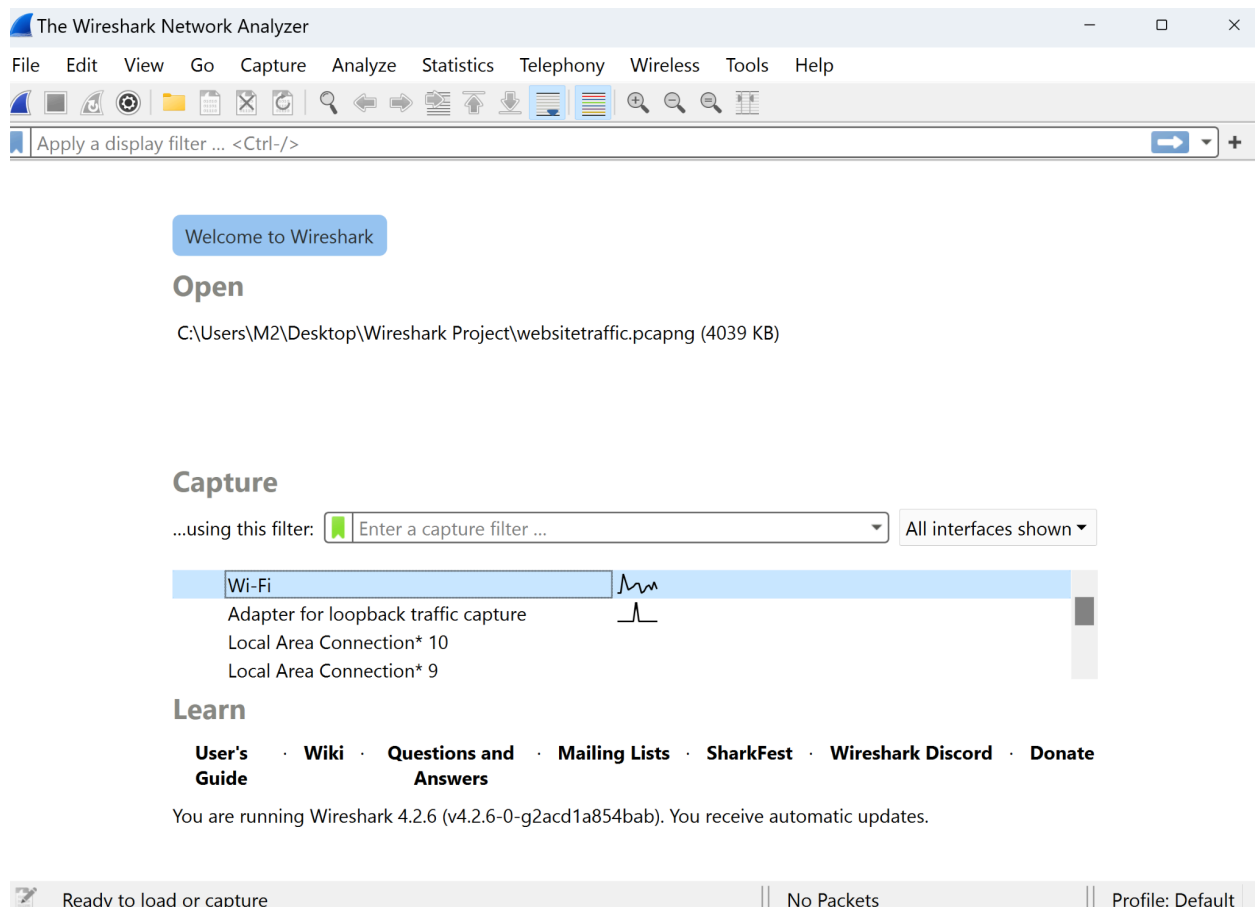


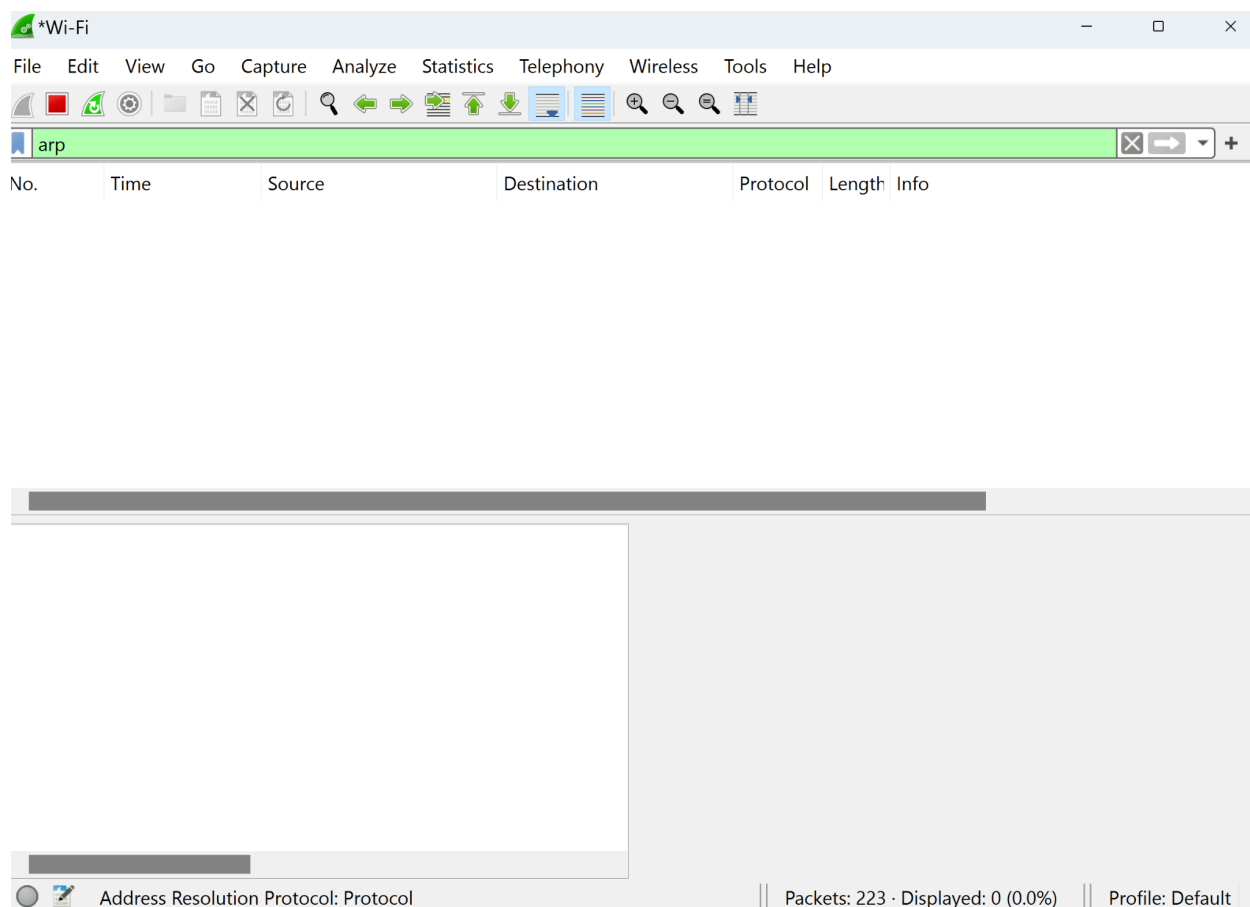
Using Nmap to Generate ARP Requests and Capturing Them with Wireshark

Step 1: Open Wireshark and Set Up for Capturing

1. **Launching Wireshark:** I opened Wireshark from my Start menu or desktop.
2. **Selecting a Network Interface:** Wireshark showed me a list of network interfaces, and I chose Wifi.



3. **Starting Capture:** I clicked the blue shark fin button to start capturing traffic on the selected interface.
4. **Filtering for ARP Traffic:** In the display filter bar at the top of Wireshark, I typed **arp** to filter only ARP traffic. This way, I could focus on ARP packets as they appeared.



Step 2: Use Nmap to Generate ARP Requests

1. **Opening Command Prompt:** I opened the Command Prompt on my Windows machine by pressing **Win + R**, typing **cmd**, and pressing **Enter**.



2. **Using Nmap for Scanning:** To generate ARP requests, I used Nmap to scan my local network. This scan sent ARP requests to identify live hosts on the network.

