🧠 Wireshark Network Traffic Analysis

📋 Project Overview

This project demonstrates basic network traffic analysis using Wireshark. The goal was to capture, filter and analyze packets generated during normal web activity to identify key network protocols and understand their roles in communication.

⚙️ Steps Performed

1.Installed Wireshark on Kali Linux.

2.Started packet capture on the active network interface.

3.Generated traffic by:

•Browsing the website: https://elevatelabs.in

•Pinging the same domain using the terminal (ping elevatelabs.in)

4.Captured packets for about one minute.

5.Filtered packets in Wireshark by:

•http — for web traffic

•dns — for domain resolution

•tcp — for transport-level communication

6.Identified at least three protocols in the captured data:

•DNS – Used for domain name resolution.

•TCP – Ensured reliable transport of data.

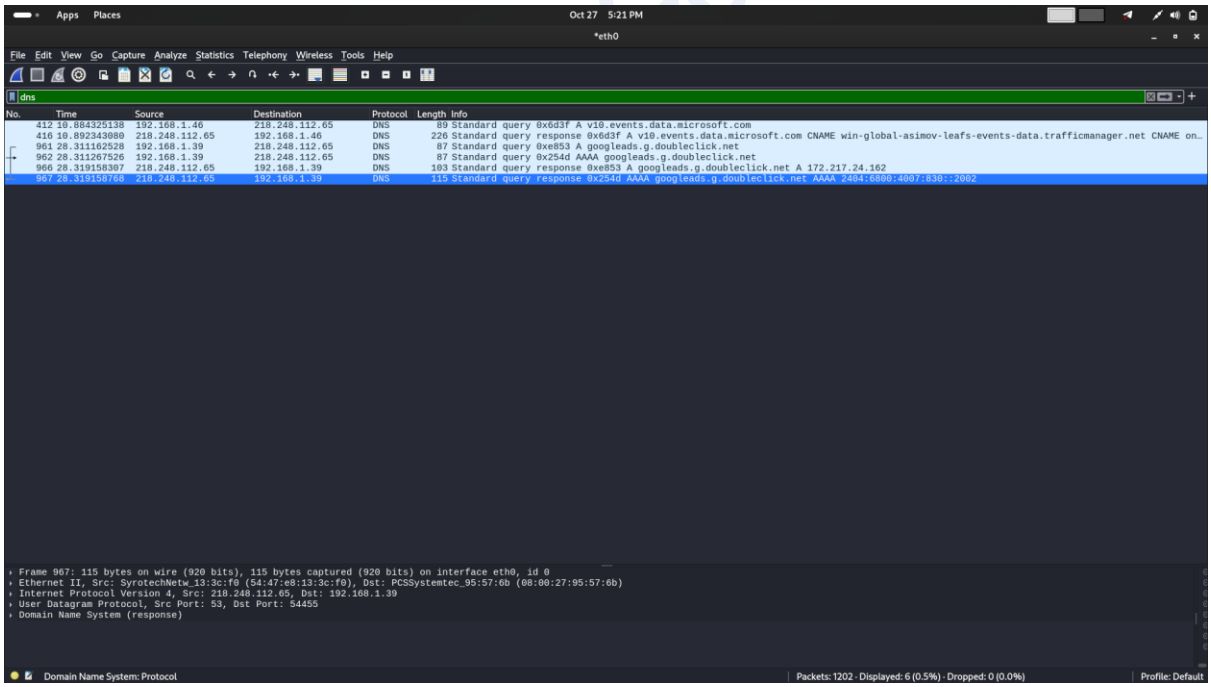
•HTTP – Managed web content exchange between client and server.

7.Exported the capture as a .pcap file for documentation and analysis.

