Encryption Report; Google.com

Introduction

This report analyzes network traffic sent to Google.com using Wireshark on a Kali Linux virtual machine. The analysis focused on timestamps, IP addresses, port numbers, and encryption status of the packets. The findings confirm all traffic used HTTPS (port 443) and TLSv1.3 encryption for secure communication.

Methodology

A. Tool used

Wireshark: Network protocol analyzer
Mozilla Browser: Web browser for generating traffic
Kali Linux Virtual Machine: Secure environment for analysis

B. Procedure Followed:

Step 1: Launch Wireshark on Kali Linux VM.

2. Select the Network Interface:

On the input page, I selected the network interface to connect to the internet (in this instance, Ethernet, Wi-Fi). I ensured that the Promiscuous mode on all interfaces was enabled. On output, I created a permanent file to capture for future reference.

Step 2: Start Capturing Traffic:

Click on the `Start` button to begin capturing packets.

Step 3: Initiate the HTTPS Connection

1. Open a Web Browser:

For this analysis, Mozilla Firefox was used.

2. Visit Google:

Navigated to https://www.google.com.

Step 4: Stop Capturing Traffic

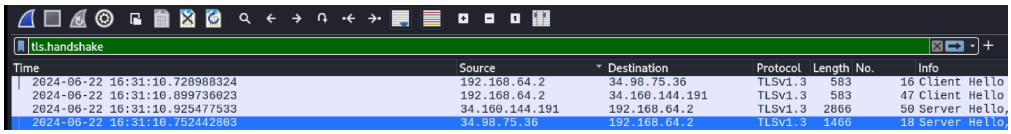
- After the page had loaded, I returned to Wireshark and clicked on the `Stop` button to stop capturing packets.

Step 5: Filter and Analyze the TLS Handshake

1. Apply Display Filter:

- the display filter was used to isolate the TLS handshake packets. In the Wireshark filter bar, the keyword ‘tls.Handshake’ was used to filter the tls handshake.

2. Result of the filtered packets URL:



- Client Hello (SNI=classify-client.service.mozilla.com)

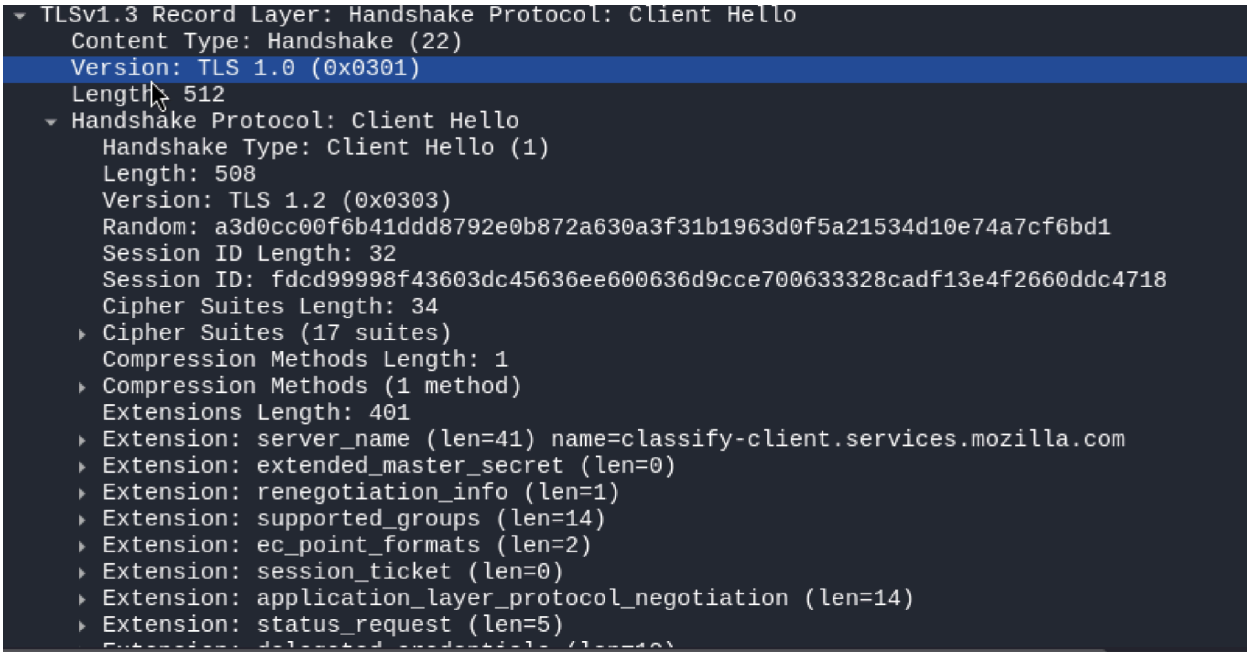
- Server Hello, Change Cipher Spec

- Client Hello(SNI=content-signature-2.cdn.mozilla.net)

