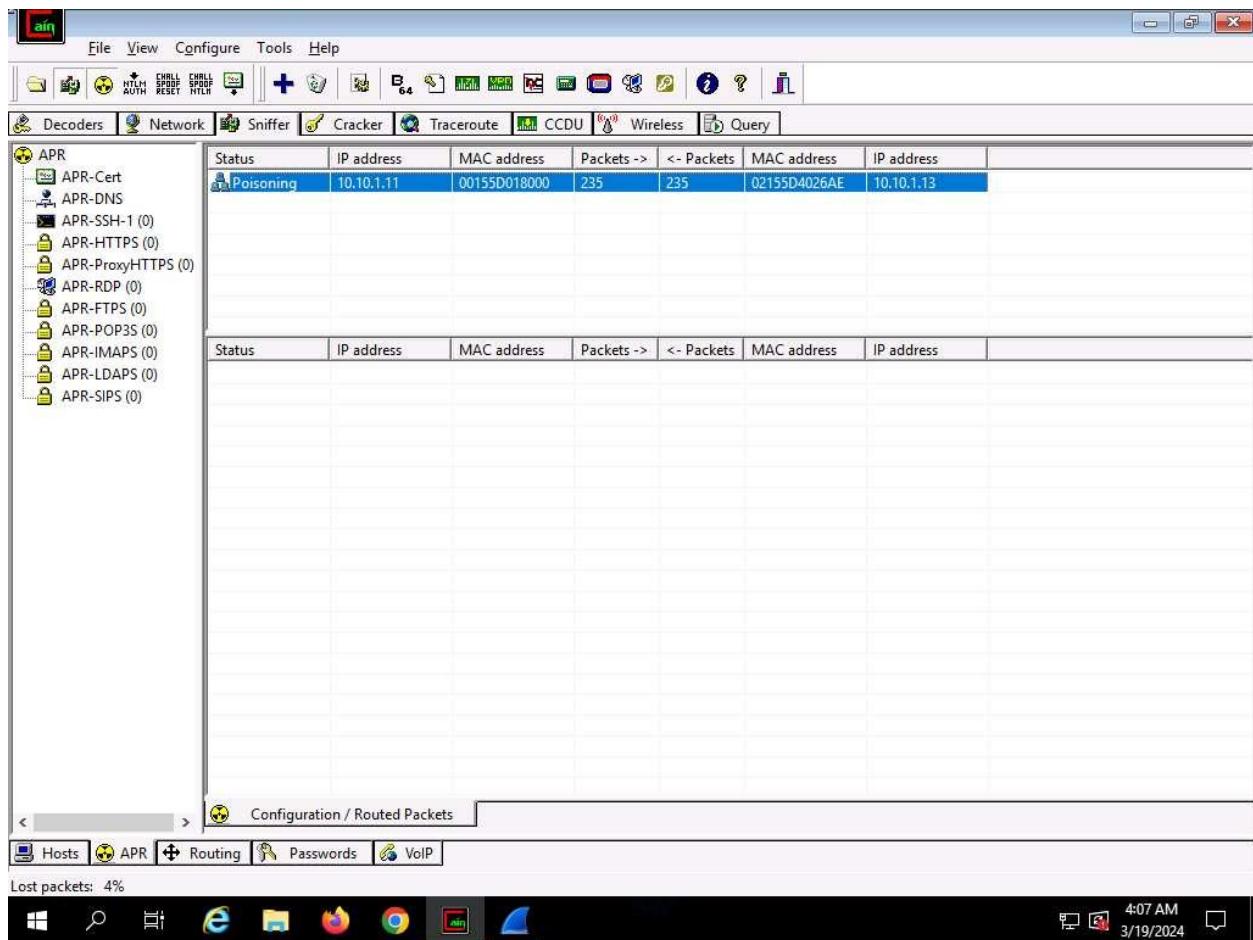
31. Now, double-click on the adapter associated with your network (here, **Ethernet2**) to start capturing the network packets.



