Experiment – 7

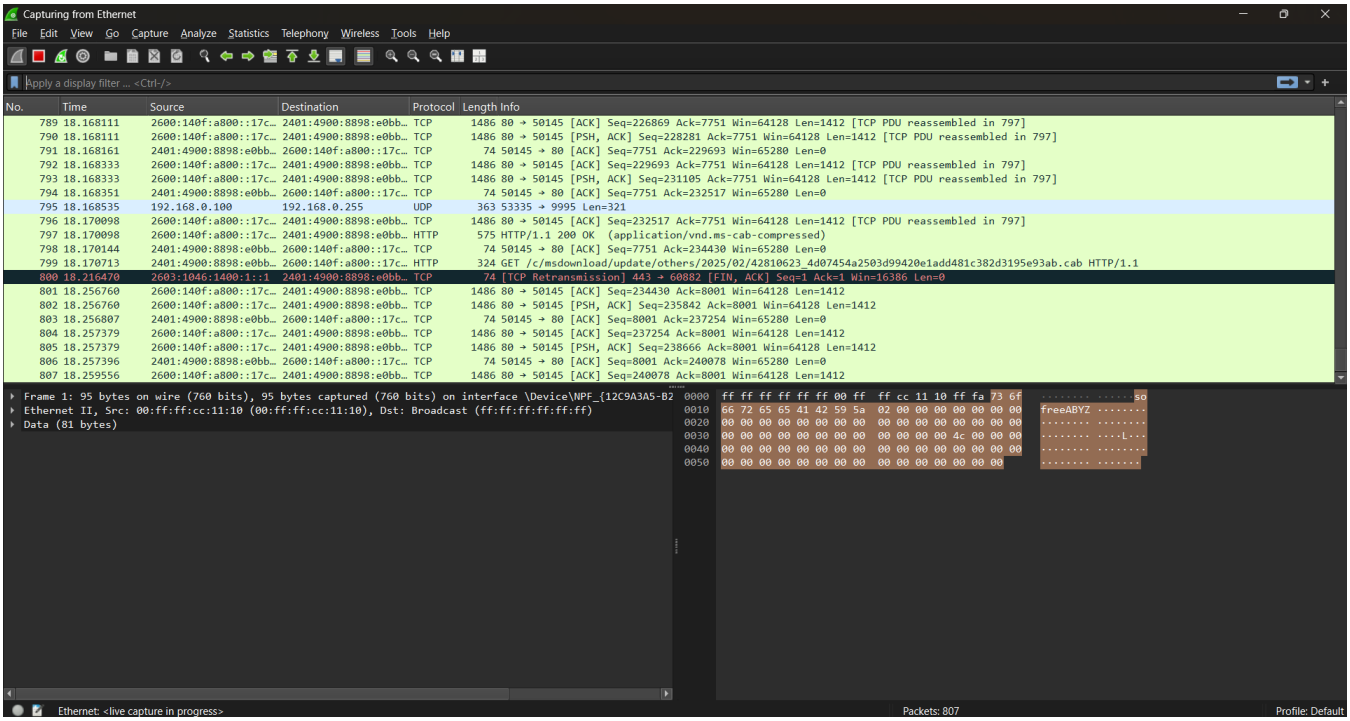
Aim: To capture network traffic on your machine and analyze packets to understand how data travels over a network.

Objective: Capture and inspect network packets to understand the protocols in use and identify potential issues in the traffic.

Theory:
Wireshark captures packets transmitted over a network and allows you to inspect them. It provides detailed information about each packet, including source and destination addresses, protocol types, and data payloads. Understanding this data is essential for network troubleshooting, security analysis, and protocol development.

Used **Commands** in Wireshark:

