Task 5 : Capture and Analyze Network Traffic Using Wireshark.

Name: Rahul Malatesh Sannapujar

Date: 30/09/2025

Objective

Capture live network packets and identify basic protocols and traffic types.

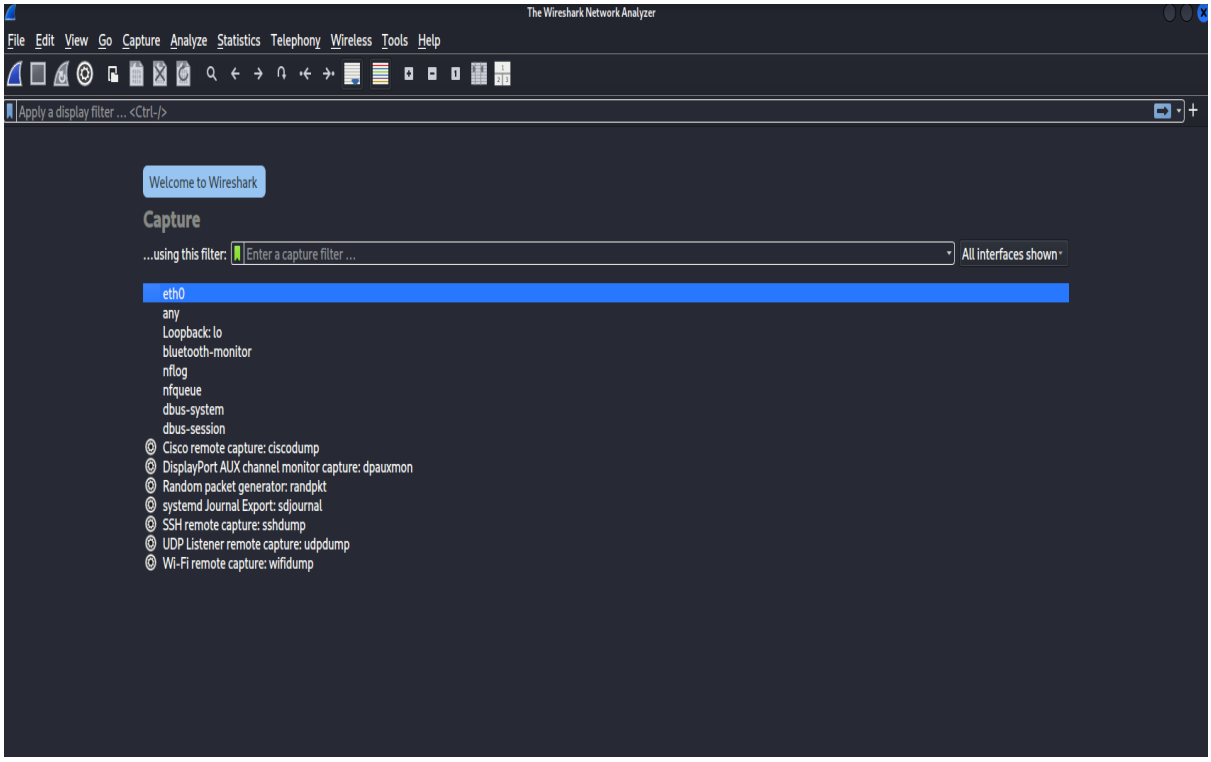
Tools Used

Wireshark

Search Optimization

Screenshots for validation

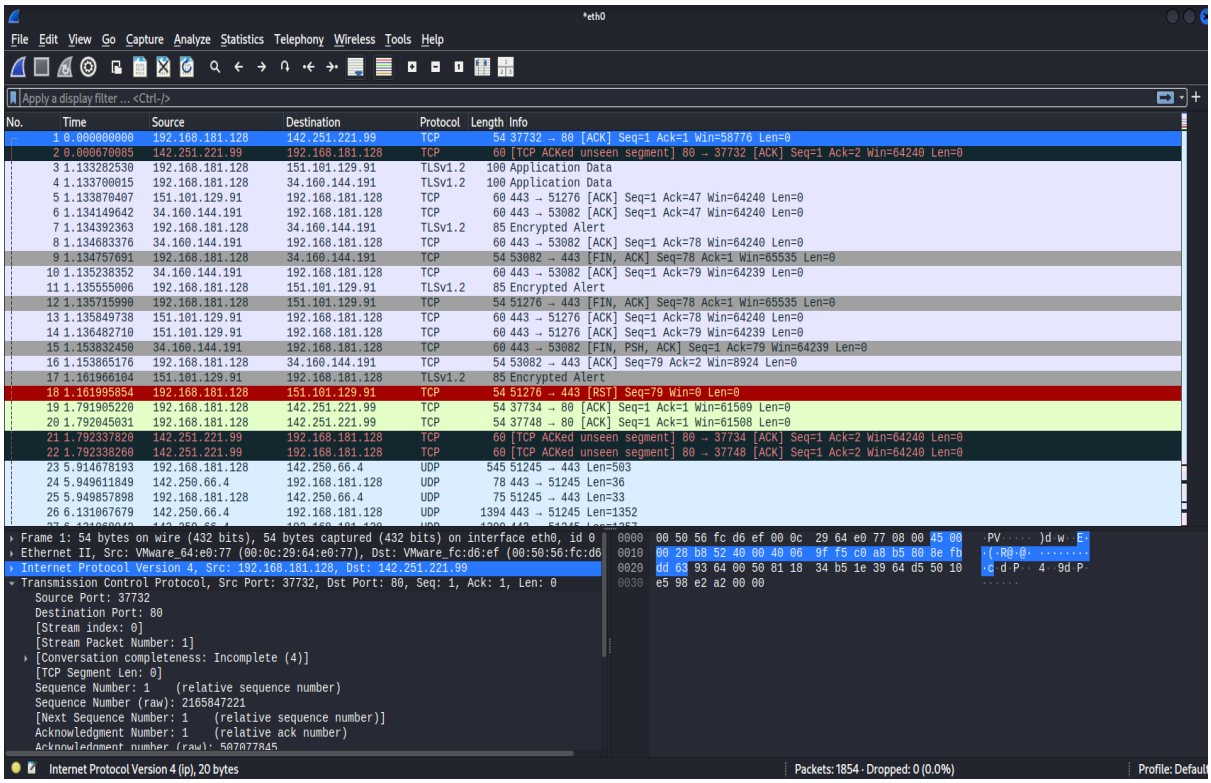
1. Image of Wireshark Interface Selection



This image shows the initial launch screen of **The Wireshark Network Analyzer**. The user is selecting the active network interface for packet capture, with **eth0** highlighted. This screen confirms the tool used and the first step of the task: choosing the interface to "Capture live network packets".

Task 5 : Capture and Analyze Network Traffic Using Wireshark.