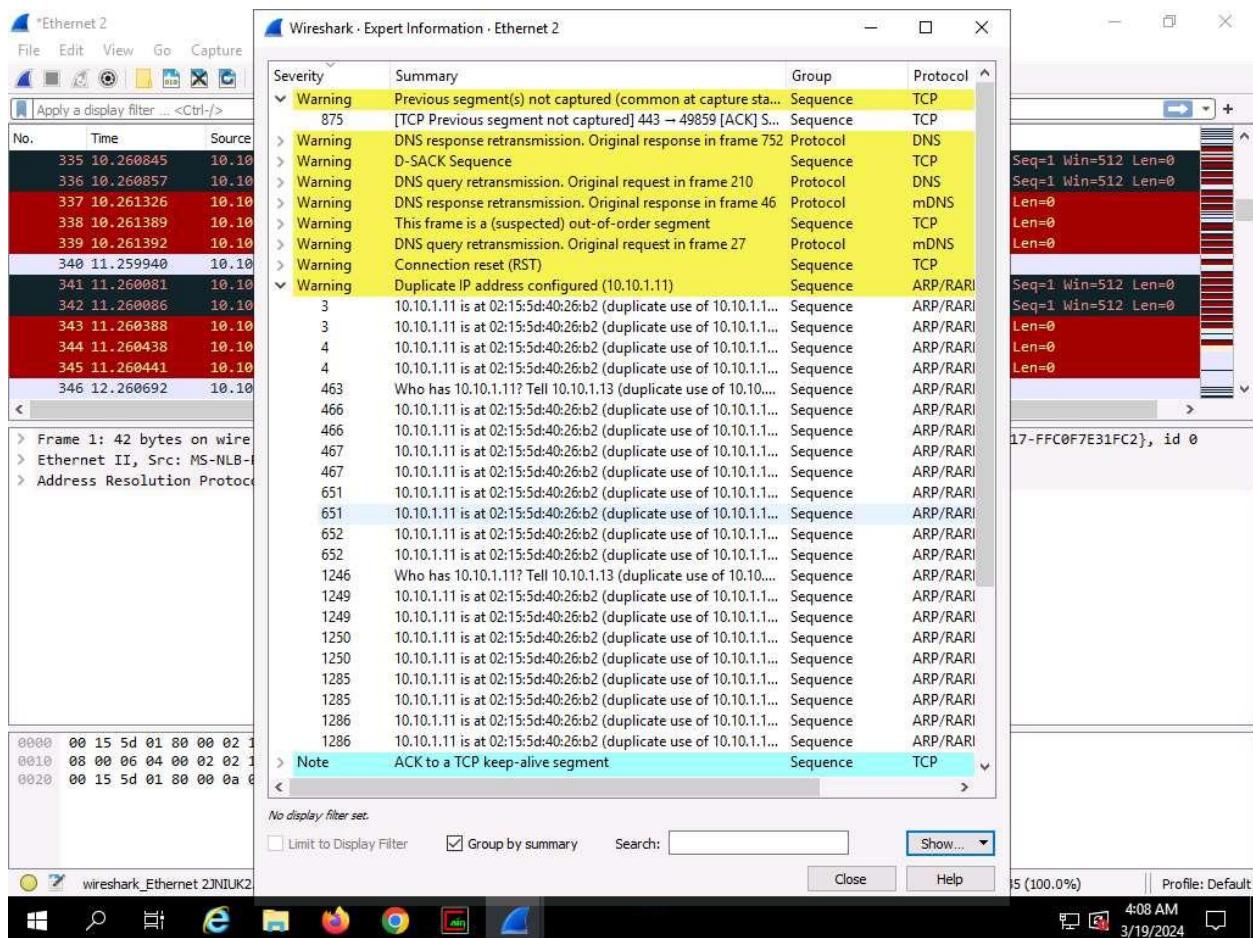
32. **Wireshark** begins to capture the traffic between the two machines, as shown in the screenshot.



