

Task 5 – Capture and Analyze Network Traffic Using Wireshark

1. Objective

The objective of this task was to capture live network packets using Wireshark and analyze them to identify basic protocols (DNS, HTTP, TCP) and their traffic characteristics.

2. Tools Used

- Wireshark (Network Protocol Analyzer)
- Operating System: [Your OS, e.g., Kali Linux/Windows]

3. Methodology

1. Installed Wireshark and started capture on active interface.
2. Generated traffic by browsing websites and pinging servers.
3. Stopped capture after 1–2 minutes.
4. Applied filters (dns, http, tcp) to analyze specific protocols.
5. Saved capture as wireshark_capture.pcap.
6. Exported screenshots of findings.

4. Findings

a) DNS (Domain Name System)

DNS queries and responses were captured.

Example: Query for contile.services.mozilla.com.

This shows how the system resolve domain names into IP addresses.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.005588551	192.168.1.11	192.168.1.1	DNS	68	Standard query 0x50f9 A contile.services.mozilla.com
4	0.00700217	192.168.1.1	192.168.1.11	DNS	68	Standard query response 0x50f9 A contile.services.mozilla.com
5	0.007587559	192.168.1.11	192.168.1.1	DNS	95	Standard query 0xc746 A content-signature-2.cdn.mozilla.net
6	0.008031348	192.168.1.11	192.168.1.1	DNS	95	Standard query 0xc645 AAAA content-signature-2.cdn.mozilla.net
7	0.008171555	192.168.1.11	192.168.1.1	DNS	161	Standard query response 0xc746 A content-signature-2.cdn.mozilla.net CNAME content-signature-chains.prod.autograph.services.mo.
8	0.009172818	192.168.1.11	192.168.1.1	DNS	169	Standard query response 0x74f1 AAAA contile.services.mozilla.com SOA ns-679.awmdns-20.net
9	0.009173459	192.168.1.11	192.168.1.1	DNS	164	Standard query response 0x50f9 A contile.services.mozilla.com A 34.36.137.203
10	0.009173876	192.168.1.11	192.168.1.1	DNS	193	Standard query response 0xc645 AAAA content-signature-2.cdn.mozilla.net CNAME content-signature-chains.prod.autograph.services.
26	0.175528854	192.168.1.11	192.168.1.1	DNS	87	Standard query 0x99ed A safefrowsing.googleapis.com
27	0.175705786	192.168.1.11	192.168.1.1	DNS	87	Standard query 0x3ff2 AAAA safefrowsing.googleapis.com
34	0.281262485	192.168.1.11	192.168.1.1	DNS	163	Standard query response 0x99ed A safefrowsing.googleapis.com A 142.250.183.74
39	0.291262885	192.168.1.11	192.168.1.1	DNS	115	Standard query response 0x3ff2 AAAA safefrowsing.googleapis.com AAAA 2404:6800:4009:822::290a
76	0.298285827	192.168.1.11	192.168.1.1	DNS	70	Standard query 0x2b73 A o.pki.goog
77	0.298948559	192.168.1.11	192.168.1.1	DNS	70	Standard query 0x5e71 AAAA o.pki.goog
78	0.324228732	192.168.1.11	192.168.1.1	DNS	121	Standard query response 0x2b73 A o.pki.goog CNAME pki-goog.l.google.com A 142.251.42.227
79	0.324229148	192.168.1.11	192.168.1.1	DNS	133	Standard query response 0x5e71 AAAA o.pki.goog CNAME pki-goog.l.google.com AAAA 2404:6800:4009:800::2903
84	0.351389529	192.168.1.11	192.168.1.1	DNS	85	Standard query 0xc0cd A push.services.mozilla.com
85	0.507118843	192.168.1.11	192.168.1.1	DNS	85	Standard query 0x35d8 AAAA push.services.mozilla.com
91	0.576623484	192.168.1.11	192.168.1.1	DNS	101	Standard query response 0xc0cd A push.services.mozilla.com A 34.107.243.93
103	0.581938828	192.168.1.11	192.168.1.1	DNS	166	Standard query response 0x35d8 AAAA push.services.mozilla.com SOA ns-679.awmdns-20.net
104	0.774326216	192.168.1.11	192.168.1.1	DNS	97	Standard query 0xc0cd A Firefox.settings.services.mozilla.com
105	0.774639889	192.168.1.11	192.168.1.1	DNS	97	Standard query 0x5d6f AAAA Firefox.settings.services.mozilla.com
196	0.882275486	192.168.1.11	192.168.1.1	DNS	197	Standard query response 0xc0cd A Firefox.settings.services.mozilla.com CNAME mozilla.map.fastly.net A 151.101.129.01 A 151.101.
197	0.882275712	192.168.1.11	192.168.1.1	DNS	240	Standard query response 0x5d6f AAAA Firefox.settings.services.mozilla.com CNAME mozilla.map.fastly.net AAAA 2a04:ac42:0000::347.
373	1.187796142	192.168.1.11	192.168.1.1	DNS	71	Standard query 0x0b79 A example.org
388	1.111870319	192.168.1.11	192.168.1.1	DNS	71	Standard query 0x0b33 A example.org
389	1.111512222	192.168.1.11	192.168.1.1	DNS	493	Standard query response 0x0b33 A example.org
395	1.141693202	192.168.1.11	192.168.1.1	DNS	73	Standard query 0x8f81 A ipv4only.arpa
396	1.141875834	192.168.1.11	192.168.1.1	DNS	73	Standard query 0x57ae AAAA ipv4only.arpa
397	1.142711866	192.168.1.11	192.168.1.1	DNS	84	Standard query 0x1862 A detectportal.firefox.com
414	1.144648487	192.168.1.11	192.168.1.1	DNS	135	Standard query response 0x0b79 A example.org A 23.215.0.132 A 23.229.75.238 A 23.215.0.133 A 23.229.75.235
420	1.145151812	192.168.1.11	192.168.1.1	DNS	84	Standard query 0x5b5c A detectportal.firefox.com