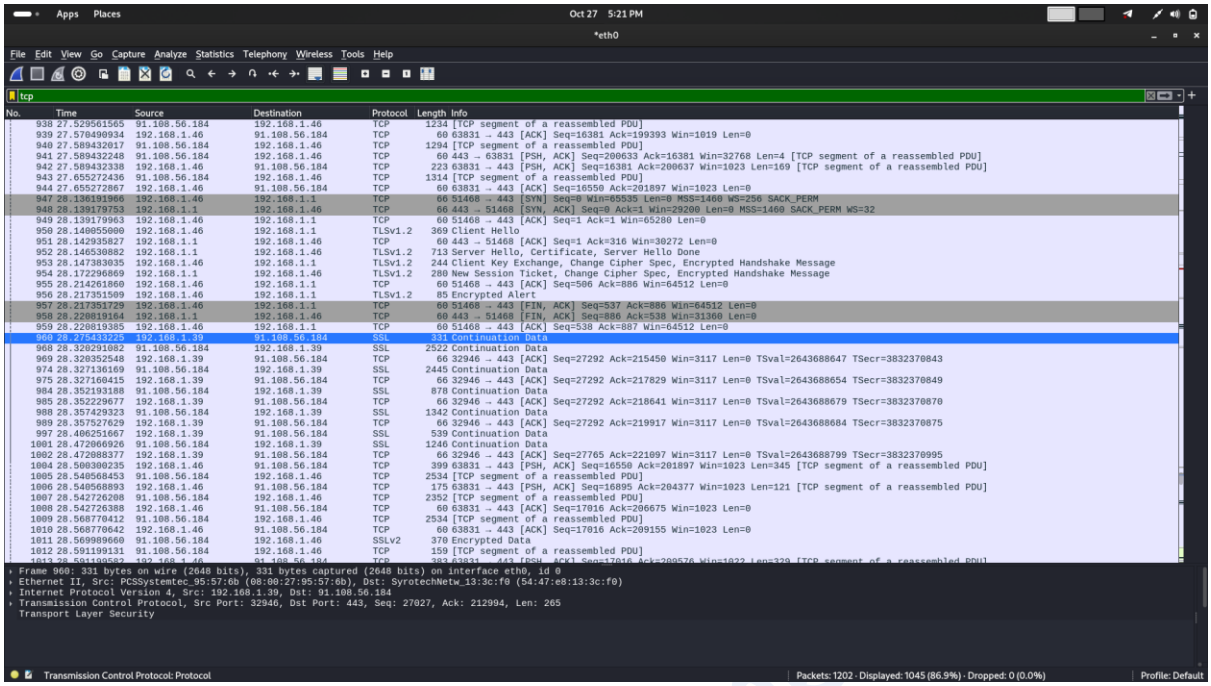
🔍 Findings Summary

Protocol	Description	Example Observation
DNS	Converts domain names (like elevatelabs.in) to IP addresses	Multiple DNS queries and responses observed.
TCP	Establishes reliable connections between client and server	3-way handshake and ACK packets visible.
HTTP	Transfers web page data between browser and website	GET and 200 OK responses noted during browsing.

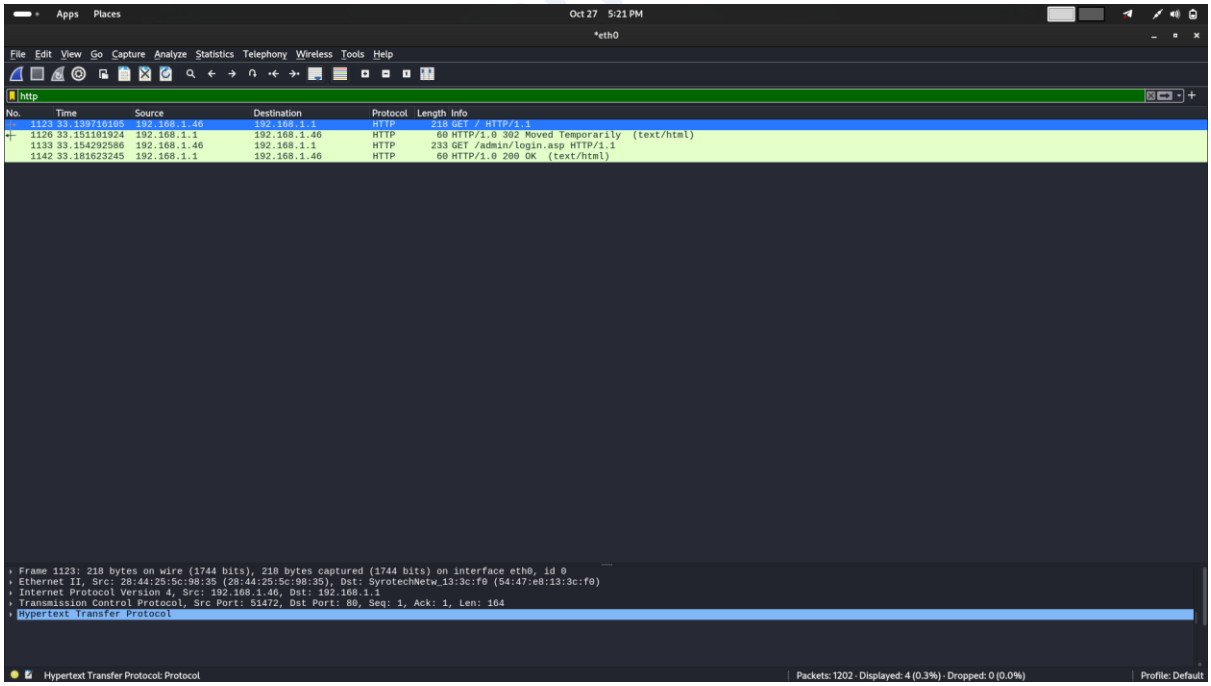
DNS packets are filtered



TCP packets are filtered



HTTP packets are filtered



📁 Files Included

•traffic_capture.pcap → Raw packet capture file.

•README.md → Project documentation.

🧩 Tools Used

• Wireshark – For capturing and analyzing network packets.

• Ping Command – To generate ICMP traffic.

• Web Browser – To create HTTP and DNS requests.

• CentralOps.net – Used to find the IP address of the target domain (elevatelabs.in).

🧠 Key Learnings

• Gained understanding of how different protocols (DNS, TCP, HTTP) interact during normal web communication.

• Learned how to capture, filter and interpret packets in Wireshark.

• Observed packet structures and relationships between layers in the OSI model.

📜 Conclusion

This exercise provided hands-on experience in network traffic analysis and protocol inspection using Wireshark.

It helped visualize real-time data exchange and understand how requests, responses and handshakes occur in network communication.