33. Switch to the **Cain & Abel** window to observe the packets flowing between the two machines.

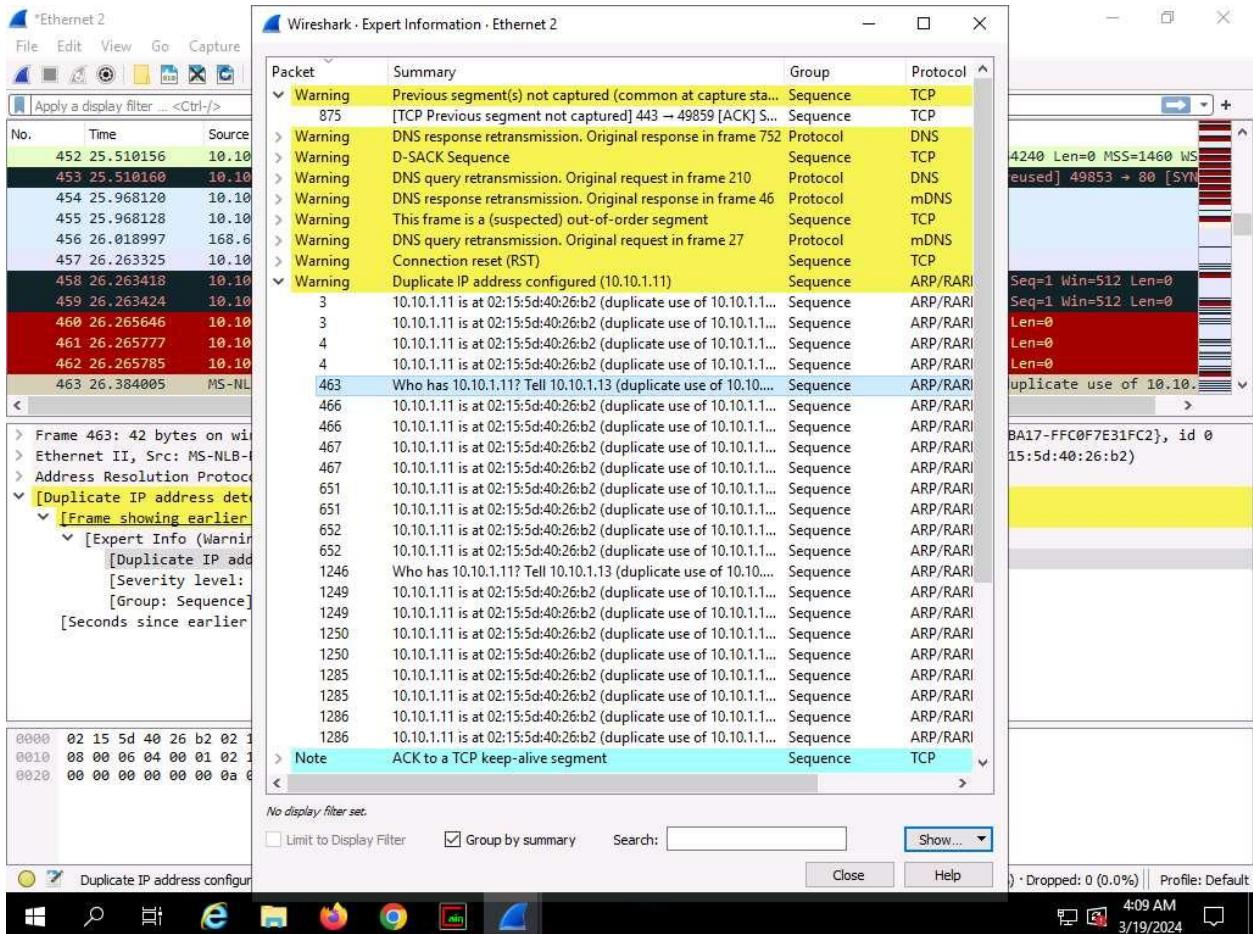


34. Now, switch to **Wireshark** and click the **Stop packet capturing** icon to stop the packet capturing.
35. Click **Analyze** from the menu bar and select **Expert Information** from the drop-down options. The **Wireshark . Expert Information** window appears; click to expand the **Warning** node labeled **Duplicate IP address configured (10.10.1.11)**, running on the **ARP/RARP** protocol.

