**Project : Packet Analysis with Wireshark – Analyzing google.com**

**Objective:**

To capture and analyze the network packets generated while accessing **https://google.com**, with a focus on DNS resolution, TCP handshakes, and encrypted HTTPS traffic.

**Tools Used:**

- **Wireshark** (Network protocol analyzer)
- **Web Browser:** Google Chrome
- **Website Visited:** https://google.com
- **Operating System:** Windows

**Steps Followed:**

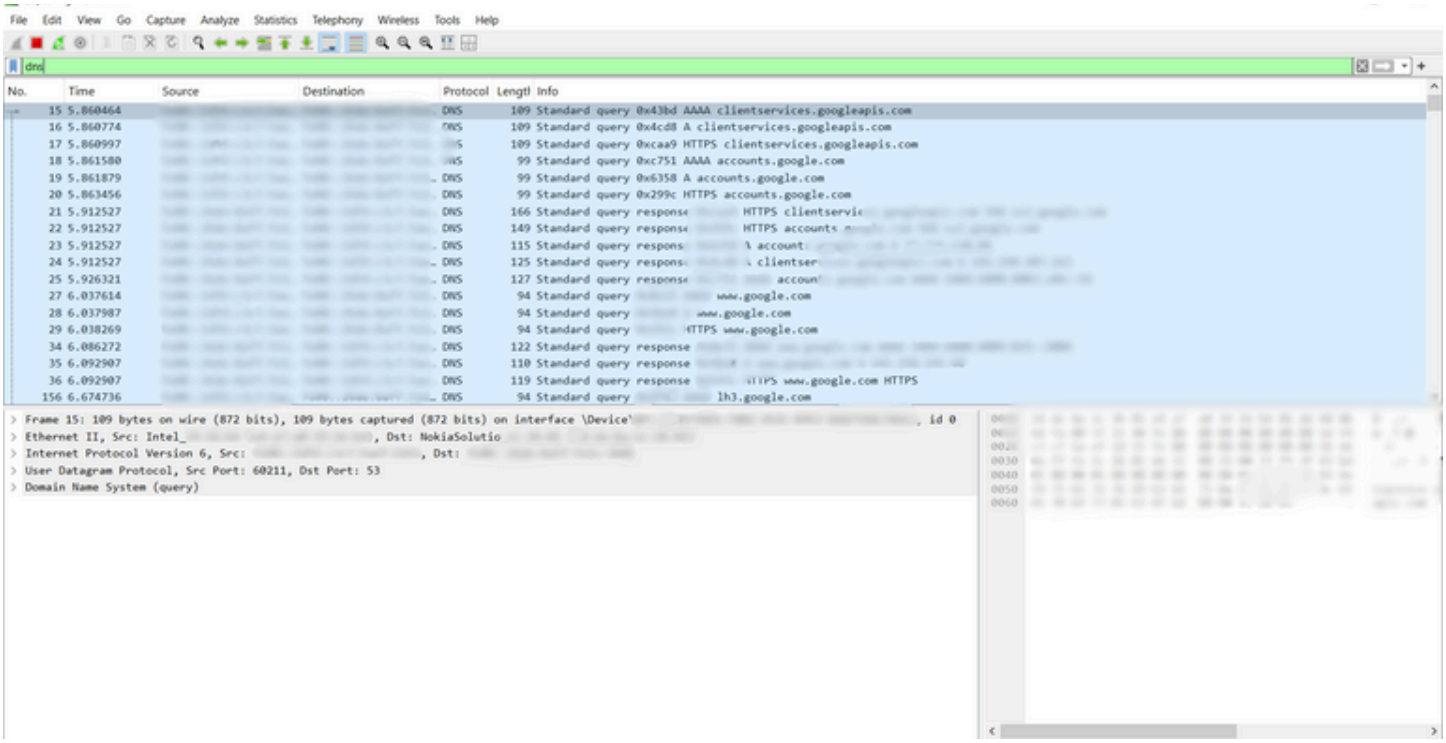
1. **Installed Wireshark** from https://www.wireshark.org and launched the tool.
2. **Selected Network Interface** (Wi-Fi) to capture live traffic.
3. **Started Packet Capture** and visited https://google.com in the browser.
4. **Applied Filters:**
    - dns → To capture the DNS request resolving google.com.
    - tcp.port == 443 → To view HTTPS traffic.
    - tls → To view the TLS handshake packets.

