

Task 3: Network Traffic Analysis Report

Name: Mohammed Fadil PK Date: January 19, 2026

1. Introduction

Network analysis is the process of capturing and inspecting data packets as they move across a network. For this task, I used **Wireshark** to analyse TCP handshakes, DNS queries, and the difference between secure (HTTPS) and insecure (HTTP) traffic.

2. Observations & Analysis

A. The TCP 3-Way Handshake

- Concept:** Before data is sent, devices must establish a connection. This is called the "Handshake."
- Observation:** I filtered for tcp and observed the three steps:
 - SYN:** My computer sent a request to the server to connect.
 - SYN-ACK:** The server acknowledged and accepted the request.
 - ACK:** My computer confirmed the connection.

*Ethernet 4						
No.	Time	Source	Destination	Protocol	Length	Info
103	23.980076	10.116.64.221	142.251.43.131	TLSv1.3	93	Application Data
104	23.988329	10.116.64.221	142.251.43.131	TCP	54	55065 → 443 [FIN, ACK] Seq=2373 Ack=6028 Win=65280 Len=0
105	24.051386	142.251.43.131	10.116.64.221	TCP	66	[TCP Dup ACK 97#] 443 → 55065 [ACK] Seq=6028 Ack=2374 Win=268032 Len=0 SLE=2373 SRE=2374
106	24.051386	142.251.43.131	10.116.64.221	TCP	54	443 → 55065 [ACK] Seq=6028 Ack=2374 Win=268032 Len=0
107	24.051386	142.251.43.131	10.116.64.221	TCP	54	443 → 55065 [FIN, ACK] Seq=6028 Ack=2374 Win=268032 Len=0
108	24.051518	10.116.64.221	142.251.43.131	TCP	54	55065 → 443 [ACK] Seq=2374 Ack=6029 Win=65280 Len=0
120	25.608795	10.116.64.221	142.251.220.100	TCP	66	57277 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
124	25.614206	10.116.64.221	142.251.222.170	TCP	66	57252 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
127	25.624902	10.116.64.221	142.251.12.84	TCP	66	52750 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM

*Ethernet 4						
No.	Time	Source	Destination	Protocol	Length	Info
128	25.634944	142.251.220.100	10.116.64.221	TCP	66	443 → 57277 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 SACK_PERM WS=256
129	25.635116	10.116.64.221	142.251.220.100	TCP	54	57277 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
130	25.636896	10.116.64.221	142.251.220.100	TCP	1354	57277 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=1300 [TCP PDU reassembled in 131]
131	25.636896	10.116.64.221	142.251.220.100	TLSv1.3	510	Client Hello (SNI=www.google.com)
133	25.656439	142.251.222.170	10.116.64.221	TCP	66	443 → 57252 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 SACK_PERM WS=256
138	25.656610	10.116.64.221	142.251.222.170	TCP	54	57252 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
139	25.657206	10.116.64.221	142.251.222.170	TCP	1354	57252 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=1300 [TCP PDU reassembled in 140]
140	25.657206	10.116.64.221	142.251.222.170	TLSv1.3	530	Client Hello (SNI=oauthaccountmanager.googleapis.com)
155	25.668045	142.251.220.100	10.116.64.221	TCP	54	443 → 57277 [ACK] Seq=1 Ack=1301 Win=268544 Len=0
156	25.669111	142.251.220.100	10.116.64.221	TCP	54	443 → 57277 [ACK] Seq=1 Ack=1757 Win=268288 Len=0
169	25.701539	142.251.222.170	10.116.64.221	TCP	54	443 → 57252 [ACK] Seq=1 Ack=1301 Win=268544 Len=0
170	25.701539	142.251.222.170	10.116.64.221	TCP	54	443 → 57252 [ACK] Seq=1 Ack=1777 Win=268288 Len=0
171	25.701539	142.251.220.100	10.116.64.221	TLSv1.3	1334	Server Hello, Change Cipher Spec
172	25.701539	142.251.220.100	10.116.64.221	TCP	1334	443 → 57277 [PSH, ACK] Seq=1281 Ack=1757 Win=268288 Len=1280 [TCP PDU reassembled in 174]
173	25.701539	142.251.220.100	10.116.64.221	TCP	292	[TCP Previous segment not captured] 443 → 57277 [PSH, ACK] Seq=3841 Ack=1757 Win=268288 Len=0
174	25.701539	142.251.220.100	10.116.64.221	TLSv1.3	1334	[TCP Retransmission], Application Data
175	25.701539	142.251.12.84	10.116.64.221	TCP	66	443 → 52750 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 SACK_PERM WS=256

Fig 1: Wireshark capture showing the TCP 3-Way Handshake (SYN, SYN-ACK, ACK)

B. DNS Traffic Analysis

- Concept:** DNS (Domain Name System) converts human names (like <https://www.google.com/url?sa=E&source=gmail&q=google.com>) into IP addresses (like 142.250.206.78).
- Observation:** I filtered for dns and pinged a website. I captured the "Query" packet asking for the IP address and the "Response" packet providing it.