2. Image of TCP and TLS Traffic (Initial View)

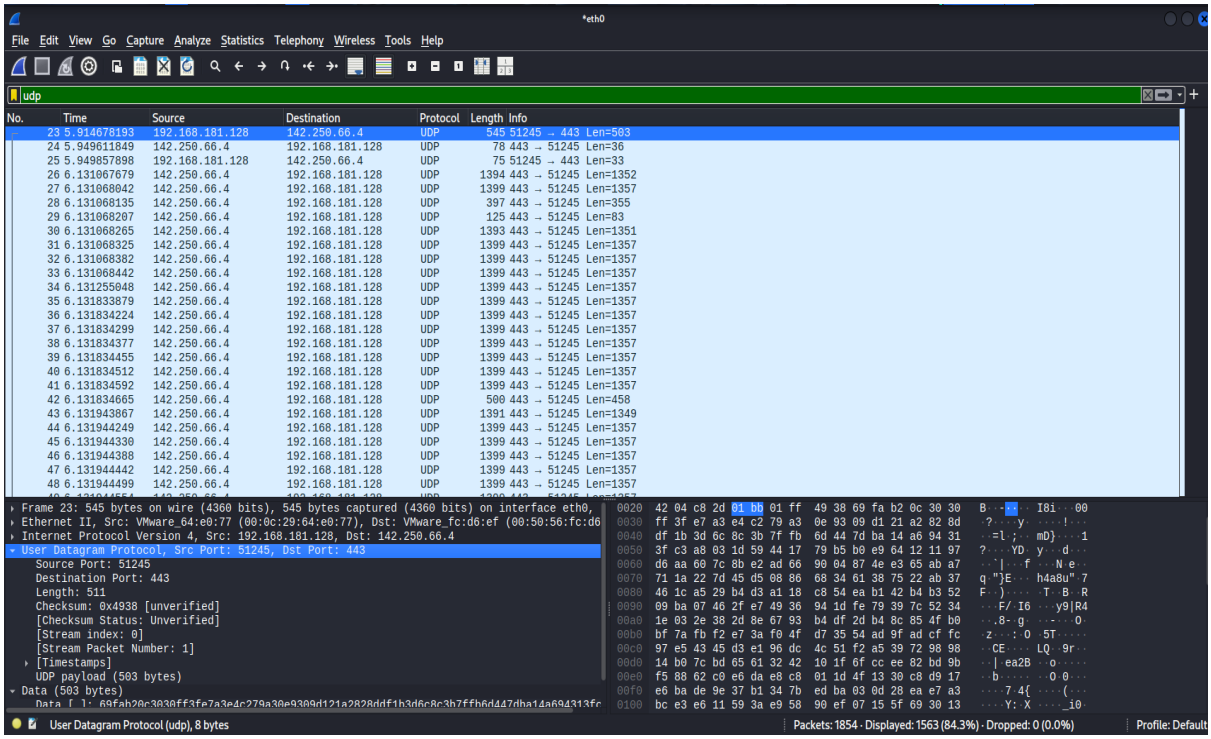


This screenshot displays the main Wireshark window after capturing traffic, specifically showing a mix of **TCP** and **TLSv1.2** packets.

- The top pane lists packets showing the source and destination IP addresses (e.g., 192.168.181.128 and 142.251.221.99) and the **Protocol**.
- The flow includes a standard **TCP handshake** and subsequent **Application Data** and **Encrypted Alert** messages related to secure, encrypted traffic.
- The packet details pane below shows the layered structure, with the **Transmission Control Protocol** highlighted, confirming its role as the transport layer.

Task 5 : Capture and Analyze Network Traffic Using Wireshark.

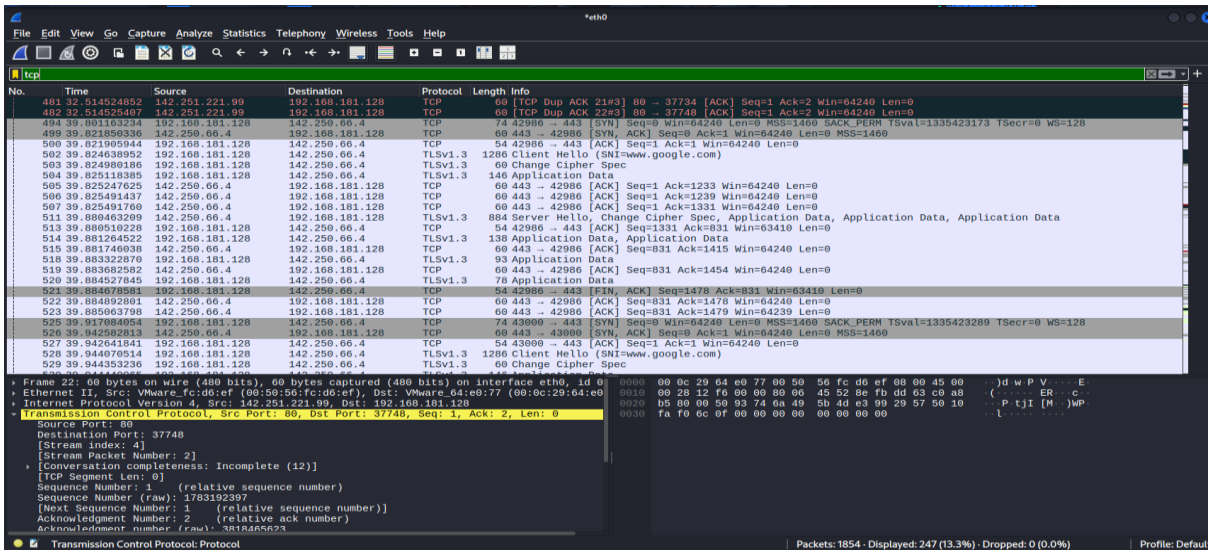
3. Image of UDP Filtered Traffic



This image shows the packet list filtered exclusively for the **udp** protocol.

