Task 5: Capture and Analyze Network Traffic Using Wireshark

Objective

The objective of this task is to capture live network traffic using Wireshark, identify protocols in use, and analyze packet details.

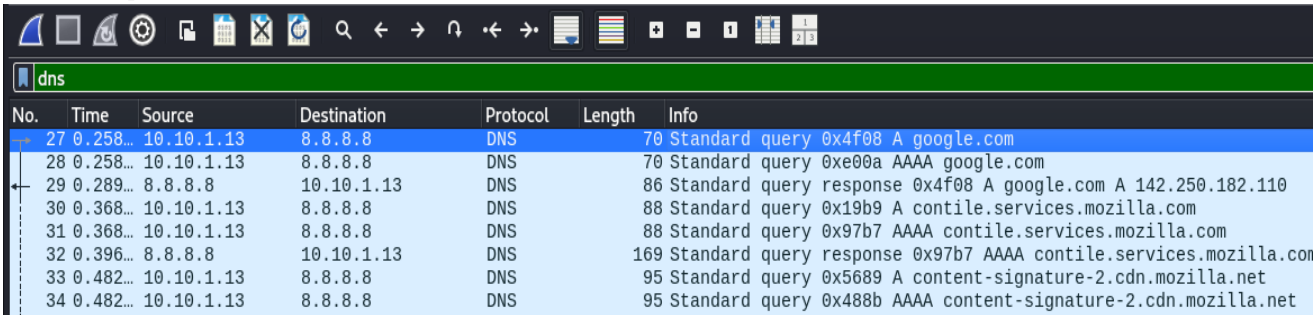
Steps Performed

1. Installed Wireshark on Windows.
2. Selected the active network interface (Wi-Fi in this case).
3. Started packet capture.
4. Opened a browser and accessed www.google.com.
5. Sent ping requests to 8.8.8.8 (Google DNS).
6. Stopped capture after ~1 minute.
7. Applied protocol filters to analyze traffic (e.g., dns, http, tcp, icmp).
8. Exported the capture as Task5_Network_Capture.pcap.

Protocols Identified

1. DNS (Domain Name System)

Used for resolving domain names (e.g., www.google.com) into IP addresses.
Example:
Source: 10.10.1.13 → Destination: 8.8.8.8
Query: A www.google.com
Response: 142.250.182.110



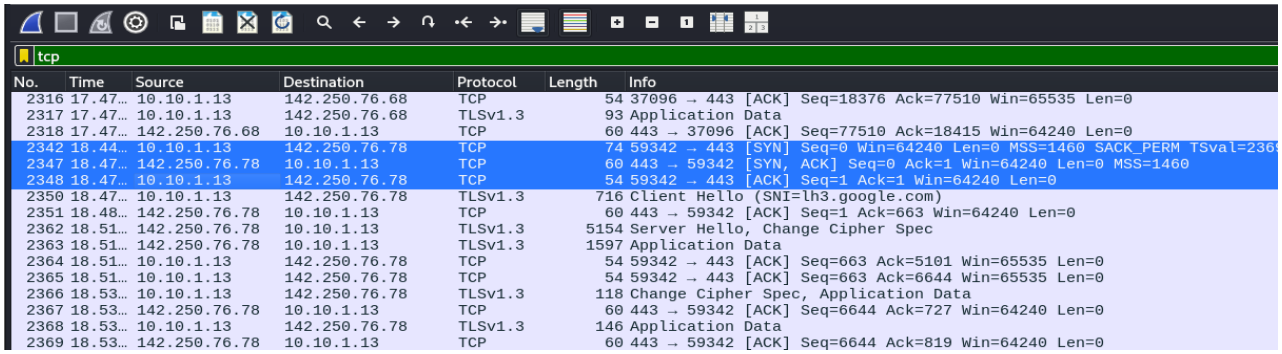
dns

No.	Time	Source	Destination	Protocol	Length	Info
27	0.258…	10.10.1.13	8.8.8.8	DNS	70	Standard query 0x4f08 A google.com
28	0.258…	10.10.1.13	8.8.8.8	DNS	70	Standard query 0xe00a AAAA google.com
29	0.289…	8.8.8.8	10.10.1.13	DNS	86	Standard query response 0x4f08 A google.com A 142.250.182.110
30	0.368…	10.10.1.13	8.8.8.8	DNS	88	Standard query 0x19b9 A contile.services.mozilla.com
31	0.368…	10.10.1.13	8.8.8.8	DNS	88	Standard query 0x97b7 AAAA contile.services.mozilla.com
32	0.396…	8.8.8.8	10.10.1.13	DNS	169	Standard query response 0x97b7 AAAA contile.services.mozilla.com
33	0.482…	10.10.1.13	8.8.8.8	DNS	95	Standard query 0x5689 A content-signature-2.cdn.mozilla.net
34	0.482…	10.10.1.13	8.8.8.8	DNS	95	Standard query 0x488b AAAA content-signature-2.cdn.mozilla.net

2. TCP (Transmission Control Protocol)

Used to establish reliable connections between client and server.