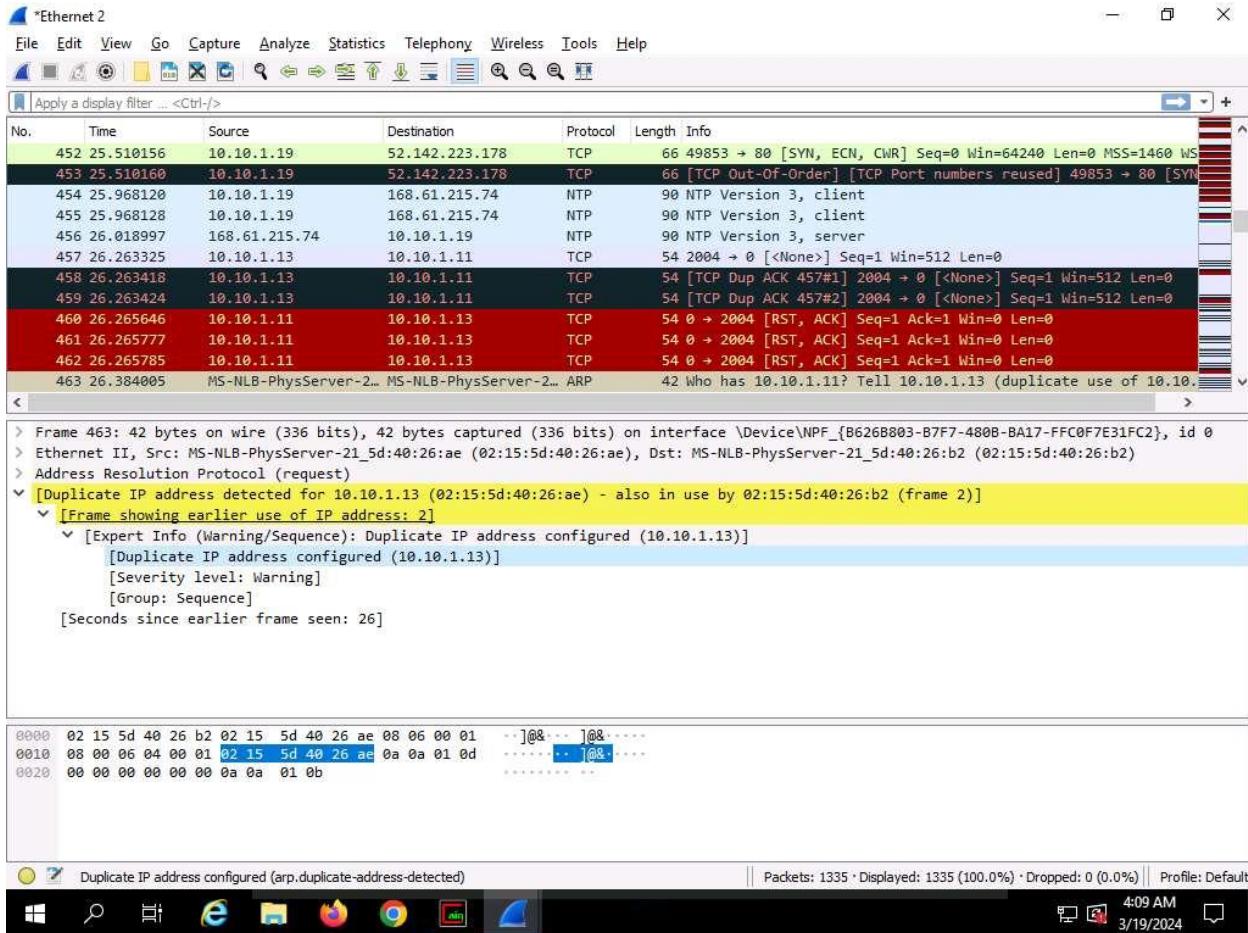


36. Arrange the **Wireshark . Expert Information** window above the **Wireshark** window so that you can view the packet number and the **Packet details** section.

