Gmail Packet Capture Analysis

A Detailed Network Traffic Examination

Prepared by:

Aayush Adhikar
Roshan Tiwari
Shishir Sharma Rijal
Sudip Acharyai

July 17, 2024

Contents

1 Introduction

This report presents a comprehensive analysis of network packets captured during the process of sending an email via Gmail's web interface. The experiment was conducted using Wireshark on a Mac computer, focusing on the intricate details of the captured packets without applying specific filters.

2 Methodology

2.1 Tools and Environment

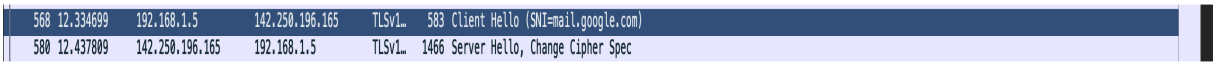
- **Packet Capture Tool:** Wireshark (latest version)
- **Operating System:** macOS (version X.X)
- **Web Browser:** Google Chrome (version X.X)
- **Network:** Wi-Fi connection to a home router

2.2 Procedure

1. Launch Wireshark and initiate packet capture on the Wi-Fi interface.
2. Open Google Chrome and navigate to Gmail (`https://mail.google.com`).
3. Log into a Gmail account.
4. Compose and send a test email.
5. Stop the Wireshark packet capture.
6. Analyze the captured packets, focusing on those related to the Gmail session filtered by `tcp.port==443` and searching for `mail.google.com`.