- The list displays numerous **UDP** packets with varying lengths and flows between local and remote IP addresses.
- The packet details pane (bottom-left) highlights the **User Datagram Protocol** layer, indicating a source port of 51245 and a destination port of 511.
- The protocol's raw datagram nature is confirmed by the lack of complex sequence data in the display.

4. Image of Detailed TCP and TLS Handshakes

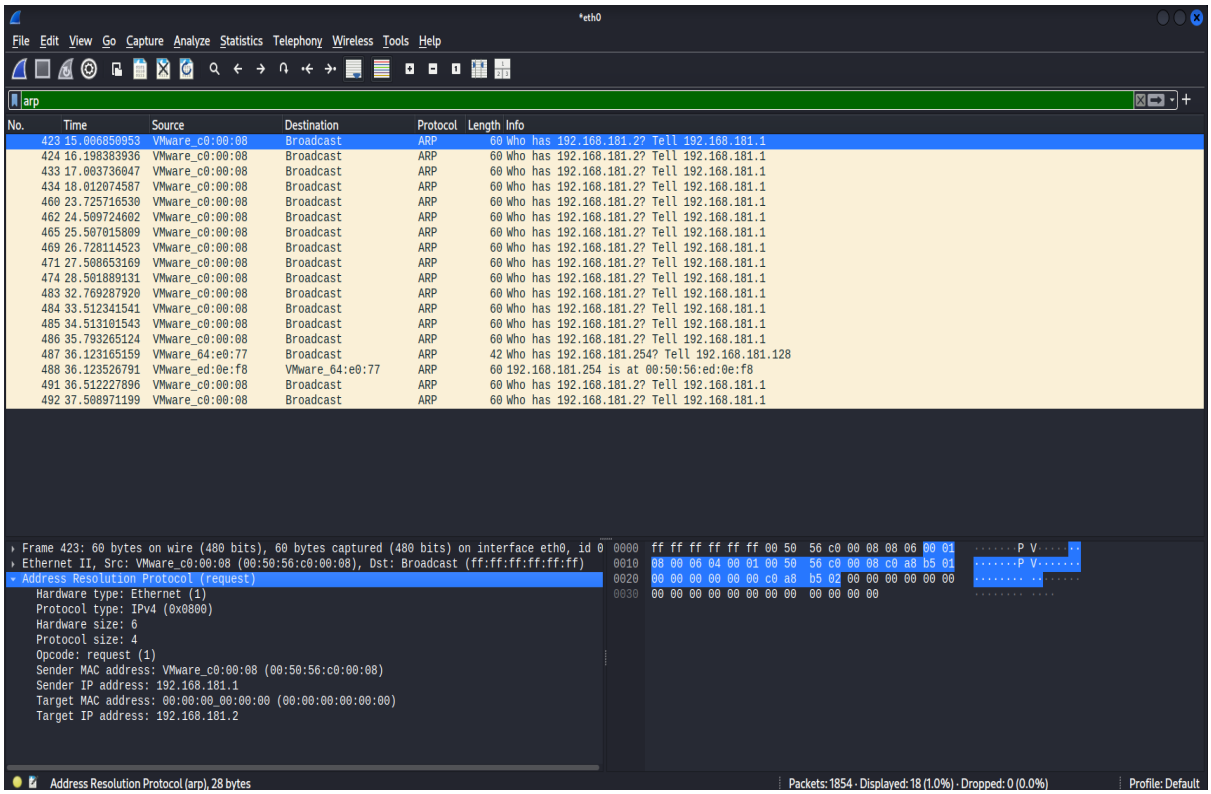


Task 5 : Capture and Analyze Network Traffic Using Wireshark.

This is a focused view on **TCP** traffic, clearly showing the **TCP three-way handshake** and **TLS handshake** components.

- The capture highlights a TCP packet that is part of an establishment sequence (Ack: 2, Len: 0).
- Crucially, this view shows the **TLSv1.3 Client Hello** message initiating a secure connection to a Google-owned domain ("https://www.google.com/search?q=i.xn--google.com"), confirming the successful use of an application-layer protocol.
- It also shows the conclusion of a session using the FIN (Finish) flag, demonstrating proper session termination.

