Encryption Report

Introduction

The objective of this report is to analyze the TLS handshake for an HTTPS connection to https://www.google.com using Wireshark on a Kali Linux virtual machine. The report will provide detailed information on the ClientHello, ServerHello, and Key Exchange messages exchanged during the TLS handshake, including timestamps, IP addresses, port numbers, and the verification of encrypted packets post-handshake.

Methodology

A. Tools Used

- Wireshark

B. Procedure followed

1. Setting Up the Environment

Ensure you have a Kali Linux VM and Wireshark installed.

1. **Install Wireshark**

2. **Run Wireshark with root privileges:**

sudo wireshark

2. Configuring Wireshark to Capture Traffic

Select the Network Interface:

- Open Wireshark.
- Choose the network interface that your VM uses to connect to the internet. This is typically eth0.

Start the Capture:

- Click on the selected interface.
- Click the **Start** button.

3. Initiate HTTPS Connection

Open a Web Browser:

- Open a web browser on your Kali VM.

Visit https://www.google.com:

- Type the URL in the address bar and press Enter.

4. Capturing the TLS Handshake

Stop the Capture:

- Once the page loads, return to Wireshark and click the **Stop** button

Filter the Traffic:

- In the Wireshark filter bar, type tls and press Enter. This filters the captured packets to show only TLS traffic.

5. Analyzing the TLS Handshake

Identify the TLS Handshake Packets:

- Look for packets labeled as ClientHello, ServerHello, and other handshake-related messages.

Examine ClientHello Packet:

- Click on the ClientHello packet.
- Expand the sections in the packet details pane:
  - **TLSv1.3 Record Layer**.
  - **Handshake Protocol: Client Hello**.

Examine ServerHello Packet:

- Click on the ServerHello packet.
- Expand the sections in the packet details pane:
  - **TLSv1.3 Record Layer**.
  - **Handshake Protocol: Server Hello**.

Select an Application Data Packet:

- Scroll through the filtered list to find a packet labeled as Application Data.

Expand Packet Details:

- Click on the Application Data packet to select it.
- In the packet details pane, expand the sections:
- **Frame**
- **Ethernet II**
- **Internet Protocol Version 4 (or IPv6)**
- **Transmission Control Protocol (TCP)**
- **Transport Layer Security (TLS)**

Highlight Encrypted Data:

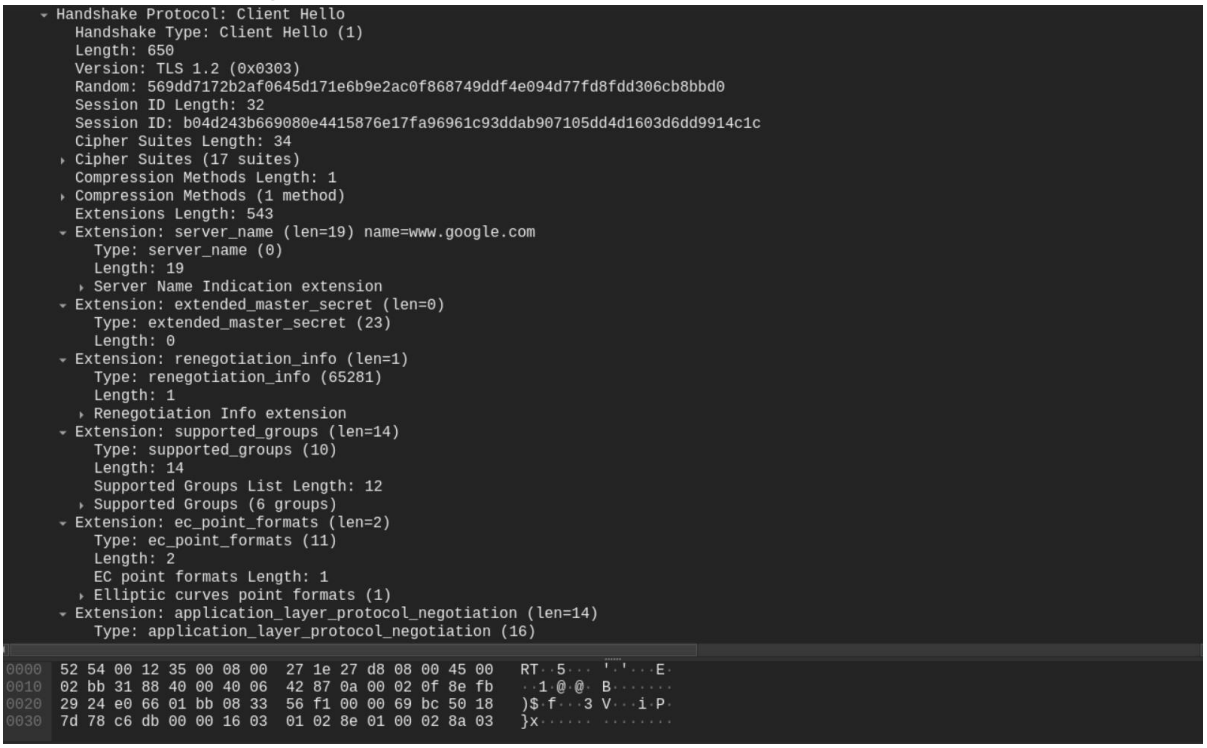
- Highlight the section showing Application Data.
- This section confirms that the data is encrypted.

