Network Traffic Analysis Report: Wireshark Protocol Interpretation

Task Objective: To capture network traffic using Wireshark, filter the data, and provide a detailed summary of key communication protocols based on packet characteristics.

Deliverable 1: Packet Capture Submission

The network traffic was captured from the active network interface during a standard web browsing session. The full dataset is submitted as the raw capture file.

- **Attachment Submitted:** MyNetworkCapture.pcapng

Deliverable 2: Summary of Findings and Packet Details

The traffic capture was filtered and analysed to isolate four fundamental protocols, demonstrating proficiency in interpreting network communication layers.

1. Transport Layer Protocols

These protocols manage how data segments are transferred between applications on hosts.

- **TCP (Transmission Control Protocol)**
  - **Function:** Provides a **reliable, connection-oriented** service, ensuring data is delivered completely and in order.
  - **Packet Details & Findings:**
    - **Connection Lifecycle:** Identified the sequential flag packets that define the full connection life cycle. This includes the establishment phase (the **three-way handshake**: **[SYN]**, **[SYN, ACK]**, and **[ACK]**), and the termination phase (**[FIN, ACK]**) for clean session closure.
- **UDP (User Datagram Protocol)**
  - **Function:** Provides a **fast, connectionless** transfer service with minimal overhead; it does not guarantee delivery or order.
  - **Packet Details & Findings:**
    - UDP was confirmed as the efficient carrier for rapid, query-based Application Layer traffic (specifically DNS). The packets' simple header structure, which lacks the sequencing and acknowledgment fields of TCP, confirms its connectionless design.

2. Application and Security Layer Protocols

These protocols handle application-specific data formatting and secure data transmission.

- **DNS (Domain Name System)**
  - **Function:** Application protocol responsible for translating domain names (URLs) into numerical IP addresses, which is necessary for traffic routing.
  - **Packet Details & Findings:**
    - The complete transaction was observed: a **"Standard query"** packet sent by the host was immediately followed by a server **"Standard query response,"** confirming successful name resolution. This traffic was consistently encapsulated within **UDP** packets.
- **TLS (Transport Layer Security)**
  - **Function:** Security protocol that provides **encryption, integrity, and authentication** for data exchange, forming the basis of HTTPS.
  - **Packet Details & Findings:**
    - Filtering by the TLS protocol revealed the necessary security handshake. The process begins with the **"Client Hello"** message, which initiates the handshake by defining the client's supported cipher suites and protocol versions to the server.

Conclusion

The comprehensive analysis of the captured traffic demonstrates a clear understanding of fundamental network layers and protocol interaction. By isolating **TCP** and **UDP** at the transport level, and identifying the functions and characteristics of **DNS** and **TLS** at the application and security levels, the project successfully fulfilled the objective. The findings confirm proficiency in using Wireshark to capture, filter, and interpret live network traffic data for technical analysis.