Example:



tcp

No.	Time	Source	Destination	Protocol	Length	Info
2316	17.47…	10.10.1.13	142.250.76.68	TCP	54	37096 → 443 [ACK] Seq=18376 Ack=77510 Win=65535 Len=0
2317	17.47…	10.10.1.13	142.250.76.68	TLSv1.3	93	Application Data
2318	17.47…	142.250.76.68	10.10.1.13	TCP	60	443 → 37096 [ACK] Seq=77510 Ack=18415 Win=64240 Len=0
2342	18.44…	10.10.1.13	142.250.76.78	TCP	74	59342 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=236
2347	18.47…	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2348	18.47…	10.10.1.13	142.250.76.78	TCP	54	59342 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2350	18.47…	10.10.1.13	142.250.76.78	TLSv1.3	716	Client Hello (SNI=lh3.google.com)
2351	18.48…	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [ACK] Seq=1 Ack=663 Win=64240 Len=0
2362	18.51…	142.250.76.78	10.10.1.13	TLSv1.3	5154	Server Hello, Change Cipher Spec
2363	18.51…	142.250.76.78	10.10.1.13	TLSv1.3	1597	Application Data
2364	18.51…	10.10.1.13	142.250.76.78	TCP	54	59342 → 443 [ACK] Seq=663 Ack=5101 Win=65535 Len=0
2365	18.51…	10.10.1.13	142.250.76.78	TCP	54	59342 → 443 [ACK] Seq=663 Ack=6644 Win=65535 Len=0
2366	18.53…	10.10.1.13	142.250.76.78	TLSv1.3	118	Change Cipher Spec, Application Data
2367	18.53…	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [ACK] Seq=6644 Ack=727 Win=64240 Len=0
2368	18.53…	10.10.1.13	142.250.76.78	TLSv1.3	146	Application Data
2369	18.53…	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [ACK] Seq=6644 Ack=819 Win=64240 Len=0

Source: 10.10.1.13 → Destination: 142.250.76.78

Flags: SYN → SYN/ACK → ACK (3-way handshake)

3. HTTP (Hypertext Transfer Protocol)

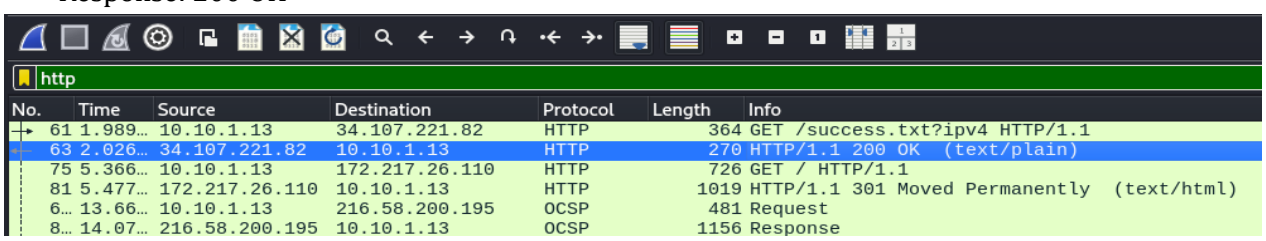
Application layer protocol used for web requests.

Example:

GET / HTTP/1.1

Host: www.google.com

Response: 200 OK



http

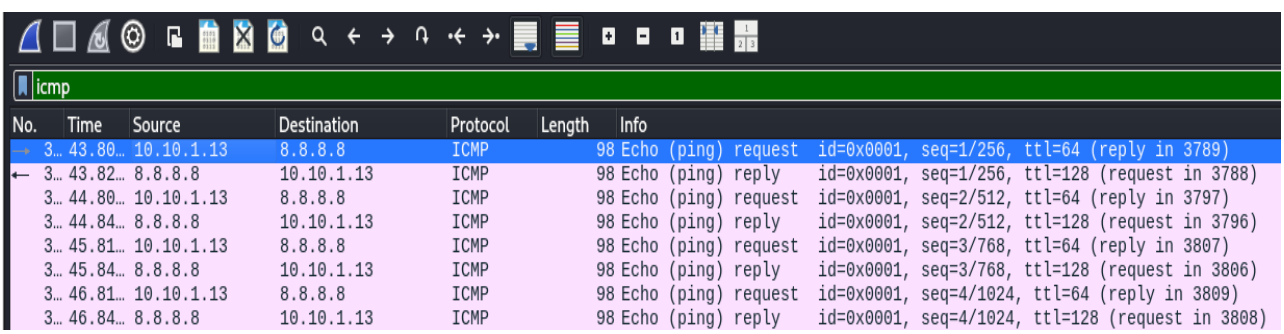
No.	Time	Source	Destination	Protocol	Length	Info
61	1.989…	10.10.1.13	34.107.221.82	HTTP	364	GET /success.txt?ipv4 HTTP/1.1
63	2.026…	34.107.221.82	10.10.1.13	HTTP	270	HTTP/1.1 200 OK (text/plain)
75	5.366…	10.10.1.13	172.217.26.110	HTTP	726	GET / HTTP/1.1
81	5.477…	172.217.26.110	10.10.1.13	HTTP	1019	HTTP/1.1 301 Moved Permanently (text/html)
6…	13.66…	10.10.1.13	216.58.200.195	OCSP	481	Request
8…	14.07…	216.58.200.195	10.10.1.13	OCSP	1156	Response

4. ICMP (Internet Control Message Protocol)

Used for ping and diagnostic messages.

Example:

Echo (ping) request → Echo reply from 8.8.8.8



icmp

No.	Time	Source	Destination	Protocol	Length	Info
3…	43.80…	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 3789)
3…	43.82…	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=1/256, ttl=128 (request in 3788)
3…	44.80…	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 3797)
3…	44.84…	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=2/512, ttl=128 (request in 3796)
3…	45.81…	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 3807)
3…	45.84…	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=3/768, ttl=128 (request in 3806)
3…	46.81…	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 3809)
3…	46.84…	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=4/1024, ttl=128 (request in 3808)

Summary of Findings

At least 4 different protocols were observed: DNS, TCP, HTTP, ICMP.

- DNS queries resolved domains successfully.
- TCP established sessions with Google servers.
- HTTP traffic showed web requests/responses.
- ICMP confirmed network connectivity.