**Analyzed Captured Packets:**

- **DNS Lookup:**
    - Observed DNS query and response for resolving google.com.
- **TCP Handshake:**
    - Captured 3-way TCP handshake between my system and Google's server.
- **TLS Handshake:**
    - Observed the start of secure communication.
    - Captured "Client Hello", "Server Hello", and key exchange mechanisms.

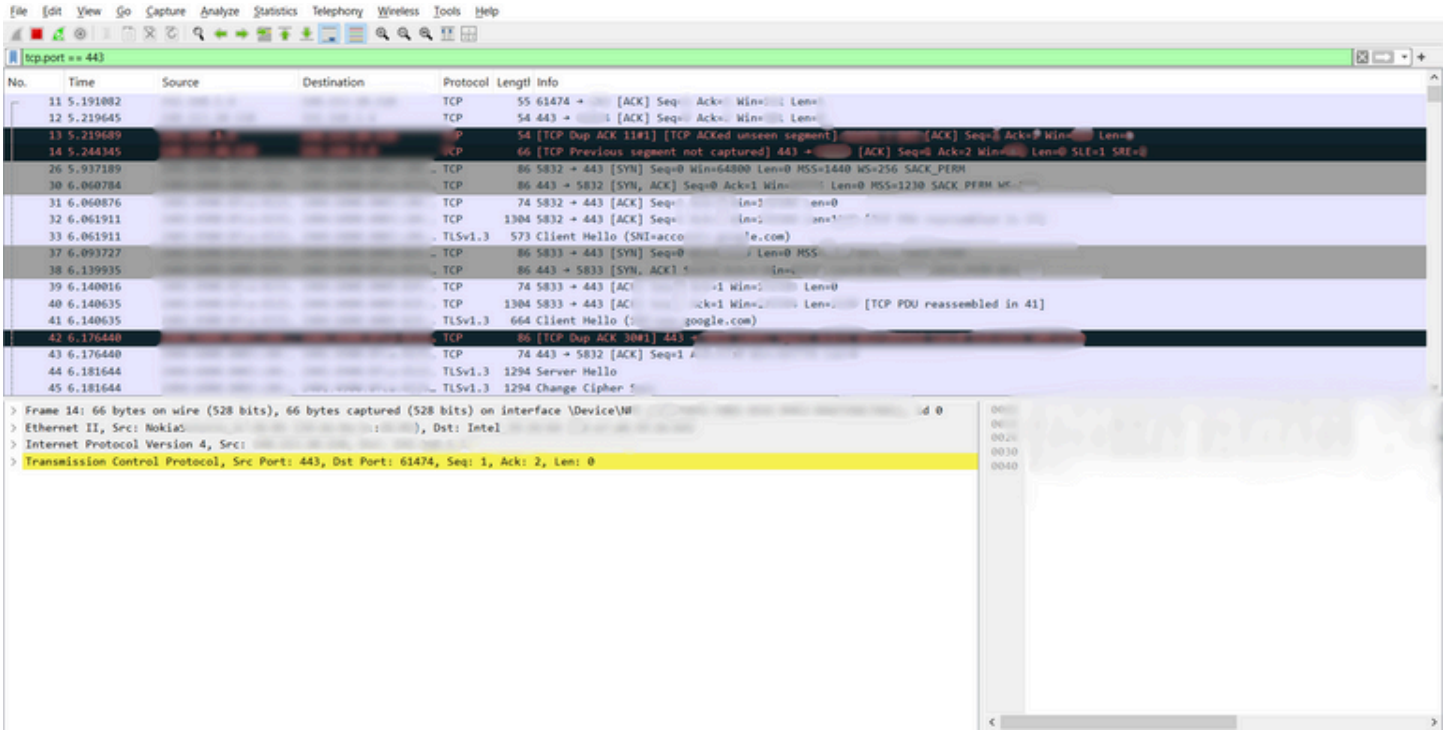
**Results:**

1. **Filter: dns**



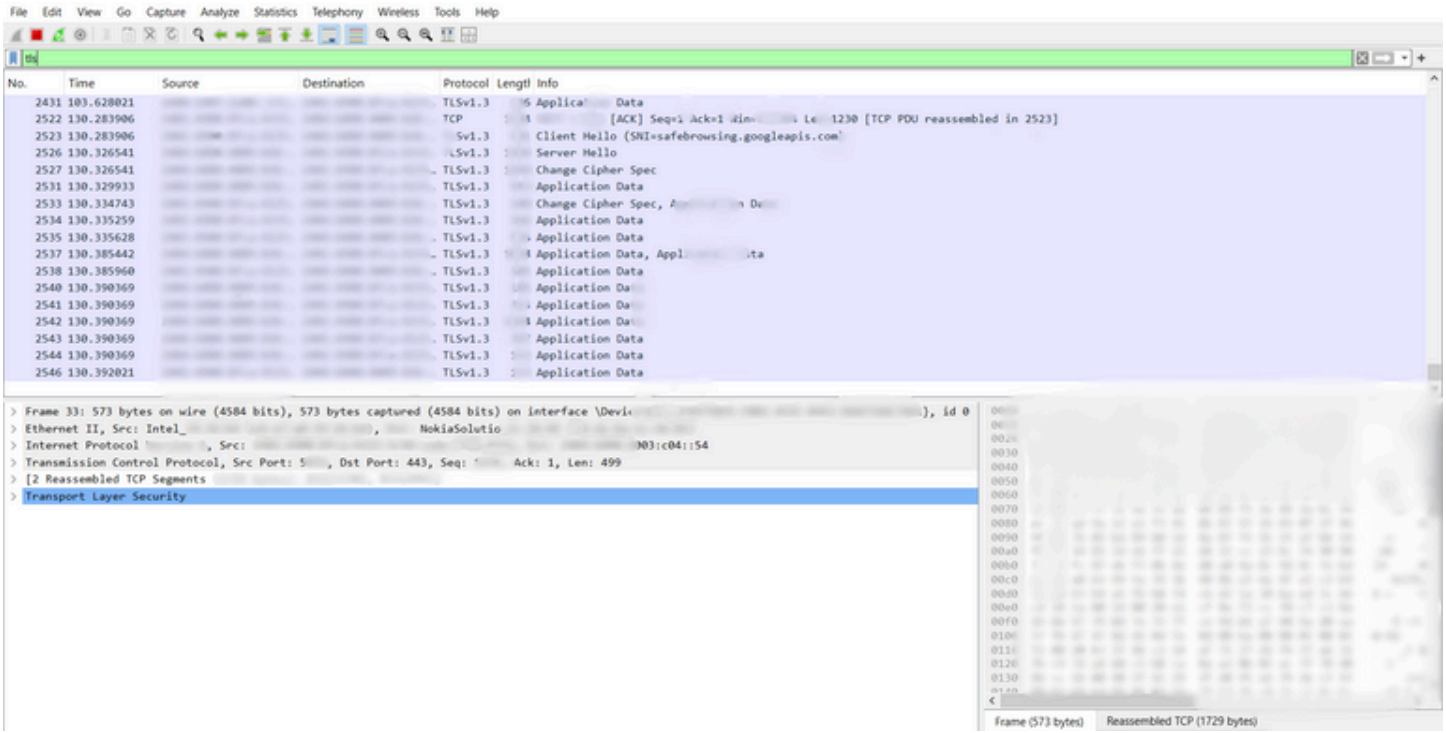

This filter displays all DNS traffic captured during browsing. It includes DNS queries and responses, showing how the domain name google.com was resolved to its IP address.

