**Name: Amarachi Anthony**

**Course: CIDM 6340 -70 NTWK MGT & INFO SEC**

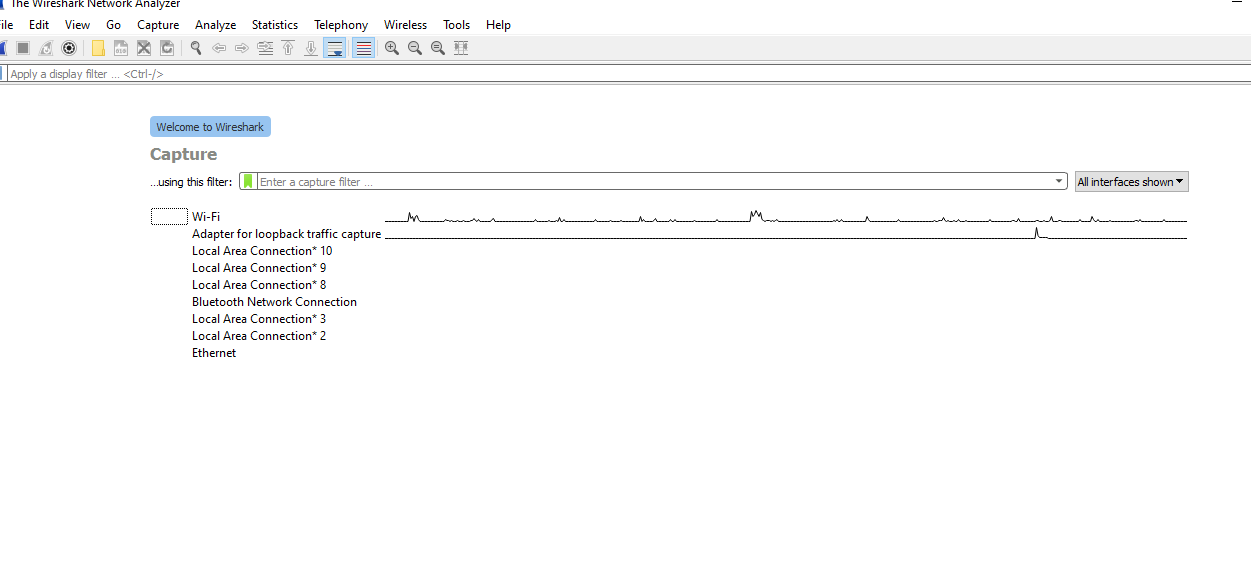
**Assignment: Research Report 2 Wireshark**