5. Image of ARP Filtered Traffic

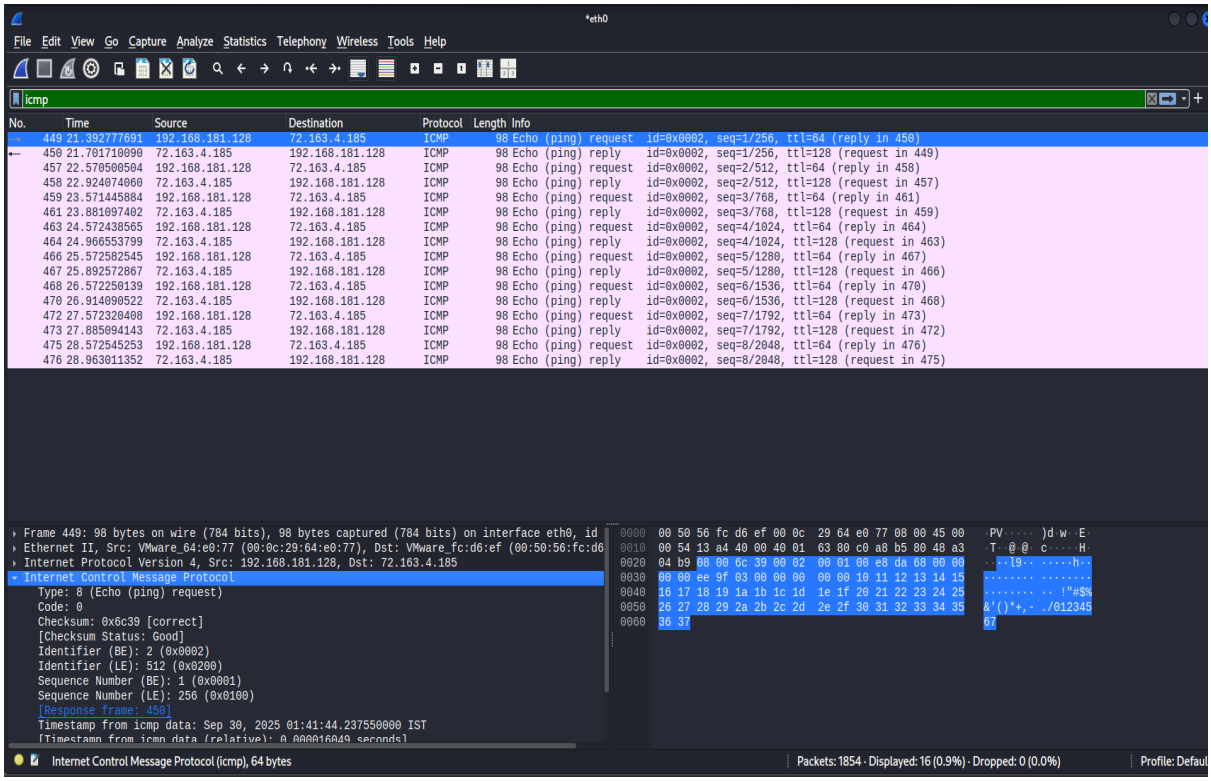


This image shows the packet list filtered for the **arp** protocol, which operates at the link layer.

- The packets are all **ARP who has** requests (broadcasts), where a device asks for the MAC address of a specific IP address (e.g., "Who has 192.168.181.2? Tell 192.168.181.1").
- The packet details pane confirms the **Address Resolution Protocol** as a request, using the Opcode request (1). This traffic is fundamental for local network addressing.

Task 5 : Capture and Analyze Network Traffic Using Wireshark.