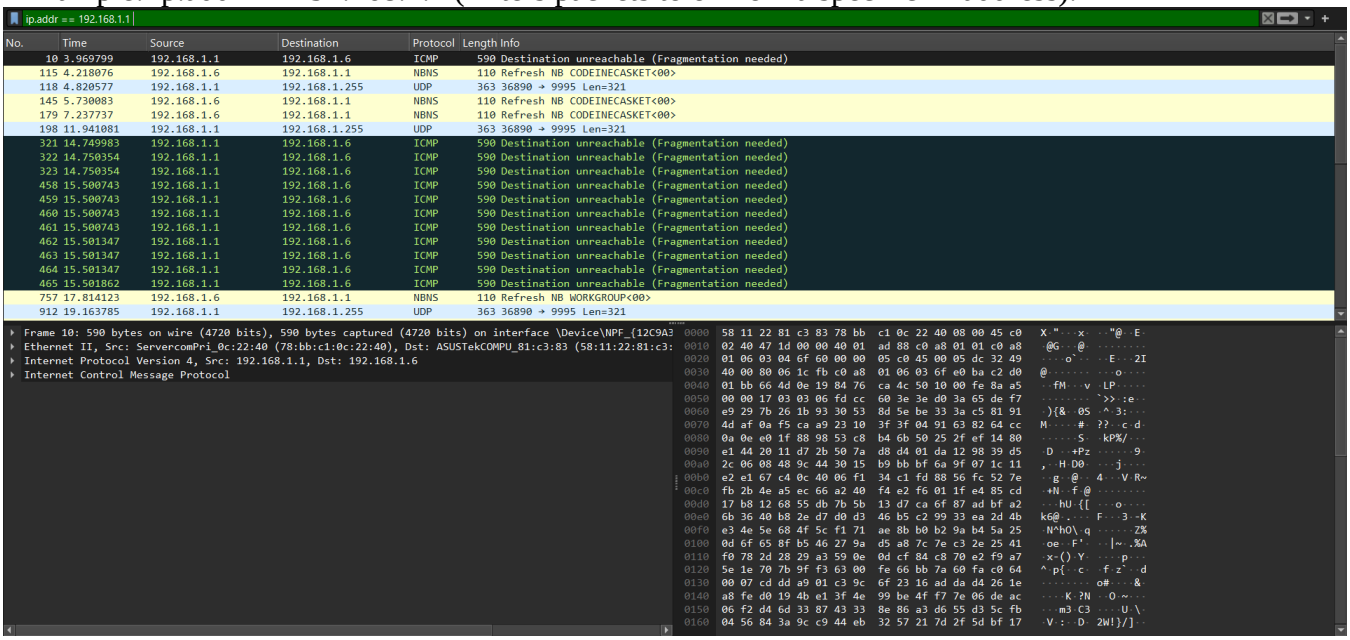
1. Start Capture:
- Go to Capture > Start or use the shortcut Ctrl + E to begin capturing packets.



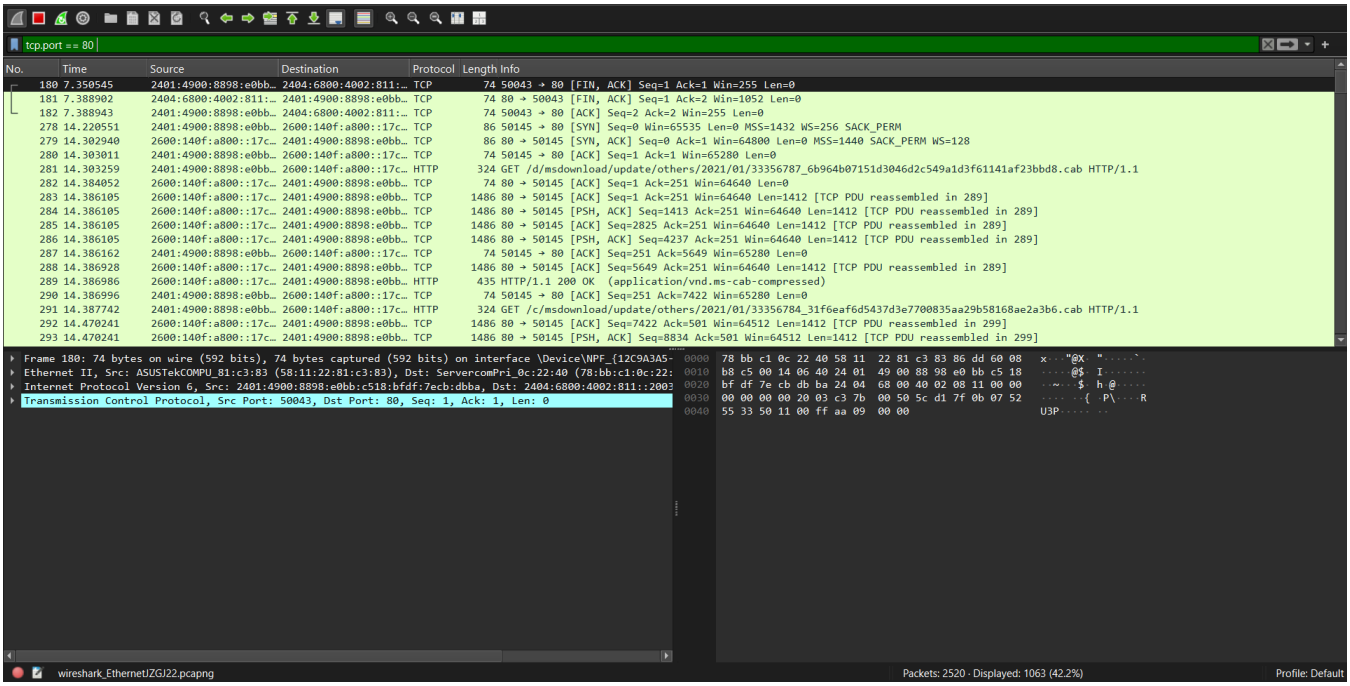
2. Display Filters (to filter the traffic):

- Use display filters to narrow down the captured traffic based on criteria such as IP address, protocol, or port number.