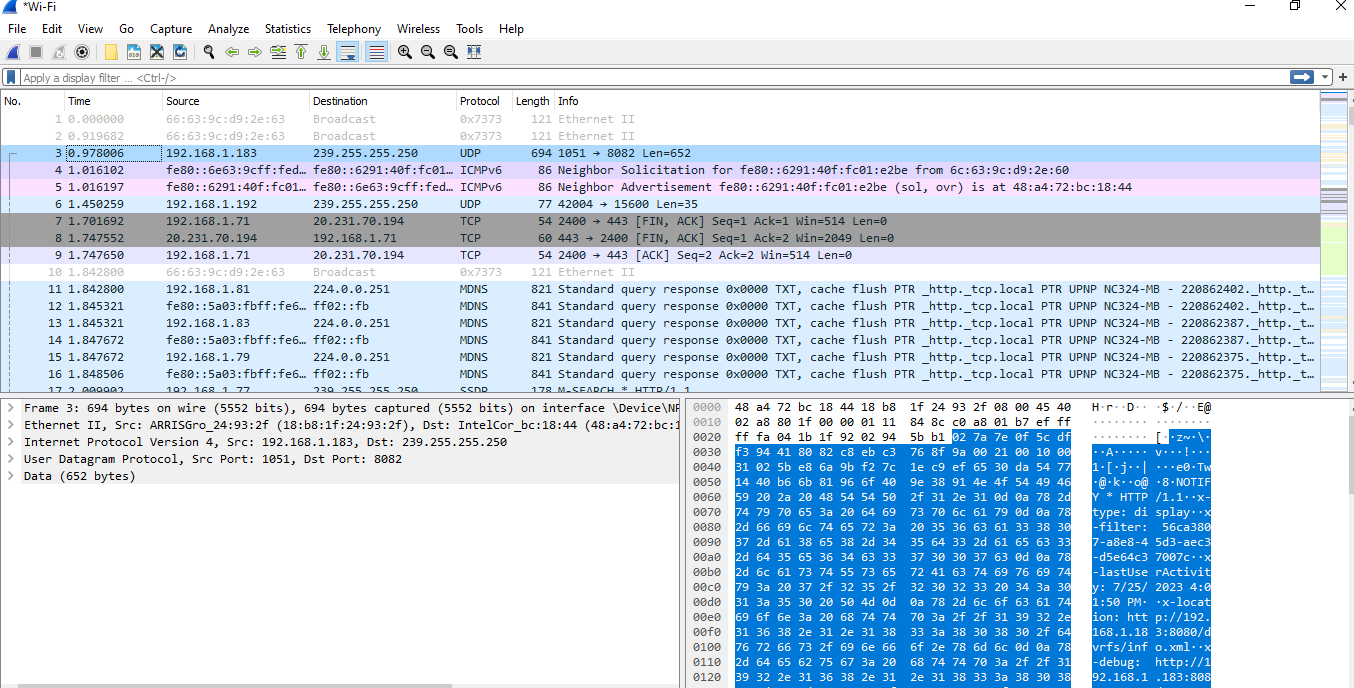
**What did you do?**

I started by installing Wireshark on my laptop and launched the application. Wireshark utilizes a packet capture library to intercept and record the data transmitted through the network interface cards (NICs) on its host device. By default, Wireshark focuses on capturing data within its device, but when operating in promiscuous mode, it can capture nearly all data circulating within its local area network (LAN). Wireshark employs the NMAP's Packet Capture library, known as npcap, to facilitate its packet-capturing capabilities.

To begin my Wireshark analysis my home network was used. I first selected the network interface I wanted to capture packets from. The goal was to capture packets sent or received through the selected interface. , Wireshark began capturing packets in real-time as they flowed through the interface. To focus on specific packets of interest and reduce the data displayed while capturing packets, I applied filters based on criteria like IP addresses, protocols, ports, and packet length. I allowed the network to capture up to 5 minutes and then I saved the captured data (packet capture) to a file. After capturing packets, Wireshark automatically decoded them based on their respective network protocols. It could understand and analyze protocols such as TCP, UDP, HTTP, DNS, and more, translating the binary data into a human-readable format. This decoding allowed me to view the contents of each packet and comprehend the communication between different devices on the network.

**What was the result?**

After installing and launching Wireshark, this window appeared on the screen.

The packet capture log shows various network activities, including TCP, UDP, and TLSv1.2 protocols. These activities involve establishing and terminating TCP connections, exchanging data, and communicating through UDP. TCP ensures reliable connections between two devices using a three-way handshake, maintaining correct data transmission order through sequence and acknowledgment numbers. To detect errors, TCP uses the "Checksum" field to verify the transmitted data's integrity.

Packets are used to exchange data between devices. Here are some analysis of different types of packets shown:

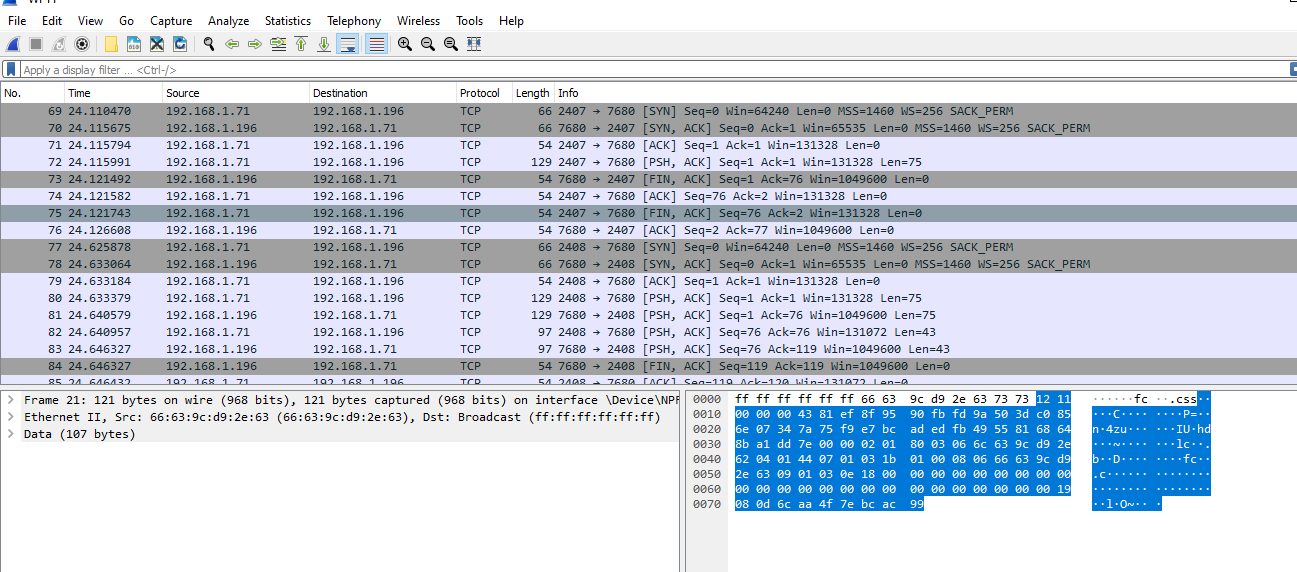
 Packet 1 and Packet 2 are Ethernet broadcast packets that are sent to all devices on the local network. They contain the same payload, represented by the hexadecimal value 0x7373, and are both 121 bytes long.