6. Image of ICMP Filtered Traffic

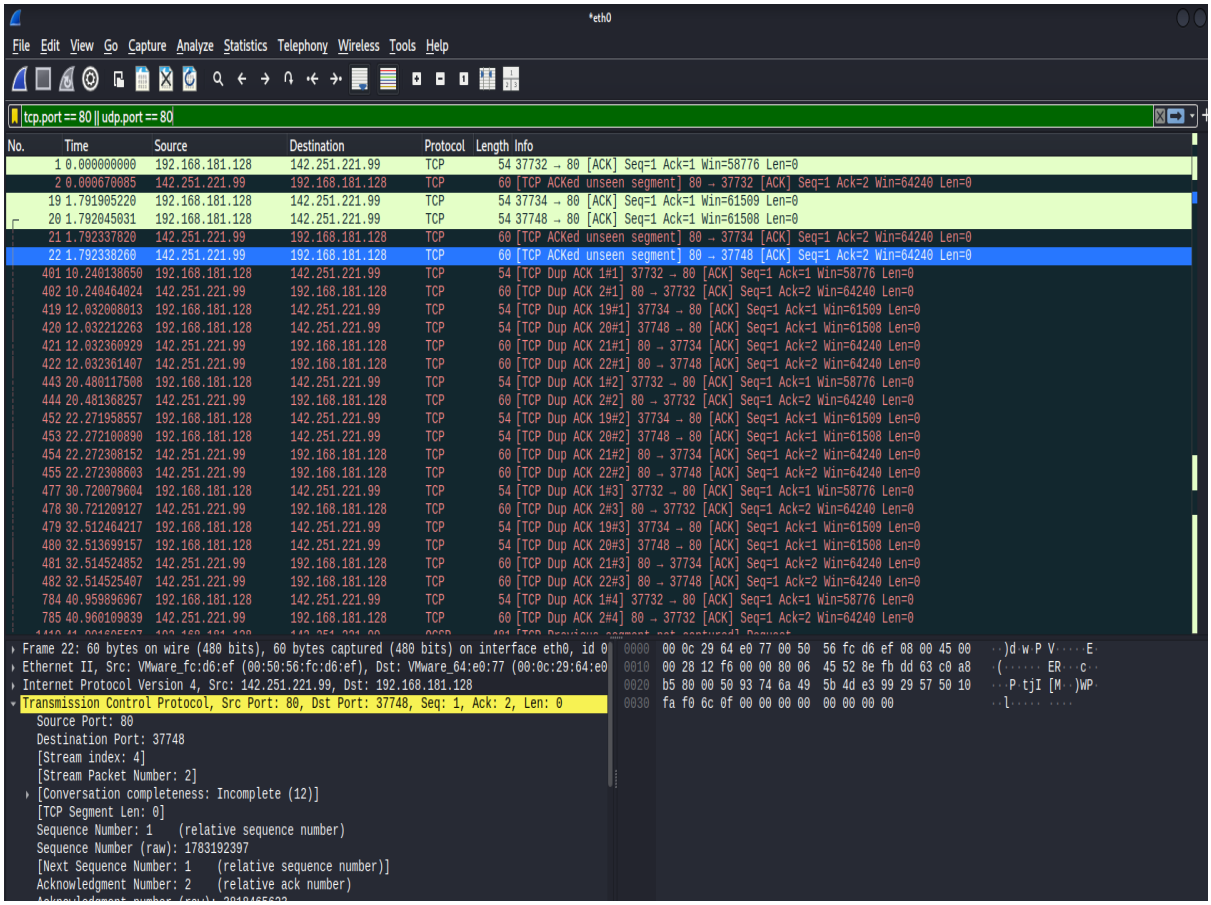


This image is filtered to show the **icmp** protocol, used for diagnostics.

- The packet list exclusively contains pairs of **ICMP Echo (ping) request** and **ICMP Echo (ping) reply** messages.
- This confirms a network connectivity test (a ping) was executed, with the source communicating with the destination IP address 72.163.4.185.
- The details pane highlights the **Internet Control Message Protocol** layer, confirming the packet type as an Echo (ping) request.

Task 5 : Capture and Analyze Network Traffic Using Wireshark.

7. Image of TCP/UDP Port 80 Filtered Traffic



This image shows a filtered view for traffic targeting **Port 80** (tcp.port==80 or udp.port==80), the standard port for unencrypted HTTP traffic.

- The list consists mainly of **TCP** packets related to web communication, including **TCP Dup ACK** (Duplicate Acknowledgement) packets, which can indicate packet loss or reordering on the network.
- This filter was likely used to isolate attempts at standard, unencrypted web browsing.