Basic ARP Scan: I ran the following command:

```
nmap -sn [my_network]/24
```

- I replaced `[my_network]` with my local network range.

Explanation:

- `nmap`: This is the command to invoke Nmap.
- `-sn`: This option tells Nmap to perform a "ping scan," which sends ARP requests without conducting a full port scan.
- `[my_network]/24`: This specifies the range of IP addresses on my local network. The `/24` subnet mask means I'm scanning all 256 addresses in a typical home network range.

3. **Running the Scan:** After typing the command, I pressed `Enter`, and Nmap started scanning the network, generating ARP requests in the process.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\M2> nmap -sn 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-13 15:03 Eastern Daylight Time
Nmap scan report for 192.168.1.1
Host is up (0.0075s latency).
MAC Address: 58:9F:8C:AD:1C:12 (TP-Link Technologies)
Nmap scan report for 192.168.1.2
Host is up (0.15s latency).
MAC Address: D8:9E:5C:12:1B:12 (Samsung Electronics)
Nmap scan report for 192.168.1.3
Host is up (0.051s latency).
MAC Address: 28:9E:5C:12:1B:12 (Amazon Technologies)
Nmap scan report for 192.168.1.4
Host is up (0.17s latency).
MAC Address: 28:9E:5C:12:1B:12 (Whisker Labs - Ting)
Nmap scan report for 192.168.1.5
Host is up (0.16s latency).
MAC Address: F0:8C:9E:5C:12:1B (Cloud Network Technology Singapore PTE.)
Nmap scan report for 192.168.1.6
Host is up (0.15s latency).
MAC Address: 08:9E:5C:12:1B:12 (Amazon Technologies)
Nmap scan report for 192.168.1.7
Host is up (0.15s latency).
MAC Address: 70:8C:9E:5C:12:1B (Intel Corporate)
Nmap scan report for 192.168.1.8
Host is up (0.15s latency).
MAC Address: D8:9E:5C:12:1B:12 (Shenzhen MTC)
Nmap scan report for 192.168.1.9
Host is up (0.18s latency).
```

Step 3: Capture ARP Requests in Wireshark

1. **Checking Wireshark for ARP Traffic:** I switched back to Wireshark and saw ARP requests appearing in real-time as a result of the Nmap scan.

*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
4672	146.270092	Microsoft_b4:26:53	ChinaDragonT_3d:a8:...	ARP	42	Who has 192.168.1.1? Tell me
4673	146.270124	ChinaDragonT_3d:a8:...	Microsoft_b4:26:53	ARP	42	192.168.1.1 is at 78:8a:86:3d:a8:d8
