Cybersecurity Internship Report — Task 5

Intern Name: Sai Aditya
Organization: Elevate Labs
Task Title: Capture and Analyze Network Traffic Using Wireshark
Date: August 11, 2025

Objective

To capture live network packets using Wireshark, identify at least three different protocols, and summarize their key characteristics and relevance in network communication.

Tools Used

- Operating System: Kali Linux
- Packet Analyzer: Wireshark
- Network Interface: eth0 (wired) / wlan0 (wireless)
- Target IP: 192.168.150.133

Scenario

A network capture was performed on the active network interface to monitor and analyze real-time traffic. During the capture session, common activities such as website browsing, DNS lookups, and ICMP pings were carried out to generate traffic. The capture lasted approximately 60 seconds and was then filtered to examine specific protocols.

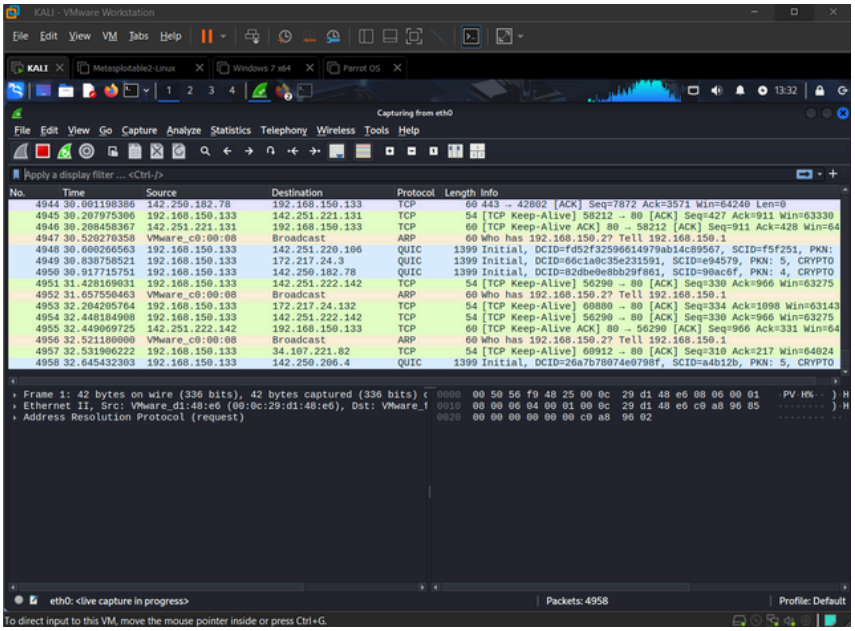
Disclaimer

This activity was conducted in a controlled environment solely for educational purposes. No unauthorized traffic interception or monitoring of third-party networks was performed.

Steps Performed

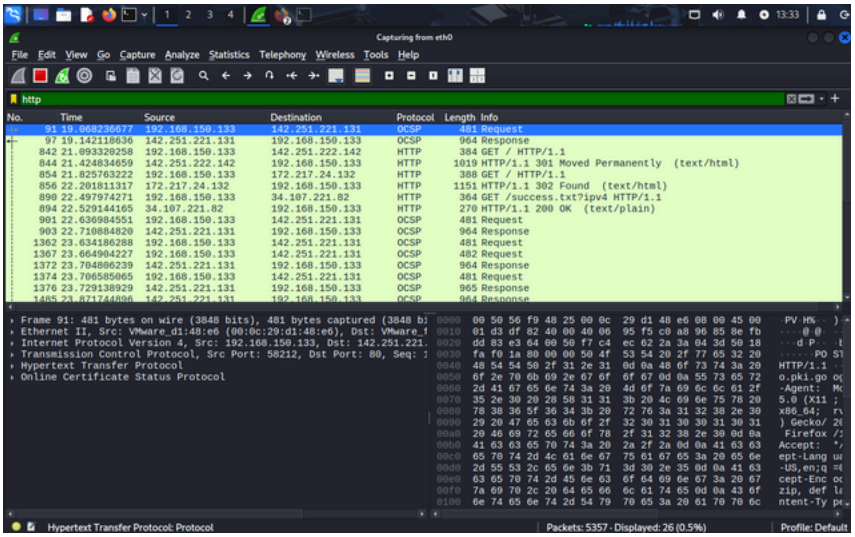
- Installed and launched Wireshark on Kali Linux.
- Selected the active network interface (wlan0).
- Started packet capture.
- Generated network activity by browsing example.com and pinging 8.8.8.8.
- Stopped capture after ~60 seconds.
- Applied protocol filters for HTTP, DNS, and TLS.
- Analyzed packet details and noted source/destination IPs, ports, and protocol functions.
- Saved the capture file as task5_capture.pcap.