Examine Finished Packets:

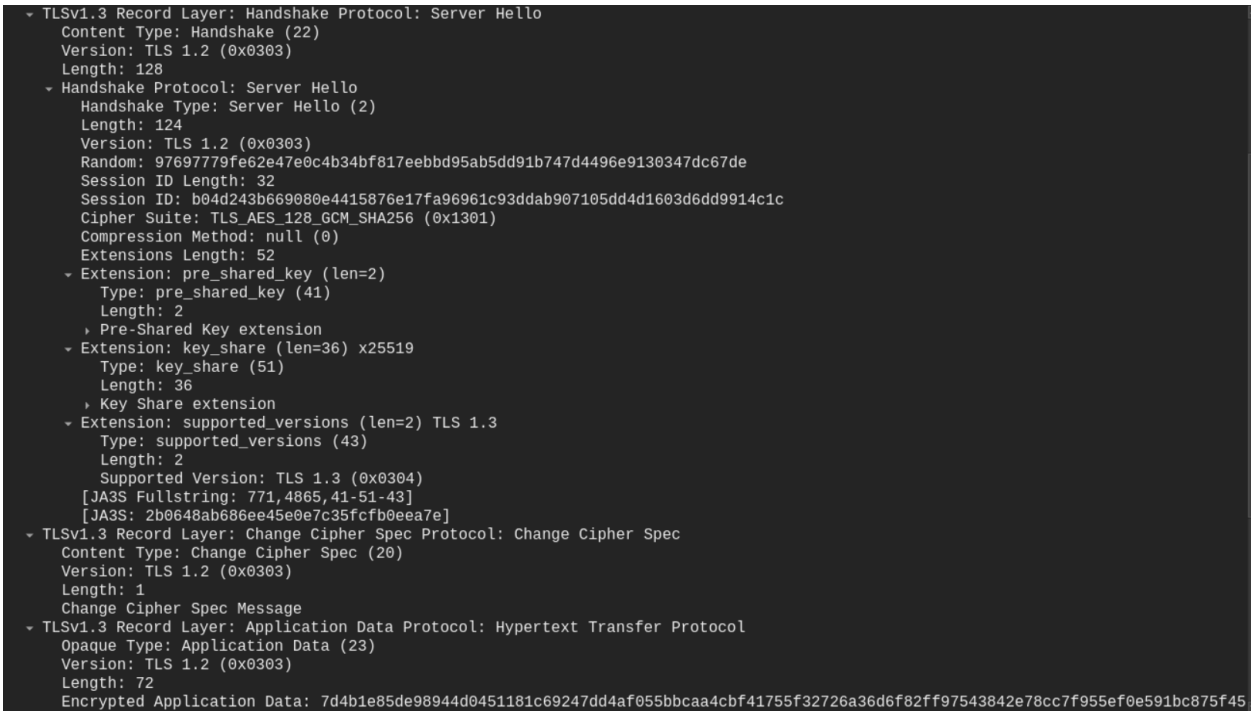
- These packets mark the end of the handshake and the beginning of encrypted application data.

TLS Handshake Analysis

- ClientHello Message

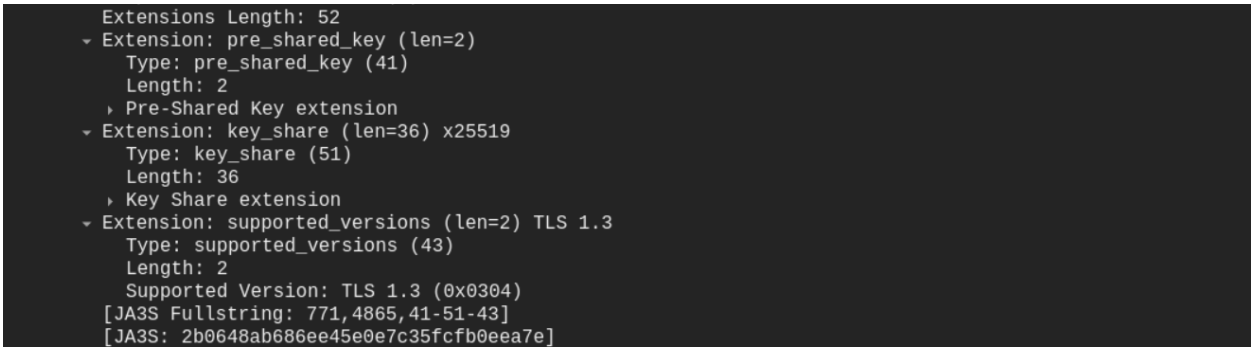


  - Timestamp: 2024/216 17:38:35 .292770846
  - Purpose and Significance
    - The ClientHello message is the initial step in the TLS handshake, where the client proposes security parameters (such as supported cipher suites and TLS versions) to the server.
- ServerHello Message



- Timestamp: 2024/216 17:38:35 .367331023
- Purpose and Significance
    - The ClientHello message is the initial step in the TLS handshake, where the client proposes security parameters (such as supported cipher suites and TLS versions) to the server.

- Key Exchange Message



    - Purpose and Significance
        - The Key Exchange message establishes the shared secret key used for encrypting the session. This message completes the key exchange process necessary for secure communication.

Packet Details

A. Source and Destination Information

1. ClientHello Packet
- Source IP: 10.0.2.15
- Source Port: 49640
- Destination IP: 142.251.41.36
- Destination Port: 443

2. ServerHello Packet
- Source IP: 142.251.41.36
- Source Port: 443
- Destination IP: 10.0.2.15
- Destination Port: 49640

3. Key Exchange Packet
- In TLSv1.3 the key exchange information is given via the ServerHello packet under extensions.

B. Encryption Verification

[Screenshot to confirm that packets are encrypted after the handshake.]