4675	146.388943	SamsungElect_21:0e:...	ChinaDragonT_3d:a8:...	ARP	42	Who has 192.168.1.1? Tell me
4676	146.388951	ChinaDragonT_3d:a8:...	SamsungElect_21:0e:...	ARP	42	192.168.1.1 is at 78:8a:86:3d:a8:d8
4677	146.563491	AmazonTechno_ce:39:...	ChinaDragonT_3d:a8:...	ARP	42	Who has 192.168.1.1? Tell me
4678	146.563511	ChinaDragonT_3d:a8:...	AmazonTechno_ce:39:...	ARP	42	192.168.1.1 is at 78:8a:86:3d:a8:d8
4683	146.795130	SamsungElect_21:0e:...	Broadcast	ARP	42	Who has 192.168.1.1? Tell me
4753	151.298009	SamsungElect_21:0e:...	Broadcast	ARP	42	Who has 192.168.1.1? Tell me
4789	152.321352	SamsungElect_21:0e:...	Broadcast	ARP	42	Who has 192.168.1.1? Tell me
4803	153.345247	SamsungElect_21:0e:...	Broadcast	ARP	42	Who has 192.168.1.1? Tell me

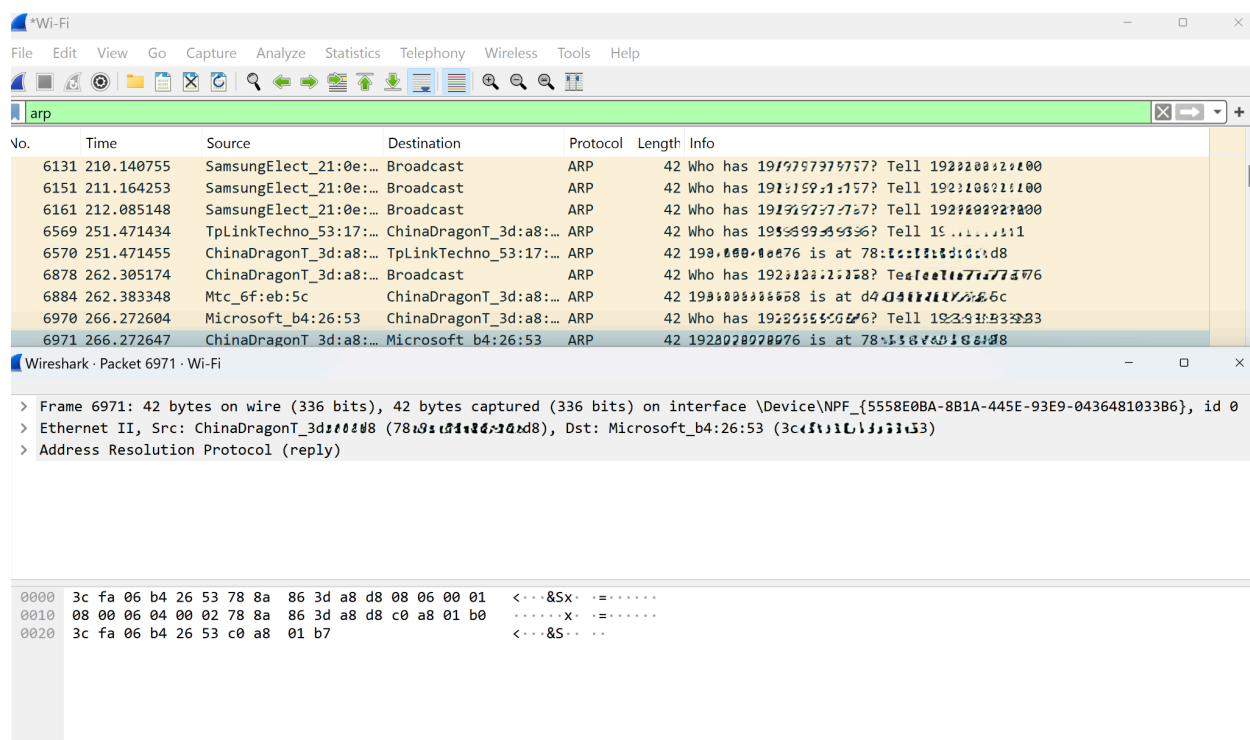
> Frame 486: 42 bytes on wire (336 bits), 42 bytes captured on interface (336 bits) on Wi-Fi
 > Ethernet II, Src: SamsungElect_21:0e:a8 (d0:c2:4e:21:0e:a8), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Address Resolution Protocol (request)

0000 78 8a 86 3d a8 d8 d0 c2 4e 21 0e a8 08 06 00 01
 0010 08 00 06 04 00 01 d0 c2 4e 21 0e a8 c0 a8 01 64
 0020 00 00 00 00 00 00 c0 a8 01 b0

wireshark Wi-FiYG2JS2.pcapng | Packets: 11483 · Displayed: 621 (5.4%) · Dropped: 0 (0.0%) | Profile: Default

2. Inspecting the ARP Packets:

- I clicked on an ARP packet in the list to highlight it.
- The packet details pane showed me the ARP request and reply information.
- I could see who was asking for which IP address and the MAC address involved.



Step 4: Stop the Capture and Save the Data

1. **Stopping Capture:** Once I had captured enough ARP requests, I clicked the red square button to stop the capture.
2. **Saving the Capture:**
 - o I went to **File > Save As** and chose a location and filename to save my capture file.
 - o I saved the file with a **.pcapng** extension.

Step 5: Analyzing the Capture

Conclusion

By following these steps, I successfully used Nmap to generate ARP requests and captured them using Wireshark. This exercise demonstrates my understanding of basic network scanning and packet analysis techniques.