37. In the **Wireshark . Expert Information** window, click any packet (here, 463).

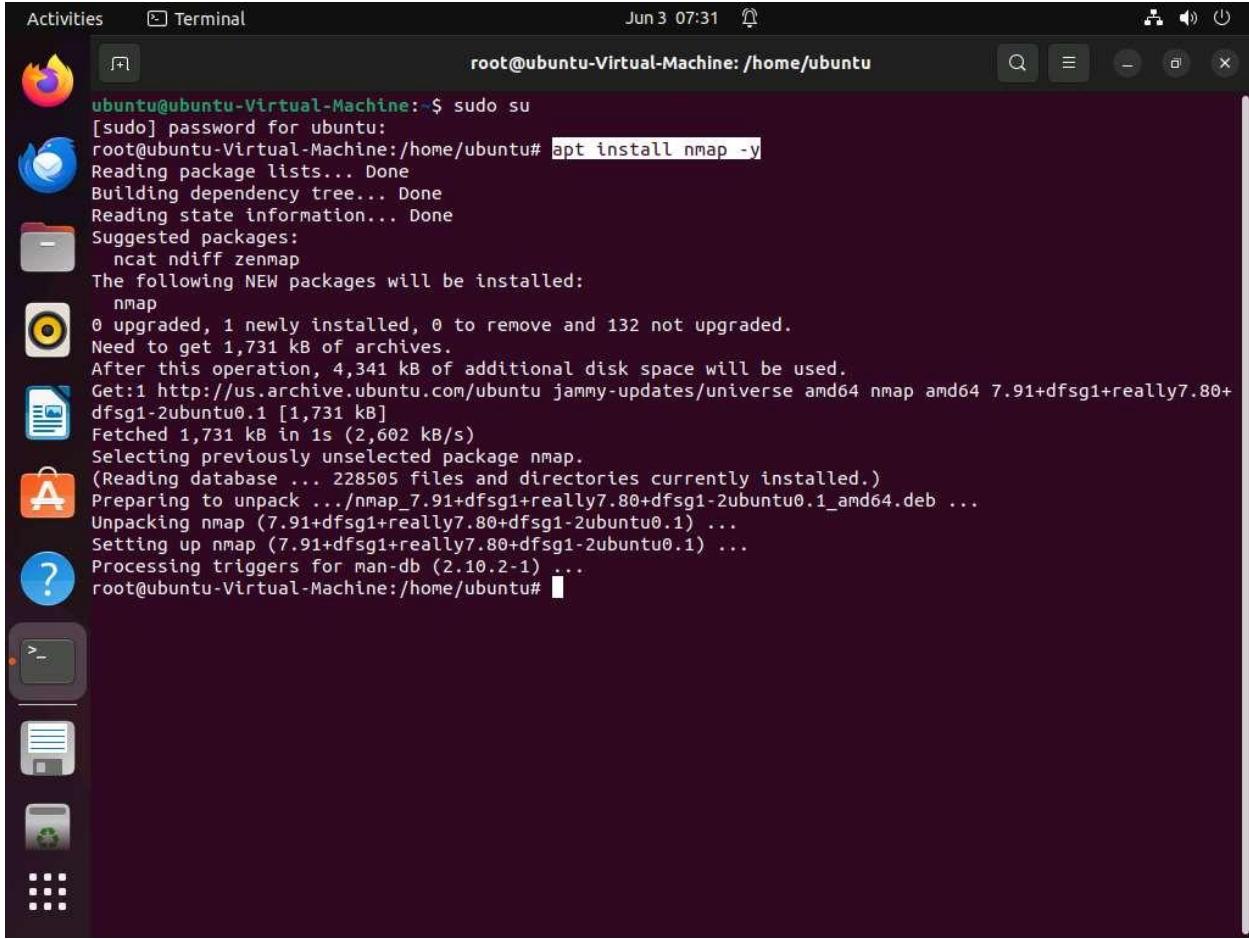


38. On selecting the packet number, **Wireshark** highlights the packet, and its associated information is displayed under the packet details section. Close the **Wireshark . Expert Information** window.
  39. The warnings highlighted in yellow indicate that duplicate IP addresses have been detected at one MAC address, as shown in the screenshot.



