TASK 5

Wireshark Network Traffic Capture & Protocol Analysis

⭐ Overview

Captured and analyzed live network traffic using Wireshark to understand key internet protocols and communication flows. This project demonstrates practical packet analysis skills, protocol identification, and basic network monitoring—crucial for entry-level cybersecurity roles.

🛠️ Tools Used

- Wireshark (free packet analyzer)
- Windows/Linux/Mac (any OS with Wireshark support)
- Internet browser & command line (to generate traffic)

📁 Deliverables

- network_analysis.pcap – Raw packet capture file (Wireshark export)
- protocol_report.md – Short report summarizing protocols and findings

🚦 Steps Performed

1. Installed Wireshark and required capture drivers (Npcap or equivalent).
2. Chose my active network interface (Wi-Fi/Ethernet) for packet capture.
3. Generated network traffic by browsing websites and running ping tests.
4. Captured live packets for 1 minute, then stopped recording.
5. Filtered and analyzed traffic by protocol: HTTP, DNS, TCP, ICMP, etc.
6. Identified source/destination IP addresses, packet purposes, and communication types.
7. Exported capture as a .pcap file for review and sharing.
8. Summarized protocol findings in a clear, concise report.

🔍 Key Findings

- HTTP: Web browsing traffic, showing requests to example domains.
- DNS: Domain name resolution packets sent to public DNS servers.
- TCP: Transport layer connections supporting web traffic and pings.
- ICMP: Ping requests/replies for connectivity testing.

📊 Example Protocol Report Snippet

text

Capture Duration: 1 minute

Packets Captured: 550

Protocols Identified:

- HTTP: 45 packets (Web communication)

- DNS: 35 packets (Domain name queries)

- ICMP: 12 packets (Ping requests and replies)

Details:

- HTTP: 192.168.1.2 → 93.184.216.34

- DNS: 192.168.1.2 → 8.8.8.8

- ICMP: 192.168.1.2 ↔ 93.184.216.34

THANK YOU