### 🔹 2. Filter: tcp.port == 443



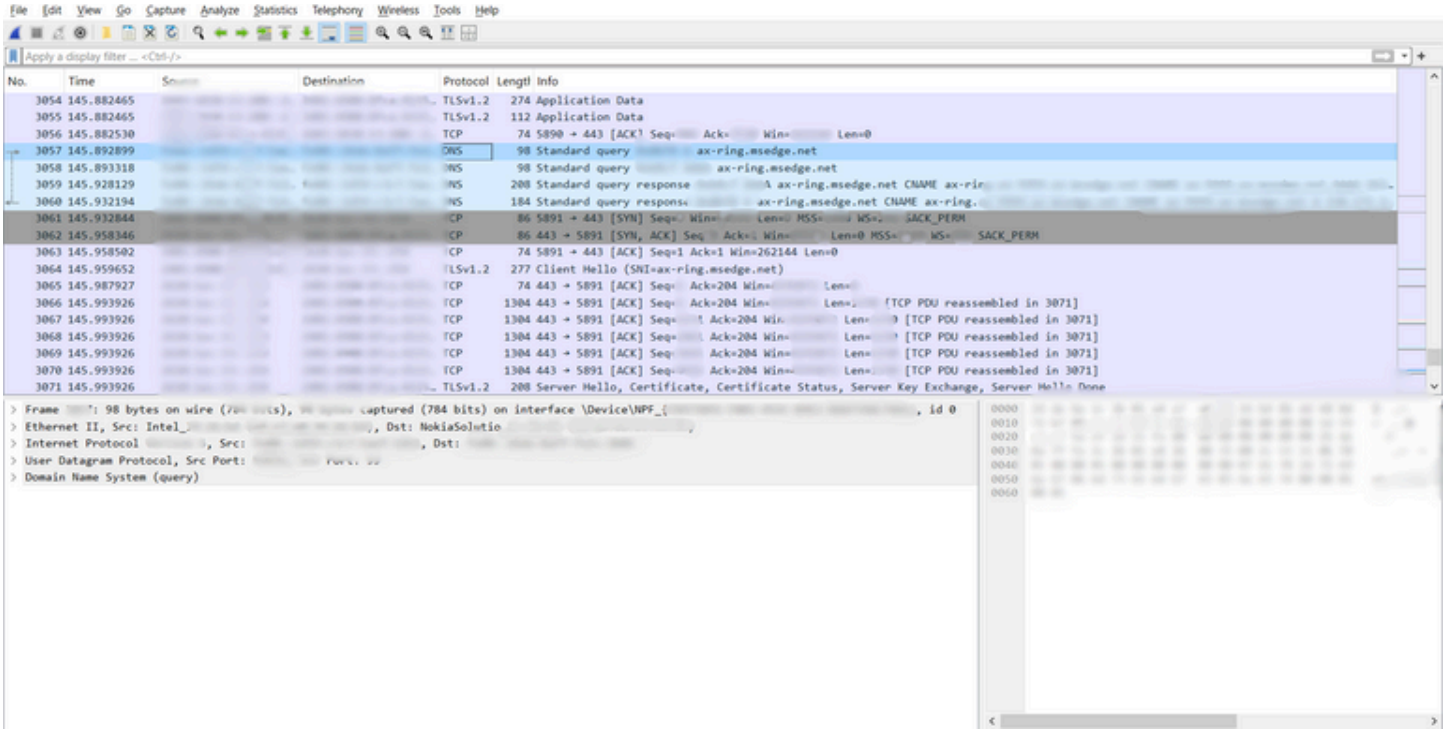

This filter highlights TCP packets using port 443, indicating encrypted HTTPS traffic. It shows the secure communication channel established between the browser and google.com.