Destination	Protocol	Length Info
10.116.64.171	DNS	88 Standard query 0x9e4c A applet-bundles.grammarly.net
10.116.64.221	DNS	152 Standard query response 0x9e4c A applet-bundles.grammarly.net A 99.86.182.37 A 99.86.18:
10.116.64.171	DNS	75 Standard query 0xe3d7 HTTPS ssl.gstatic.com
10.116.64.171	DNS	75 Standard query 0xacbb A ssl.gstatic.com
10.116.64.221	DNS	132 Standard query response 0xe3d7 HTTPS ssl.gstatic.com SOA ns1.google.com
10.116.64.221	DNS	91 Standard query response 0xacbb A ssl.gstatic.com A 142.251.222.131
10.116.64.171	DNS	80 Standard query 0x6e29 HTTPS beacons.gcp.gvt2.com
10.116.64.171	DNS	80 Standard query 0x37ba A beacons.gcp.gvt2.com
10.116.64.221	DNS	126 Standard query response 0x37ba A beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.co
10.116.64.221	DNS	167 Standard query response 0x6e29 HTTPS beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt:
10.116.64.171	DNS	74 Standard query 0x55cc HTTPS www.google.com
10.116.64.171	DNS	74 Standard query 0x26e3 A www.google.com
10.116.64.221	DNS	90 Standard query response 0x26e3 A www.google.com A 142.251.220.100
10.116.64.221	DNS	99 Standard query response 0x55cc HTTPS www.google.com HTTPS
10.116.64.171	DNS	88 Standard query 0xe4e0 HTTPS applet-bundles.grammarly.net
10.116.64.171	DNS	88 Standard query 0x0455 A applet-bundles.grammarly.net
10.116.64.171	DNS	88 Standard query 0xf780 A applet-bundles.grammarly.net
10.116.64.221	DNS	173 Standard query response 0xe4e0 HTTPS applet-bundles.grammarly.net SOA ns-1253.awsdns-28
10.116.64.221	DNS	152 Standard query response 0x0455 A applet-bundles.grammarly.net A 99.86.182.44 A 99.86.18:
10.116.64.221	DNS	152 Standard query response 0xf780 A applet-bundles.grammarly.net A 99.86.182.44 A 99.86.18:
10.116.64.171	DNS	76 Standard query 0xe36b HTTPS us.i.posthog.com
10.116.64.171	DNS	76 Standard query 0x6b11 A us.i.posthog.com
10.116.64.221	DNS	230 Standard query response 0xe36b HTTPS us.i.posthog.com CNAME posthog-ingress-prod-us-256
10.116.64.221	DNS	276 Standard query response 0x6b11 A us.i.posthog.com CNAME posthog-ingress-prod-us-2564554:
10.116.64.171	DNS	70 Standard query 0x5c47 A google.com
10.116.64.171	DNS	70 Standard query 0x5c47 A google.com
10.116.64.221	DNS	86 Standard query response 0x5c47 A google.com A 142.250.206.78
10.116.64.171	DNS	96 Standard query 0x9f3a A win-extension.femetrics.grammarly.io
10.116.64.221	DNS	192 Standard query response 0x9f3a A win-extension.femetrics.grammarly.io A 98.86.251.215 A

Fig 2: DNS Traffic analysis showing the standard query and response for

<https://www.google.com/search?q=google.com>

C. HTTP vs. HTTPS (Security Check)

- HTTP (Port 80):** Traffic is sent in **plain text**. If I entered a password on an HTTP site, a hacker using Wireshark could read it easily.
- HTTPS (Port 443):** Traffic is **encrypted** using TLS/SSL. In Wireshark, the data looks like random scrambled characters ("Application Data"), making it secure from sniffing.

3. Interview Questions & Answers

1. **What is the TCP Handshake?** It is the process used to establish a reliable connection between a client and server. It involves three steps: SYN (Synchronise), SYN-ACK (Synchronise-Acknowledge), and ACK (Acknowledge).
2. **What is the difference between TCP and UDP?**
 - **TCP (Transmission Control Protocol):** Reliable. It checks if data arrived correctly (used for emails, websites).
 - **UDP (User Datagram Protocol):** Fast but unreliable. It sends data without checking if it arrived (used for video streaming, gaming).
3. **What is packet sniffing?** The act of capturing and monitoring data packets flowing across a network using tools like Wireshark. It can be used for debugging (by admins) or stealing data (by hackers).
4. **Why is HTTPS more secure than HTTP?** HTTPS uses encryption (TLS) to scramble data. Even if a hacker captures the packets, they cannot read the contents. HTTP sends data in plain text, which is easily readable.