EXPERIMENT 09

Aim: Detect ARP spoofing using nmap and/or open source tool ARPWATCH and wireshark **Objectives**:

- To understand ARP spoofing.
- To understand ARPWATCH and use it to detect ARP spoofing.

Theory:

1. **Nmap (Network Mapper):**

While Nmap isn't specifically designed for ARP spoofing detection, it can be used indirectly. Nmap can perform a quick network scan to identify active devices and their MAC addresses. You can then compare this information with the ARP table on your machine (using arp -a on Linux/macOS) to identify any discrepancies. For example, if Nmap identifies a device with a specific IP address but the ARP table shows a different MAC address associated with that IP, it might indicate ARP spoofing. However, this method can be unreliable as legitimate network configurations can also cause MAC address changes.

2. **Arpwatch**:

Arpwatch is a dedicated tool for monitoring ARP activity on your network. It keeps track of learned MAC addresses for IPs and monitors for any changes. Here's how it helps detect ARP spoofing:

Database: Arpwatch maintains a database of learned IP/MAC mappings.

Monitoring: It continuously monitors ARP packets on the network.

Alerting: If Arpwatch detects an unsolicited ARP reply (attacker trying to modify the ARP table) or a change in the MAC address associated with a known IP, it raises an alert in the system logs.

3. **Wireshark**:

Wireshark is a powerful network packet analyzer. While not solely for ARP spoofing detection, it can be used for in-depth analysis of network traffic. Here's how it helps:

Packet Capture: Wireshark can capture live network traffic.

Filtering: You can filter captured packets to focus specifically on ARP traffic.

Analysis: By examining ARP packets, you can identify inconsistencies. For instance, if you see multiple ARP replies for the same IP address with different MAC addresses, it might indicate ARP spoofing.

Implementation :

Using nmap

`nmap` is primarily a network scanning tool, but it can be used to detect ARP spoofing by checking for duplicate IP addresses or multiple MAC addresses associated with a single IP.

1. Scan the Network:

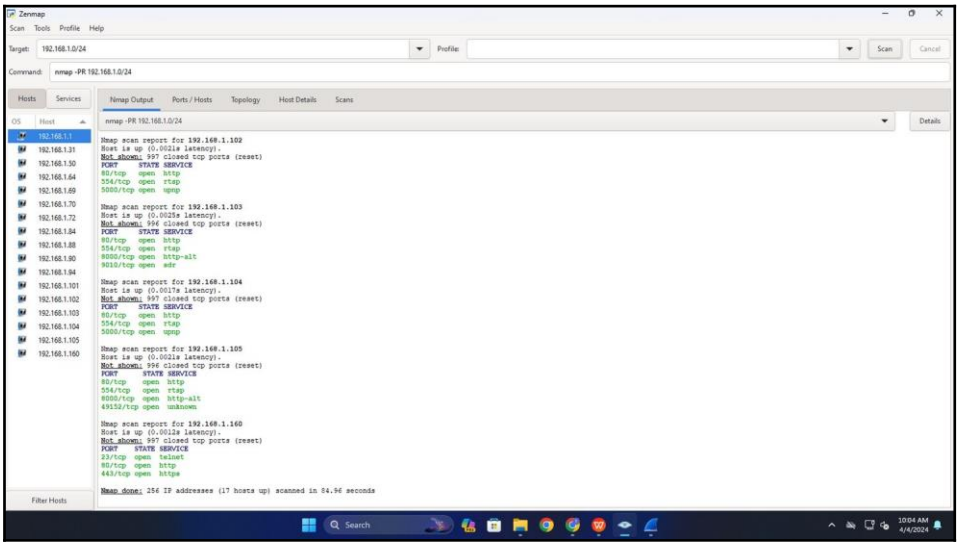
Use `nmap` to scan your network and identify all live hosts.