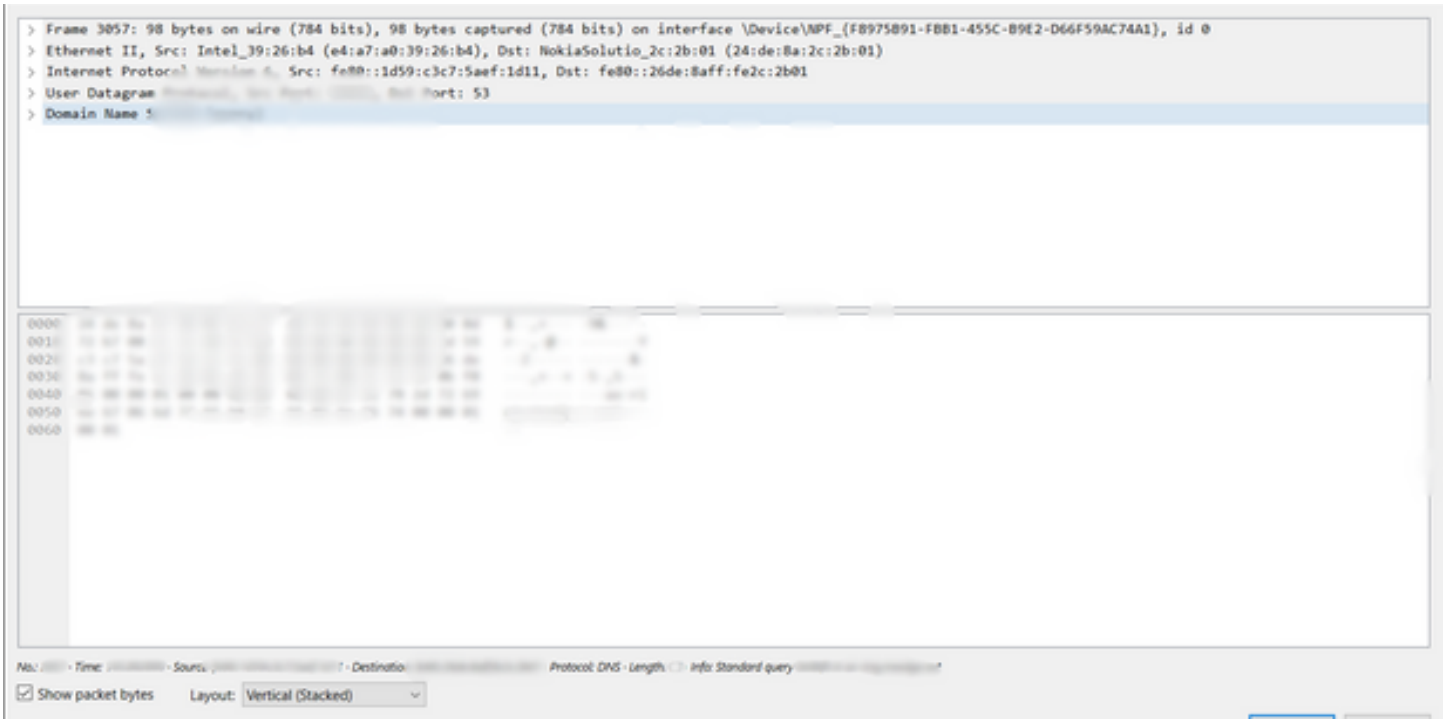
### 🔹 3. Filter: tls




This filter displays packets related to the TLS handshake process. It includes details like Client Hello and Server Certificate, confirming secure encrypted communication with google.com.

**-DNS Packet Details for google.com:**






This screenshot shows the breakdown of a DNS packet captured during access to google.com.

**Screenshots Included:**

- DNS query and response for google.com
- TCP 3-way handshake packets
- TLS handshake sequence
- Encrypted HTTPS traffic packets

**Key Observations:**

- **DNS resolution** is always the first step in accessing a domain, regardless of the protocol.
- **TLS handshake** ensures secure encryption between client and server.
- **All application data was encrypted**, confirming Google's HTTPS implementation is secure.

**Conclusion:**

This project demonstrated how HTTPS traffic is handled on a network level. Even though actual data isn't visible due to encryption, the initial DNS and TLS handshake packets still provide meaningful insight. This exercise strengthened my understanding of:

- DNS resolution process
- TCP and TLS handshake protocols
- The importance of HTTPS for secure communication

By analyzing google.com, I learned how modern websites ensure secure data transmission and how tools like Wireshark help in observing the underlying protocols.