 Packet 3 uses the UDP protocol to communicate. It originates from IP address 192.168.1.183 and is sent to the multicast IP address 239.255.255.250. The source port is 1051, and the destination port is 8082. This packet's payload is 652 bytes long.

 Packet 4 and Packet 5 involve the ICMPv6 protocol and are used to inquire about the MAC address of a neighbor with a known IPv6 address and respond with the neighbor's link-layer address (MAC address).

Packet 6 is another UDP communication but with a smaller payload length of 35 bytes. It is from IP address 192.168.1.192 to multicast IP address 239.255.255.250, with source and destination ports specified as 42004 and 15600, respectively.

 Packet 7 to Packet 9 are part of a TCP communication between IP address 192.168.1.71 and a remote IP address 20.231.70.194. These packets suggest that the TCP connection is being terminated, as they include the FIN and ACK flags. TCP ensures reliable, ordered, and error-checked delivery of data.



The TCP Handshake Initiation took place between Packets 69-71, where the host with IP address 192.168.1.71 (port 2407) sent an SYN packet to host 192.168.1.196 (port 7680) to initiate a TCP connection. The packet contained several key components, including the SYN flag, sequence number (seq) of 0-window size (Win) of 64240, Maximum Segment Size (MSS) of 1460, Window Scale (WS) of 256, and Selective Acknowledgment Permitted (SACK\_PERM).

 In response to the SYN packet, Host 192.168.1.196 (port 7680) sent a SYN-ACK packet to 192.168.1.71 with the SYN and ACK flags set, an initial sequence number of 0, an Acknowledgment number (Ack) of 1, a Window size of 65535, Maximum Segment Size of 1460, and Selective Acknowledgment Permitted (SACK\_PERM).

 Host 192.168.1.71 then acknowledged the SYN-ACK from host 192.168.1.196 with an ACK packet containing a sequence number of 1, an acknowledgment number of 1, and a Window size of 131328.

 After the TCP Handshake process was complete, Host 192.168.1.71 sent a data packet to host 192.168.1.196 with the PSH and ACK flags set, a sequence number of 1, an acknowledgment number of 1, a Window size of 131328, and a payload length of 75 bytes.

 The Connection Termination Initiation (First Attempt) took place between Packets 73-75, where Host 192.168.1.196 sent a FIN-ACK packet to host 192.168.1.71 to start the connection termination process. The packet contained the FIN and ACK flags, a sequence number of 1, and an acknowledgment number of 76. However, the other host did not immediately acknowledge the termination process.

 Following the termination process, Host 192.168.1.71 acknowledged the receipt of the FIN-ACK packet from the previous attempt with an ACK packet containing a sequence number of 2 and an acknowledgment number of 77.

 A New TCP communication was initiated between Packets 77-81, with a handshake from host 192.168.1.71 (port 2408) to host 192.168.1.196 (port 7680). The SYN, SYN-ACK, and ACK packets were exchanged, similar to the previous handshake.

 Data was transmitted between the two hosts in both directions between Packets 82-83.

 Finally, Host 192.168.1.196 initiated a connection termination with a FIN-ACK packet, and Host 192.168.1.71 acknowledged the process.

The packet capture log shows various network activities, including TCP, UDP, and TLSv1.2 protocols. These activities involve establishing and terminating TCP connections, exchanging data, and communicating through UDP. TCP ensures reliable connections between two devices using a three-way handshake, maintaining correct data transmission order through sequence and acknowledgment numbers. To detect errors, TCP uses the "Checksum" field to verify the transmitted data's integrity.

Overall, the Wireshark results mainly show normal network communications and standard protocol activities within the local network. There are no immediate signs of malicious intent or exploitable vulnerabilities. However, it's essential to maintain good network hygiene, keep systems up-to-date with security patches, and monitor network traffic continuously to detect any potential security threats or suspicious activities that may emerge over time. Regular security audits and risk assessments are also crucial to maintaining a secure attack surface.

**What did you learn?**

After the analysis, I learned a lot; Wireshark is an invaluable tool for conducting comprehensive network analysis, as it can detect and capture network traffic. However, obtaining proper authorization before utilizing Wireshark within an organization is imperative, as this ensures compliance with security policies and regulations. One of the critical benefits of Wireshark is its ability to identify both wired and wireless networks, enabling the user to understand network activity completely.

Even capturing packets for a limited time can yield significant insights into network performance and potential security issues. Analyzing these captured packets can provide valuable information on network protocols, traffic patterns, and other vital metrics. Additionally, analyzing open networks (within ethical boundaries) can help users better understand different protocols and network behaviors.

Wireshark's detailed packet analysis helps identify potential network issues, optimize network performance and troubleshoot problems. However, adhering to legal and ethical guide is crucial when using Wireshark or any other network analysis tool. Unauthorized packet capture or analyzing networks without proper permission can severely violate privacy and potentially lead to legal consequences. Therefore, it is essential for users always to exercise caution, discretion, and sound judgment when working with Wireshark.