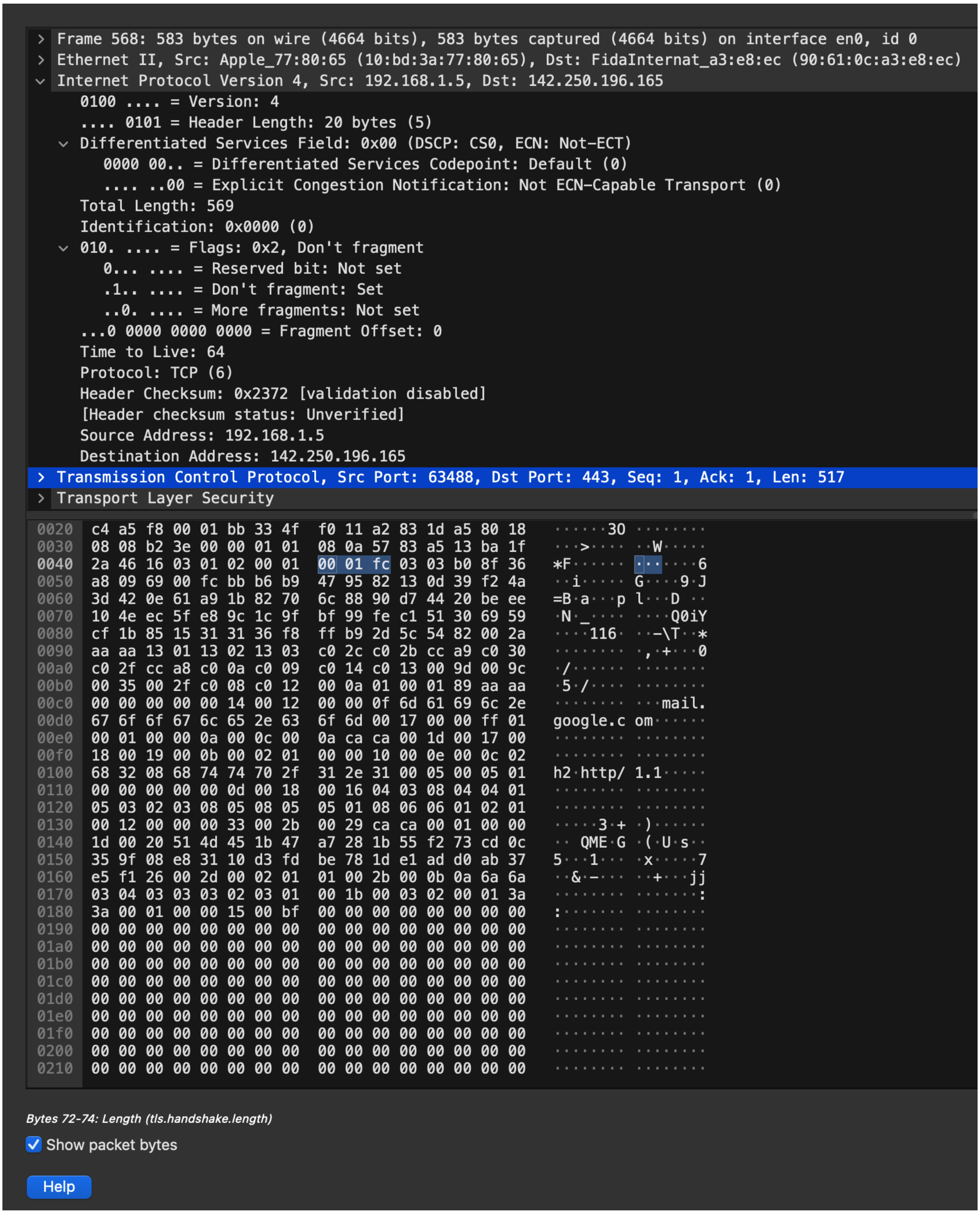
3 Results and Analysis

3.1 Overview of Captured Packet



```
568 12.334699   192.168.1.5       142.250.196.165   TLSv1…   583 Client Hello (SNI=mail.google.com)
580 12.437809   142.250.196.165   192.168.1.5       TLSv1…   1466 Server Hello, Change Cipher Spec
```