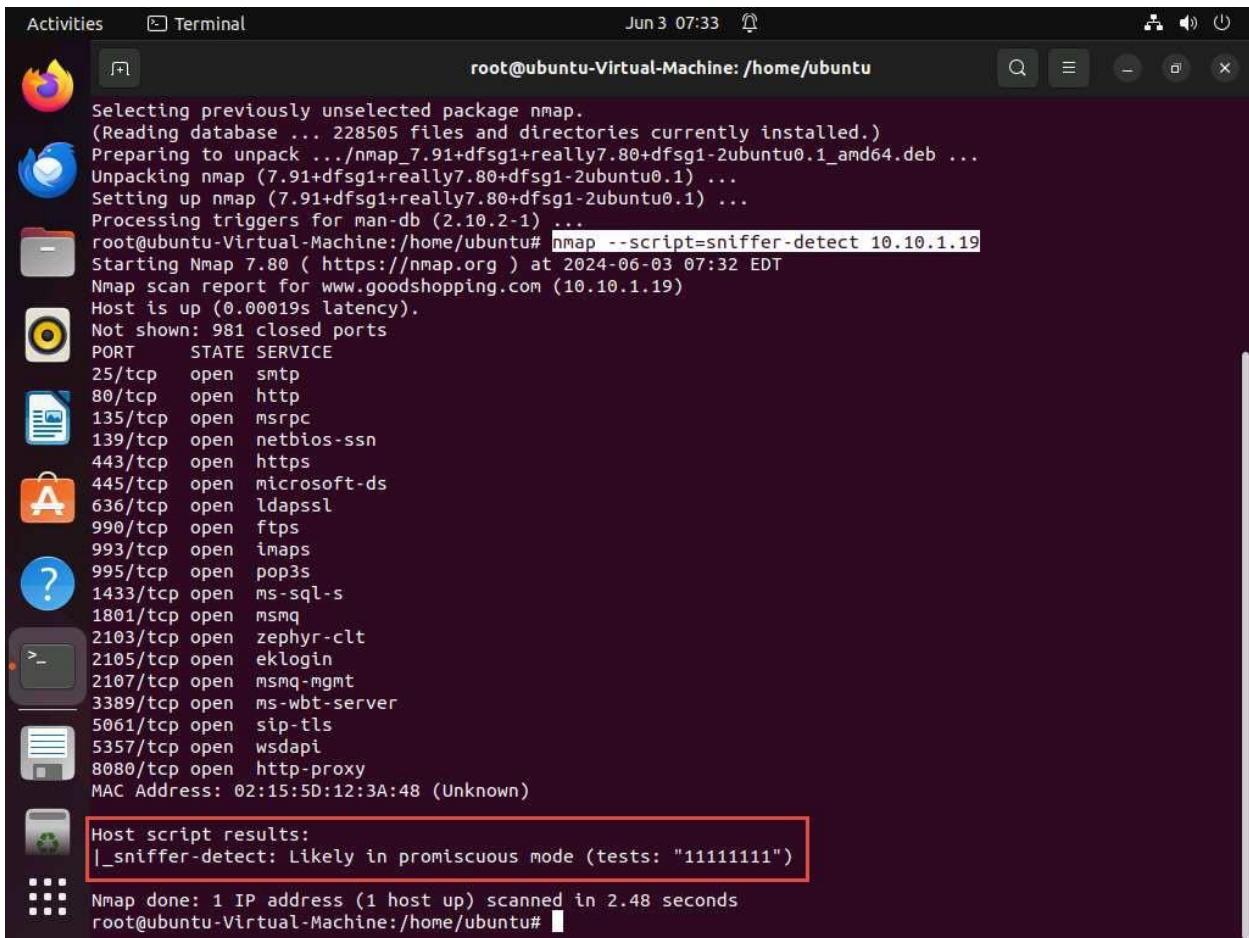
ARP spoofing succeeds by changing the IP address of the attacker's computer to the IP address of the target computer. A forged ARP request and reply packet find a place in the target ARP cache in this process. As the ARP reply has been forged, the destination computer (target) sends frames to the attacker's computer, where the attacker can modify the frames before sending them to the source machine (User A) in an MITM attack. At this point, the attacker can launch a DoS attack by associating a non-existent MAC address with the IP address of the gateway or may passively sniff the traffic, and then forward it to the target destination.