- Server Hello, Change Cipher Spec

TLS Handshake Analysis

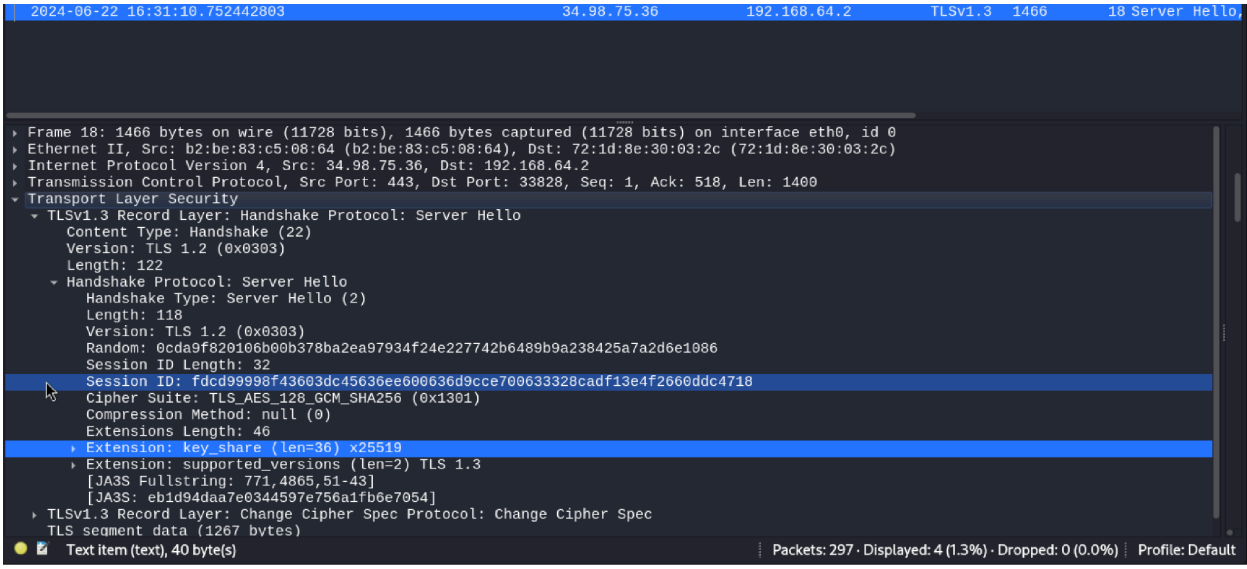
- ClientHello Message



In the TLS handshake, the ClientHello message acts as an initiation and introduction. Here's a breakdown of its purpose and significance:

- **Purpose:**
  - Initiates the secure communication process between a client (your browser) and a server (website).
  - Informs the server about the client's capabilities for secure communication.
- **Significance:**
  - Establishes a foundation for secure communication by initiating the negotiation process.
  - Allows the server to choose the most compatible and secure encryption settings from the options offered by the client.
  - Contains essential information like:
  - Encrypted: Yes (TLS handshake initiated)
  - Supported TLS versions (e.g., TLSv1.3)
  - Preferred cipher suites (combinations of encryption algorithms)
  - A random number used to generate encryption keys

- ServerHello Message



Purpose:

- Respond to the client's initiation (ClientHello message) and acknowledge the request for a secure connection.
- Informs the client about the server's chosen settings for secure communication.

Significance:

- Completes the initial negotiation phase by confirming the chosen encryption parameters.

- Establishes a foundation for secure communication by agreeing on the specific algorithms and keys used for encryption.
- Contains essential information like:
- - Encrypted: Yes (TLS handshake response)
- The selected TLS version (e.g., TLSv1.3) - must be compatible with what the client offered.
- Chosen cipher suite (encryption algorithms) - picked from the options proposed by the client based on compatibility and server policy.
- The server's random number - it is used along with the client's random number to generate encryption keys for the session.
- Server certificate (optional): If server authentication is required, the certificate containing the server's public key is sent for client verification.

- **Key Exchange Message**
  -
  - Purpose and Significance
    - Explain the purpose and significance of the ServerHello message in the TLS handshake process.

Packet Details

A. Source and Destination Information

1. ClientHello Packet
   - Timestamp: 2024-06-22 16:31:10.728988324
   - - Source IP: `192.168.64.2`
   - - Destination IP: `34.98.75.36`
   - - Port: `443`
2. ServerHello Packet
   - Timestamp: 2024-06-22 16:31:10.899736023
   - Source IP: `34.98.75.36
   - Destination IP: 192.168.64.2`
   - Port: 443
3. Key Exchange Packet
   - Source IP:
   - Source Port:
   - Destination IP:
   - Destination Port:

B. Encryption Verification