Packet Analysis with Wireshark on a Personal Network

1. Installation of Wireshark

- **Action:** Installed Wireshark on the system.
- **Details:** Downloaded the software from the official website: https://www.wireshark.org/download.html. Selected the version compatible with the operating system (e.g., Windows 11, macOS, or Linux) and followed the installation wizard. Launched Wireshark to confirm successful installation.

2. Initiation of Packet Capture

- **Action:** Began capturing packets on the active network interface.
- **Details:** Opened Wireshark and selected the active interface (e.g., Wi-Fi or Ethernet) from the list of available options. Clicked "Start" with administrative privileges to initiate real-time packet capture.

3. Generation of Network Traffic

- **Action:** Generated network traffic by browsing a website or pinging a server.
- **Details:** Opened a web browser and visited https://www.example.com, or used the command line to execute ping google.com. Performed these actions for approximately 30-60 seconds to create sample traffic for analysis.

4. Termination of Capture

- **Action:** Stopped the packet capture after one minute.
- **Details:** Returned to Wireshark and clicked the "Stop" button (red square) after capturing traffic for about one minute. Verified that a sufficient number of packets were recorded in the interface.

5. Application of Protocol Filters

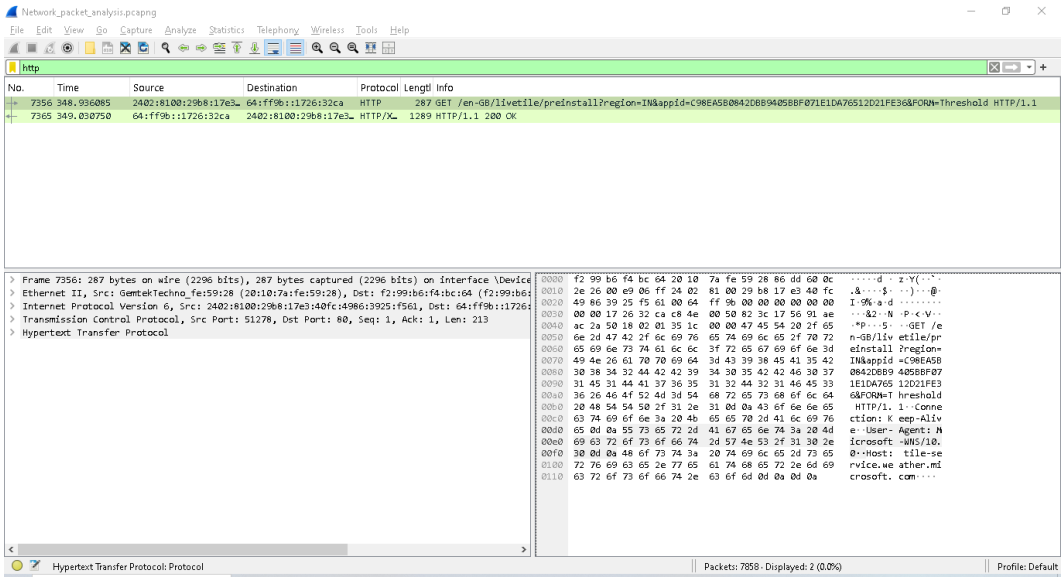
- **Action:** Applied filters to isolate packets by protocol.
- **Details:** Utilized the Wireshark filter bar to apply specific protocol filters: typed http to view HTTP traffic, dns for DNS queries, and tcp for TCP packets. Pressed Enter after each filter to display relevant packets.

6. Identification of Protocols

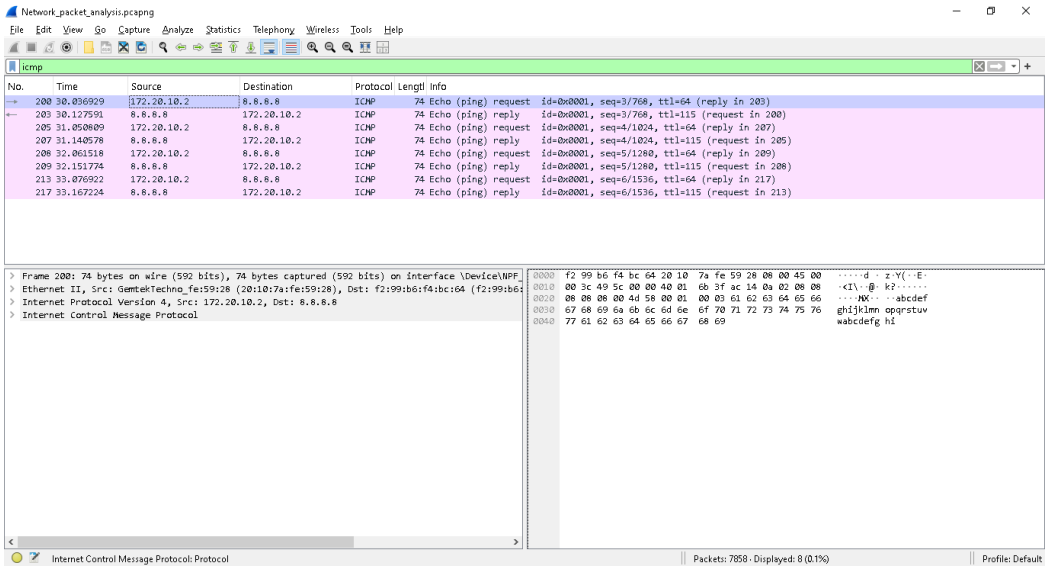
- **Action:** Identified at least three different protocols in the captured traffic.
- **Details:** Analyzed the filtered packets and used the "Protocol Hierarchy" option (right-click a packet > Statistics > Protocol Hierarchy) to confirm protocols. Identified examples include:
    - **HTTP**: Traffic on port 80 related to web browsing.
    - **DNS**: Queries on port 53 for domain name resolution.
    - **TCP**: Handshake packets on various ports for reliable data transfer.

