Cyber Security Internship – Task 5

Task Title: Capture and Analyze Network Traffic Using Wireshark

Submitted By: Gourav Swaroop

Date: 11/08/2025

Objective:
To capture live network packets and identify basic protocols and traffic types using Wireshark, gaining hands-on experience in packet analysis and protocol awareness.

Tools Used:

- **Wireshark** (Version: latest available at time of capture)

Capture Details:

- **Capture Duration:** ~1 minute
- **Network Interface:** Active network interface of the host system
- **Capture Files Generated:**
  - Full Capture.pcapng (complete raw capture)
  - Filtered Packets.pcapng (filtered using: dns or tcp or udp or icmp or http)

Procedure Followed:

1. **Wireshark Installation:**
   Installed and configured Wireshark to capture packets on the main network interface.
2. **Traffic Generation:**
   During the capture, normal browsing activity was performed, including visiting websites and executing ping commands to generate ICMP traffic.
3. **Packet Capture:**
   Started capture on the active interface, generated traffic, and stopped capture after about 1 minute.
4. **Filtering:**
   Applied protocol filters to narrow down to key protocols (DNS, TCP, ICMP, ARP, TLSv1.2).
5. **Analysis:**
   Identified and examined multiple packet types, reviewing their source/destination, length, and detailed protocol information.

Protocols Identified:

1. **DNS (Domain Name System)**
   - **Purpose:** Resolves human-readable domain names into IP addresses.
   - **Observation:** Multiple standard DNS queries and responses were captured, such as requests to google.com and reverse DNS lookups for IPv6 addresses.
2. **ICMPv6 (Internet Control Message Protocol, IPv6)**
   - **Purpose:** Used for diagnostic functions like ping, as well as IPv6 neighbor discovery.
   - **Observation:** Captured echo requests and replies (ping), with hop limits set and successful responses received.
3. **TLSv1.2 (Transport Layer Security)**
   - **Purpose:** Encrypts communication between client and server for secure data transfer.
   - **Observation:** Multiple TLSv1.2 packets were seen, indicating secure HTTPS connections to remote servers over TCP port 443.
4. **TCP (Transmission Control Protocol)**
   - **Purpose:** Provides reliable, ordered, and error-checked delivery of data between applications.
   - **Observation:** TCP handshake and acknowledgment packets were present, supporting HTTPS and other communications.
5. **ARP (Address Resolution Protocol)**
   - **Purpose:** Resolves IPv4 addresses to MAC addresses within a local network.
   - **Observation:** One ARP request was detected, mapping the local network device’s MAC address.

Sample Packet Details:

- **DNS Query Example:**
  - **Source:** 172.20.10.1
  - **Destination:** 8.8.8.8
  - **Query:** google.com
  - **Response:** IPv4 address of google.com
- **ICMPv6 Echo Request:**

  - **Source:** Local IPv6 address
  - **Destination:** Remote IPv6 host
  - **Info:** Request with hop limit of 64, successfully replied.
- **TLSv1.2 Packet:**
  - **Source Port:** 443
  - **Destination Port:** Random high TCP port (e.g., 34770)
  - **Purpose:** Encrypted HTTPS communication.

Conclusion:

This exercise successfully demonstrated the process of capturing and analyzing network traffic using Wireshark. At least **five protocols** (DNS, ICMPv6, TCP, TLSv1.2, and ARP) were identified, analyzed, and documented. The findings highlight the variety of background and active communications happening on a typical network connection, reinforcing the importance of protocol awareness in network security and diagnostics.