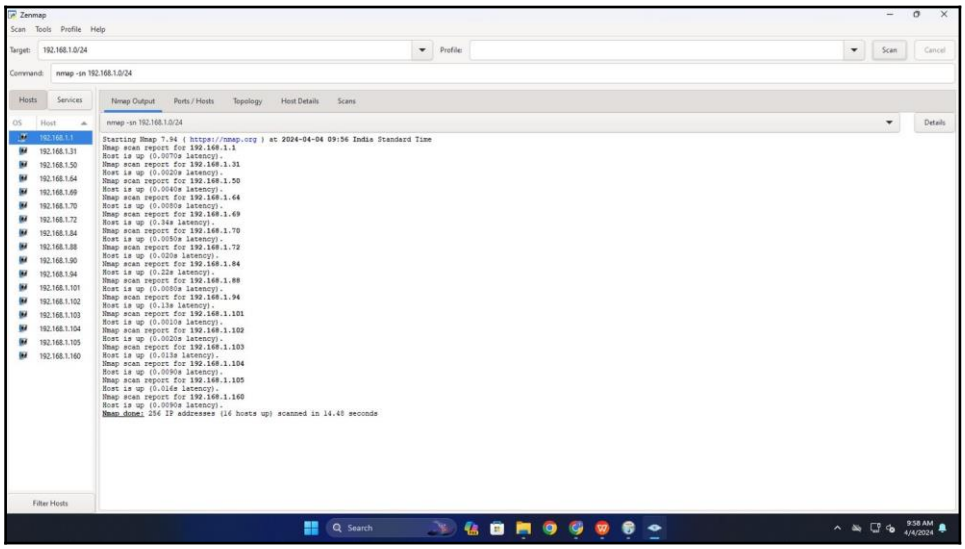
nmap -sn 192.168.1.0/24

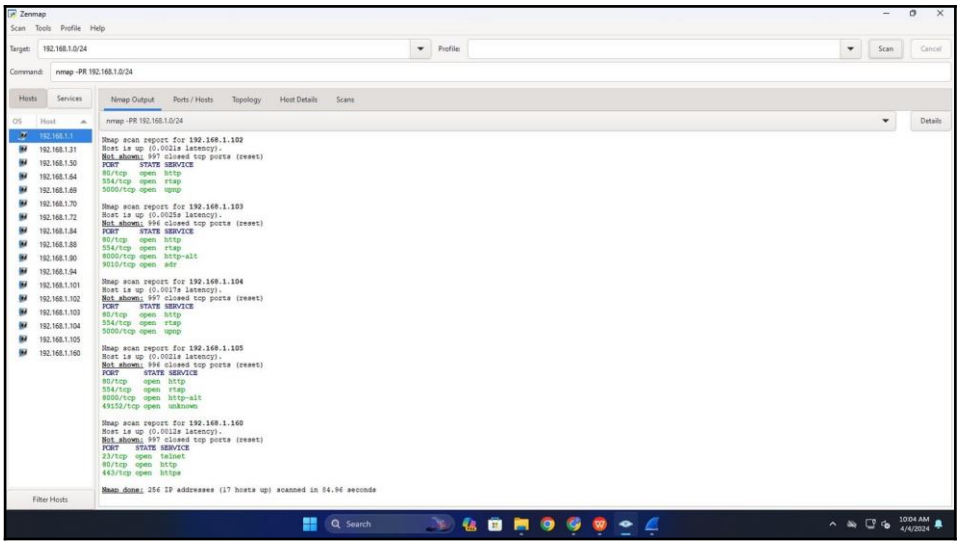
2. Check for Duplicate IPs:

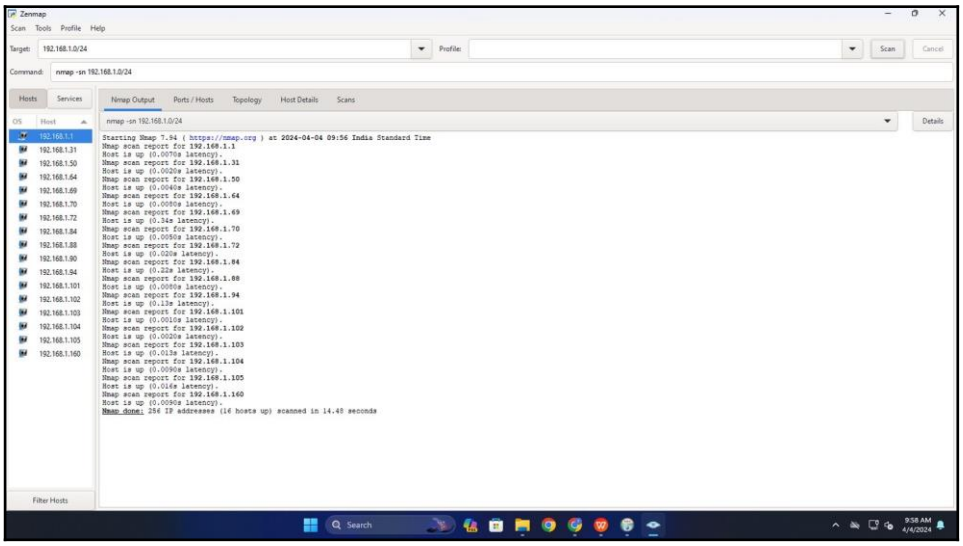
Look for multiple MAC addresses associated with a single IP, which could indicate an ARP spoofing attack.

nmap -PR 192.168.1.0/24 **Output:**









● **Using Wireshark**

`Wireshark` is a powerful network protocol analyzer that can be used to capture and analyze network traffic, making it suitable for detecting ARP spoofing attacks.