Figure 1: Packet



```
> Frame 568: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface en0, id 0
> Ethernet II, Src: Apple_77:80:65 (10:bd:3a:77:80:65), Dst: FidaInternat_a3:e8:ec (90:61:0c:a3:e8:ec)
v Internet Protocol Version 4, Src: 192.168.1.5, Dst: 142.250.196.165
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  v Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 569
    Identification: 0x0000 (0)
  v 010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x2372 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.5
    Destination Address: 142.250.196.165
> Transmission Control Protocol, Src Port: 63488, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
> Transport Layer Security

0020  c4 a5 f8 00 01 bb 33 4f  f0 11 a2 83 1d a5 80 18   ······3O ········
0030  08 08 b2 3e 00 00 01 01  08 0a 57 83 a5 13 ba 1f   ···>···· ··W·····
0040  2a 46 16 03 01 02 00 01  00 01 fc 03 03 b0 8f 36   *F······ ·······6
0050  a8 09 69 00 fc bb b6 b9  47 95 82 13 0d 39 f2 4a   ··i····· G····9·J
0060  3d 42 0e 61 a9 1b 82 70  6c 88 90 d7 44 20 be ee   =B·a···p l···D ··
0070  10 4e ec 5f e8 9c 1c 9f  bf 99 fe c1 51 30 69 59   ·N·_···· ····Q0iY
0080  cf 1b 85 15 31 31 36 f8  ff b9 2d 5c 54 82 00 2a   ····116· ··-\T··*
0090  aa aa 13 01 13 02 13 03  c0 2c c0 2b cc a9 c0 30   ········ ·,·+···0
00a0  c0 2f cc a8 c0 0a c0 09  c0 14 c0 13 00 9d 00 9c   ·/······ ········
00b0  00 35 00 2f c0 08 c0 12  00 0a 01 00 01 89 aa aa   ·5·/···· ········
00c0  00 00 00 00 00 14 00 12  00 00 0f 6d 61 69 6c 2e   ········ ···mail.
00d0  67 6f 6f 67 6c 65 2e 63  6f 6d 00 17 00 00 ff 01   google.c om······
00e0  00 01 00 00 0a 00 0c 00  0a ca ca 00 1d 00 17 00   ········ ········
00f0  18 00 19 00 0b 00 02 01  00 00 10 00 0e 00 0c 02   ········ ········
0100  68 32 08 68 74 74 70 2f  31 2e 31 00 05 00 05 01   h2·http/ 1.1·····
0110  00 00 00 00 00 0d 00 18  00 16 04 03 08 04 04 01   ········ ········
0120  05 03 02 03 08 05 08 05  05 01 08 06 06 01 02 01   ········ ········
0130  00 12 00 00 00 33 00 2b  00 29 ca ca 00 01 00 00   ·····3·+ ·)······
0140  1d 00 20 51 4d 45 1b 47  a7 28 1b 55 f2 73 cd 0c   ·· QME·G ·(·U·s··
0150  35 9f 08 e8 31 10 d3 fd  be 78 1d e1 ad d0 ab 37   5···1··· ·x·····7
0160  e5 f1 26 00 2d 00 02 01  01 00 2b 00 0b 0a 6a 6a   ··&·-··· ··+···jj
0170  03 04 03 03 03 02 03 01  00 1b 00 03 02 00 01 3a   ········ ·······:
0180  3a 00 01 00 00 15 00 bf  00 00 00 00 00 00 00 00   :······· ········
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

Bytes 72-74: Length (tls.handshake.length)

☑ Show packet bytes

Figure 2: Header fields

Based on the provided Wireshark capture:

Table 1: Overview of Captured Packet

Attribute	Value
Frame Number	568
Bytes on Wire	583 (4664 bits)
Capture Interface	en0, id 0
Ethernet Source	Apple_77:80:65 (10:bd:3a:77:80:65)
Ethernet Destination	FidaInternat_a3:e8:ec (90:61:0c:a3:e8:ec)
Source IP	192.168.1.5
Destination IP	142.250.196.165
Protocol	TCP
Source Port	63488
Destination Port	443 (HTTPS)

3.2 Detailed IPv4 Header Analysis

The IPv4 header from the captured packet contains the following information:

Field	Value	Description
Version	4	IPv4
Header Length	20 bytes (5)	Standard IPv4 header length
Differentiated Services Field	0x00	DSCP: CS0, ECN: Not-ECT
Total Length	569	Total IP packet length
Identification	0x0000 (0)	Packet identifier
Flags	0x2	Don't fragment flag set
Fragment Offset	0	No fragmentation
Time to Live	64	Maximum hop count
Protocol	TCP (6)	Transport layer protocol
Header Checksum	0x2372	Validation disabled
Source Address	192.168.1.5	Sender's IP address
Destination Address	142.250.196.165	Recipient's IP address (Google server)

Table 2: IPv4 Header Fields

3.3 TCP Header Information

Field	Value
Source Port	63488
Destination Port	443
Sequence Number	1
Acknowledgment Number	1
Data Offset	8 bytes
Flags	SYN
Window Size	65535
Checksum	0x7e9f (unverified)
Urgent Pointer	0

Table 3: TCP Header Information

3.4 TLS Handshake Analysis

The packet capture reveals the initiation of a TLS handshake:

- **Client Hello:**
  - SNI (Server Name Indication): mail.google.com
  - TLS Version: TLS 1.2
  - Cipher Suites: [List of supported cipher suites]
- **Server Hello:**
  - Selected Cipher Suite: [Specific cipher suite]
  - TLS Version: TLS 1.2
- **Change Cipher Spec:** Indicates transition to encrypted communication

4 Discussion

4.1 HTTPS vs SMTP

The capture clearly shows the use of HTTPS (port 443) instead of SMTP (port 25). This is because:

- Gmail's web interface operates over HTTPS for security.
- SMTP is typically used for server-to-server email transfer or by email clients, not webmail interfaces.
- HTTPS provides end-to-end encryption, crucial for protecting user credentials and email content.

4.2 Inability to Use Telnet

Telnet could not be used in this scenario due to:

- Telnet's lack of encryption, contrasting with Gmail's secure HTTPS protocol.
- Gmail's servers not supporting unencrypted connections on port 23 (standard Telnet port).
- Modern web applications, especially those handling sensitive data, requiring secure protocols.

4.3 Encryption and Data Privacy

The use of HTTPS with TLS ensures:

- Encryption of all data transferred between the client and server.
- Protection against eavesdropping and man-in-the-middle attacks.
- Assurance that user credentials and email content remain confidential.

5 Conclusion

In conclusion, this report provided a detailed analysis of network traffic captured during the process of sending an email via Gmail's web interface. It highlighted the use of HTTPS for secure communication, the details of IPv4 and TCP headers, and the significance of encryption in protecting sensitive information. The report also included a step-by-step explanation of IPv4 header checksum calculation.