Task 5 Report: Capture and Analyse Network Traffic Using Wireshark

Objective

The objective of this task was to capture live network packets using Wireshark, analyze the captured traffic, and identify different protocols. This exercise helps in understanding network communication and troubleshooting network issues.

Tools Used

- Wireshark – A free and open-source packet analyser.

Steps Followed

1. Installed Wireshark on my system.
2. Started capturing packets on the active network interface (Wi-Fi/Ethernet).
3. Generated traffic by:
   - Browsing a website (e.g., google.com).
   - Running a ping command (e.g., ping 8.8.8.8).
4. Stopped the capture after approximately one minute.
5. Applied filters to identify different protocols (e.g., http, dns, tcp, udp).
6. Exported the capture as a .pcap file.
7. Summarized findings in this report.

Protocols Identified

1. HTTP (Hypertext Transfer Protocol)
   - Used for web browsing.
   - Example: Packets sent/received when accessing google.com.
2. DNS (Domain Name System)
   - Resolves domain names to IP addresses.
   - Example: google.com → 142.250.190.46.
3. TCP (Transmission Control Protocol)
   - Ensures reliable, connection-oriented communication.
   - Example: Three-way handshake (SYN, SYN-ACK, ACK).

4. ICMP (Internet Control Message Protocol)
    - Used for ping requests and responses.
    - Example: ping 8.8.8.8 generated ICMP Echo Request/Reply packets.

Key Observations

- Most traffic was encrypted (HTTPS/TLS), making it difficult to inspect payloads without decryption keys.
- DNS queries were visible, showing how domain names are resolved.
- TCP handshake was observed before HTTP/HTTPS connections.
- ICMP packets confirmed network reachability tests (ping).

Challenges Faced

- Filtering packets: Initially struggled with Wireshark's filter syntax but later learned to use expressions like tcp.port == 80 or dns.
- Encrypted traffic: Could not inspect HTTPS payloads, only metadata (headers, timing, etc.).
- Noise in capture: Unnecessary background traffic had to be filtered out.

Conclusion

This task provided hands-on experience in network packet analysis using Wireshark. By capturing and filtering traffic, I identified key protocols (HTTP, DNS, TCP, ICMP) and understood their roles in network communication. This skill is essential for cybersecurity, troubleshooting, and network monitoring.