1. Capture Traffic:

Start capturing traffic on the network interface where you suspect ARP spoofing is happening. In Wireshark, select the appropriate network interface and start the capture.

2. Filter ARP Packets:

Use Wireshark's display filters to only show ARP packets.

arp

3. Analyze ARP Requests and Replies:

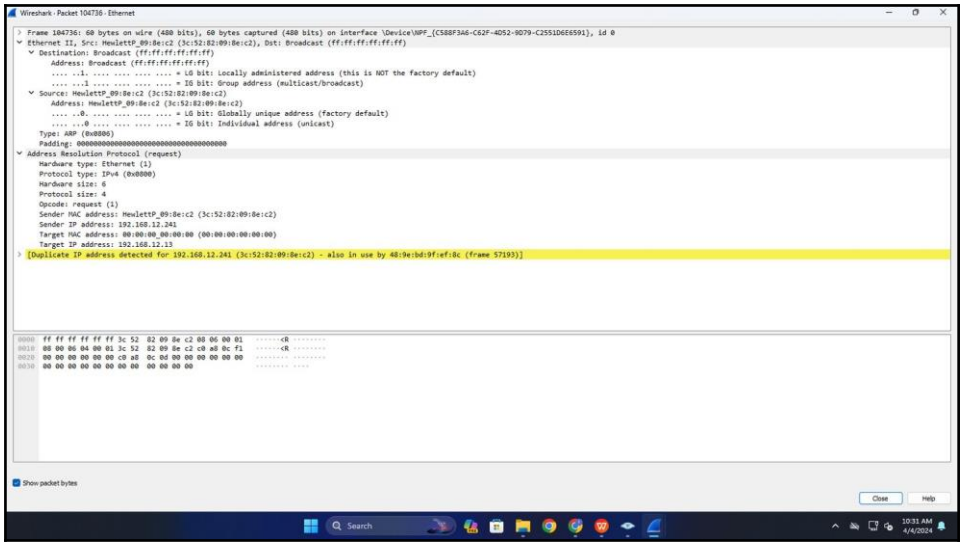
Look for inconsistencies in ARP requests and replies. In a typical ARP spoofing attack, you might see:

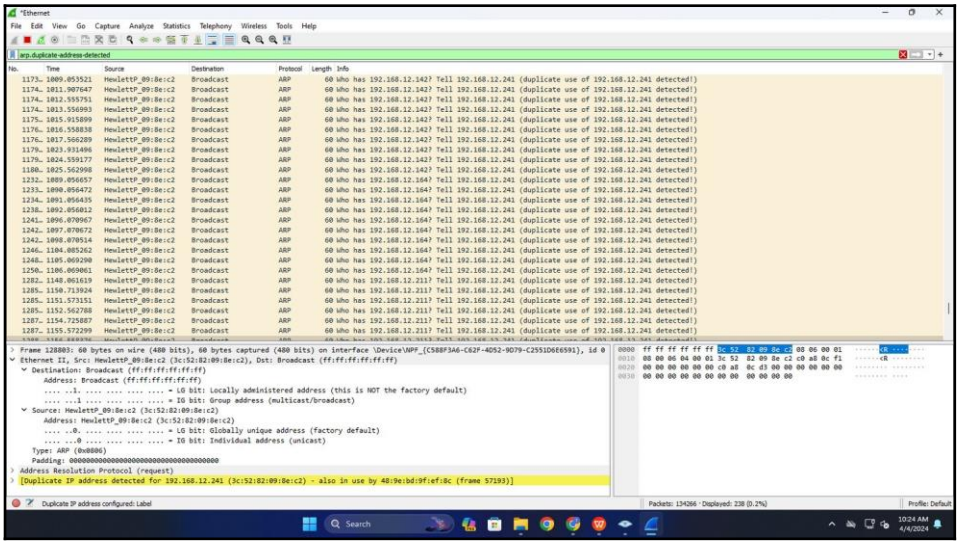
- ARP replies for ARP requests that were never sent.

- Multiple ARP replies for the same IP address with different MAC addresses.
- ARP requests and replies with suspicious or unexpected information.

Pay close attention to the source and destination MAC addresses, as well as the IP addresses involved in ARP requests and replies.

Output:





Conclusion:

In conclusion, ARP spoofing poses a significant security threat by allowing attackers to intercept network traffic. Utilizing tools like Nmap, ARPWATCH, and Wireshark can aid in detecting such attacks. Nmap provides a quick network scan to identify discrepancies in IP-MAC mappings, ARPWATCH continuously monitors ARP activity for any anomalies, and Wireshark allows for in-depth analysis of ARP packets for inconsistencies. By

combining these tools and techniques, network administrators can enhance their ability to identify and mitigate ARP spoofing attacks, bolstering overall network security.