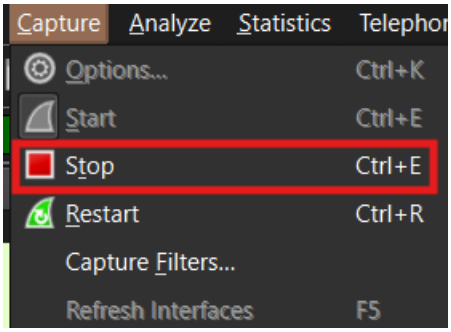
- Example: ip.addr == 192.168.1.1 (Filters packets to or from a specific IP address).



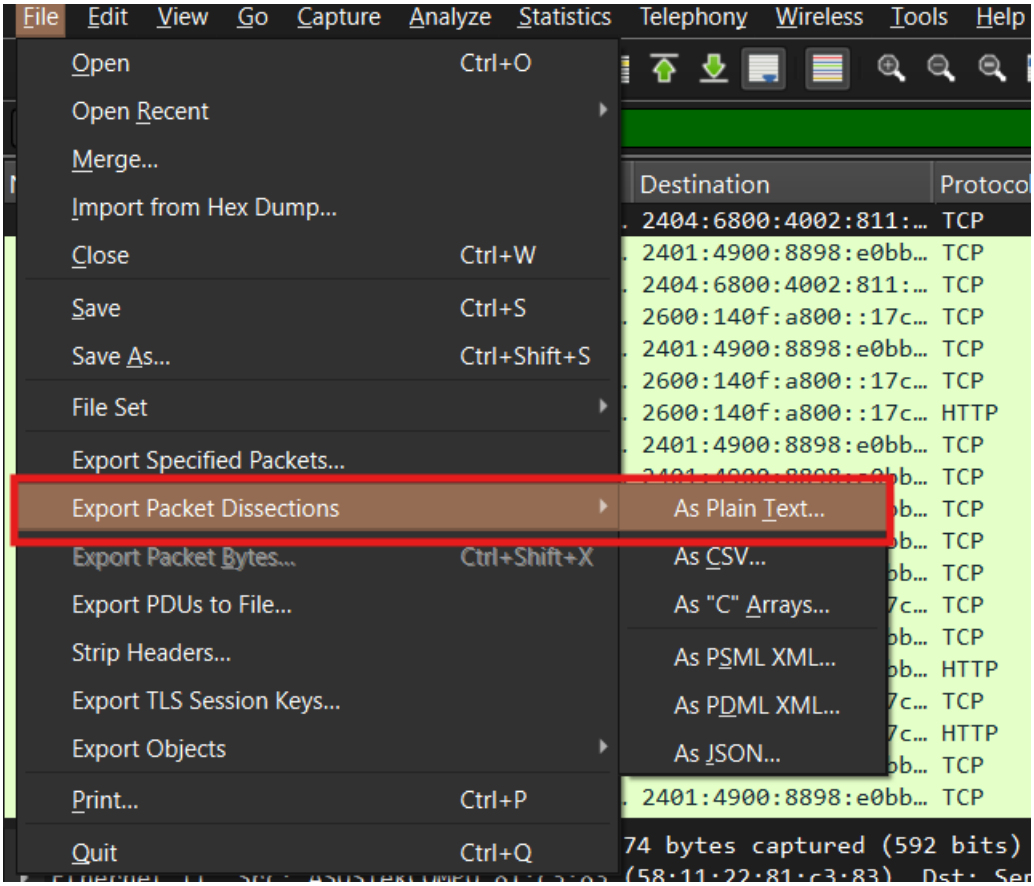
- Example: tcp.port == 80 (Filters TCP packets on port 80, which is HTTP).



3. Stop Capture:
- Go to Capture > Stop or press Ctrl + E to stop the capture.



4. Export Packet Capture:
- Go to File > Export Packet Dissections > As Plain Text to save captured packets for later analysis.



Conclusion: By capturing and analyzing network packets with Wireshark, we gain insights into data transmission, protocols, and potential network issues. Filtering traffic helps focus on specific IPs, protocols, or ports for detailed inspection. Understanding captured packets is crucial for troubleshooting, security monitoring, and protocol analysis. Wireshark serves as a powerful tool for network analysis and diagnostics.