Packet Analysis with Wireshark on a Personal Network

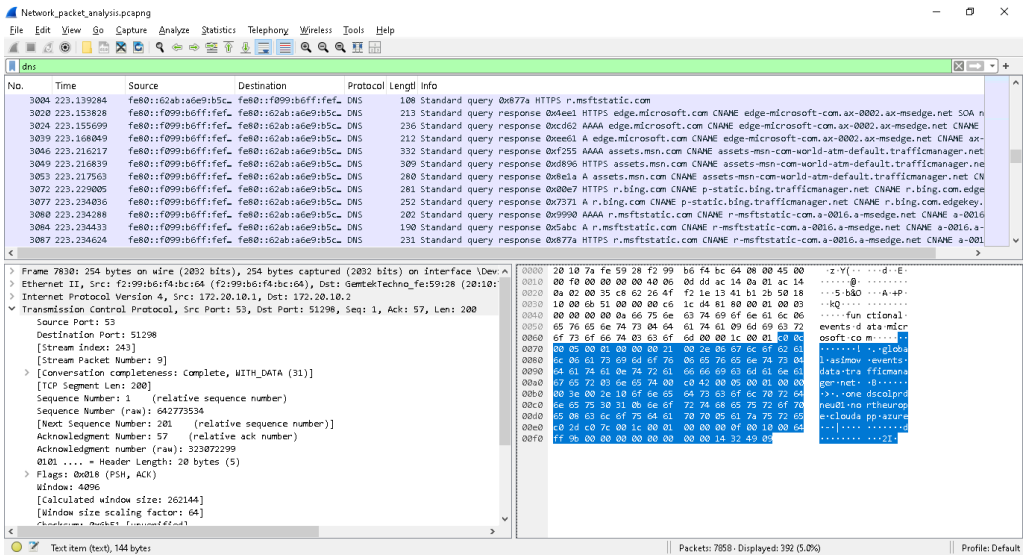
HTTP Packets:



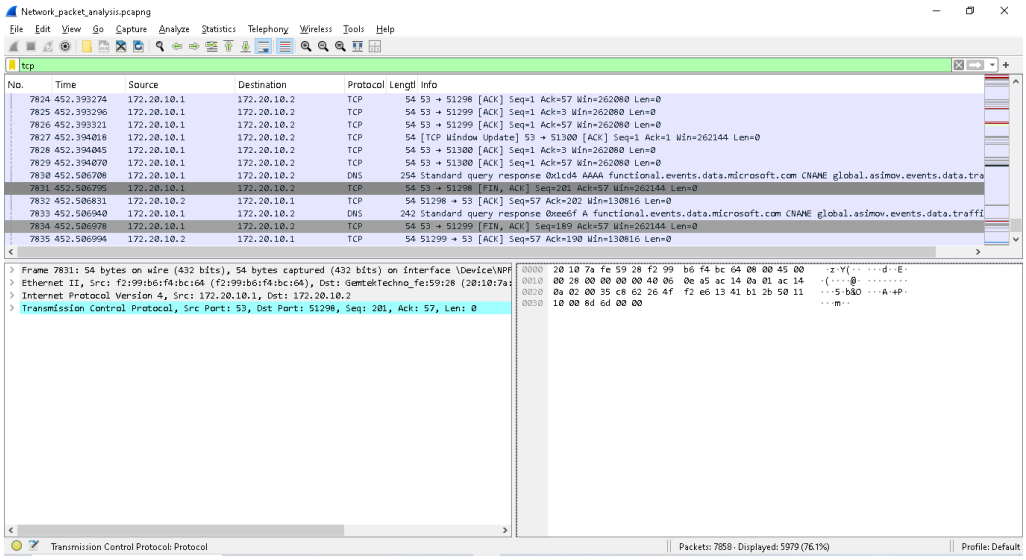
ICMP Packets:



DNS Packets:



DNS Packets:



7. Export of Capture File

- **Action:** Exported the captured traffic as a .pcap file.
- **Details:** Navigated to File > Save As in Wireshark, chose a save location, named the file (e.g., network_capture_20250630.pcap), and saved it in .pcap format for future reference or analysis.

8. Summary of Findings

- **Action:** Summarized the analysis of captured packets and their details.
- **Details:** Reviewed the capture, which included approximately 500 packets over one minute. Key observations:
    - **HTTP**: Approximately 100 packets showed GET requests to example.com, indicating webpage loading.

- **DNS**: Around 20 packets resolved example.com to an IP address (e.g., 93.184.216.34).
- **TCP**: About 300 packets included handshake sequences (SYN, SYN-ACK, ACK) on ports like 80 or 443.
- No anomalies (e.g., excessive traffic or unexpected ports) were noted in this controlled test.