Screenshots



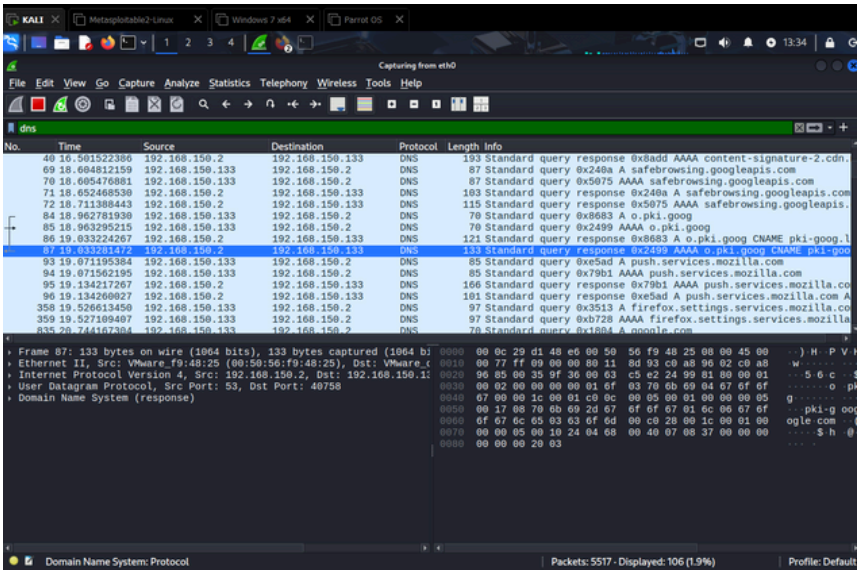
Overview of Wireshark

This image shows a Wireshark capture in progress on your Kali Linux system, specifically on the wlan0 interface. It displays a variety of network packets with different protocols, including TCP, QUIC, and ARP. This screenshot demonstrates the initial phase of capturing live network traffic before any filters are applied.



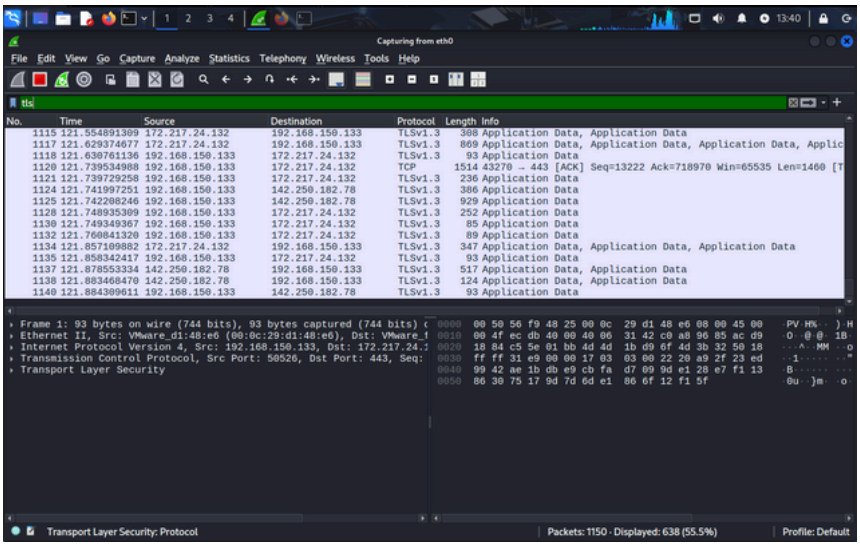
http filter

This image shows Wireshark with an http filter applied. You can see several packets related to HTTP, including GET and POST requests. This screenshot visually confirms the capture of unencrypted web traffic, showing the transfer of web page data between a client and a server.



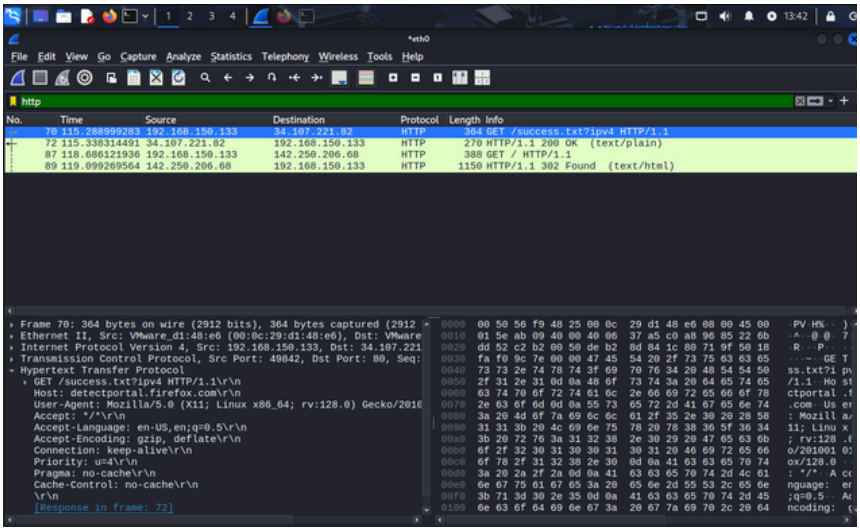
dns filter

This screenshot displays Wireshark filtered for dns traffic. The packets visible are DNS queries and responses, which are used to resolve domain names to IP addresses. This image provides visual evidence of how the computer performs DNS lookups during network activity.



tls filter

This image shows Wireshark filtered for tls traffic. The packets in the capture are related to Transport Layer Security (TLS), which is used to encrypt web traffic for confidentiality and integrity. This screenshot demonstrates that secure, encrypted communication is also being captured and analyzed.



packet details

Protocols Identified

PROTOCOL	PURPOSE	EXAMPLE FROM CAPTURE
HTTP	Transfers web page data between client and server	GET request to example.com
DNS	Resolves domain names to IP addresses	Query for example.com
TLS	Encrypts web traffic for confidentiality and integrity	Encrypted HTTPS session to example.com

Key Learnings

- Wireshark provides in-depth visibility into network communications.
- Protocol filters help isolate and study specific traffic types.
- Packet-level analysis can aid in troubleshooting network issues and detecting suspicious activity.

Conclusion

This task demonstrated how Wireshark can be used to capture and analyze network traffic effectively. By applying filters for HTTP, DNS, and TLS, it was possible to identify key protocols and their functions within normal network activity. The exercise reinforced packet analysis skills and highlighted the role of encrypted communication in protecting data confidentiality.