40. This concludes the demonstration of detecting ARP poisoning in a switch-based network.
41. Close the **Wireshark** window and leave all other windows running.
42. Now, we shall perform promiscuous mode detection using Nmap.
43. Now, Click [Ubuntu](#) to switch to the **Ubuntu** machine and login with **Ubuntu/toor**.
44. In the **Ubuntu** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**)
45. Run **apt install nmap -y** command to install **nmap**.

A screenshot of an Ubuntu desktop environment. On the left, there's a dock with icons for Dash, Home, File Manager, Terminal, and others. A terminal window is open at the top, showing root privileges. The terminal output is as follows:

```
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# apt install nmap -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  ncat ndiff zemmap
The following NEW packages will be installed:
  nmap
0 upgraded, 1 newly installed, 0 to remove and 132 not upgraded.
Need to get 1,731 kB of archives.
After this operation, 4,341 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap amd64 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1 [1,731 kB]
Fetched 1,731 kB in 1s (2,602 kB/s)
Selecting previously unselected package nmap.
(Reading database ... 228505 files and directories currently installed.)
Preparing to unpack .../nmap_7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1_amd64.deb ...
Unpacking nmap (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Setting up nmap (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu-Virtual-Machine:/home/ubuntu#
```

46. Run **nmap --script=sniffer-detect [Target IP Address/ IP Address Range]** (here, target IP address is **10.10.1.19 [Windows Server 2019]**) to start scanning.
47. The scan results appear, displaying **Likely in promiscuous mode** under the **Host script results** section. This indicates that the target system is in promiscuous mode.

A screenshot of a Ubuntu desktop environment. On the left, there's a dock with icons for various applications like a browser, file manager, and terminal. The main window is a terminal window titled "root@ubuntu-Virtual-Machine:/home/ubuntu". The terminal shows the following Nmap command and its output:

```
Selecting previously unselected package nmap.  
(Reading database ... 228505 files and directories currently installed.)  
Preparing to unpack .../nmap_7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1_amd64.deb ...  
Unpacking nmap (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...  
Setting up nmap (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...  
Processing triggers for man-db (2.10.2-1) ...  
root@ubuntu-Virtual-Machine:/home/ubuntu# nmap --script=sniffer-detect 10.10.1.19  
Starting Nmap 7.80 ( https://nmap.org ) at 2024-06-03 07:32 EDT  
Nmap scan report for www.goodshopping.com (10.10.1.19)  
Host is up (0.00019s latency).  
Not shown: 981 closed ports  
PORT      STATE SERVICE  
25/tcp    open  smtp  
80/tcp    open  http  
135/tcp   open  msrpc  
139/tcp   open  netbios-ssn  
443/tcp   open  https  
445/tcp   open  microsoft-ds  
636/tcp   open  ldapssl  
990/tcp   open  ftps  
993/tcp   open  imaps  
995/tcp   open  pop3s  
1433/tcp  open  ms-sql-s  
1801/tcp  open  msmq  
2103/tcp  open  zephyr-clt  
2105/tcp  open  eklogin  
2107/tcp  open  msmq-mgmt  
3389/tcp  open  ms-wbt-server  
5061/tcp  open  sip-tls  
5357/tcp  open  wsdapi  
8080/tcp  open  http-proxy  
MAC Address: 02:15:5D:12:3A:48 (Unknown)  
  
Host script results:  
|_sniffer-detect: Likely in promiscuous mode (tests: "11111111")  
  
Nmap done: 1 IP address (1 host up) scanned in 2.48 seconds
```

48. Close the terminal window and document all the acquired information.
49. Close all open windows in all machines (ensure that ARP poisoning is not running in **Windows Server 2019**), and document all the acquired information.

#### Question 8.3.1.1

Use Cain and Abel on the Windows Server 2019 machine to perform ARP poisoning, and sniff traffic between the Windows 11 and Parrot Security machines. Further, use Wireshark on the same Windows Server 2019 machine to detect ARP poisoning. What is the severity level of ARP/RARP packets as shown in the expert information window of Wireshark?

#### Question 8.3.1.2

Use the Nmap Scripting Engine (NSE) to check if a system on the local Ethernet has its network card in the promiscuous mode. Which Nmap NSE script detects if a network interface is in the promiscuous mode?