b) HTTP Hypertext Transfer Protocol)

HTTP GET requests and responses were observed.

Example: GET /success.txt?ipv4 HTTP/1.1 request, responded with 200 OK.

This confirms successful client-server communication.

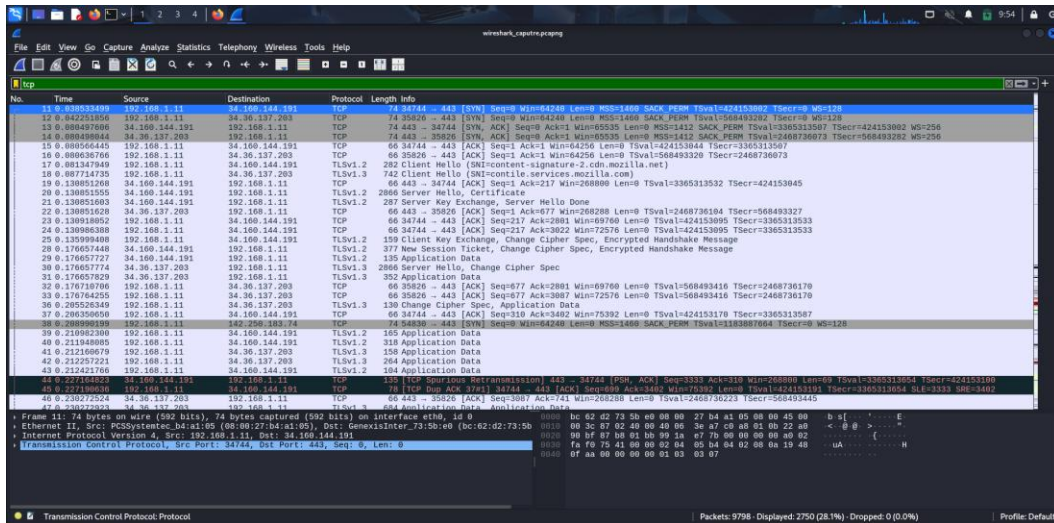
No.	Time	Source	Destination	Protocol	Length	Info
89	0.508656647	142.251.42.227	192.168.1.11	OSPF	493	Request
437	1.186218545	192.168.1.11	192.168.1.1	HTTP	376	GET /success.txt?ipv4 HTTP/1.1
441	1.238893685	34.107.221.82	192.168.1.11	HTTP	282	HTTP/1.1 200 OK (text/plain)
457	1.562377853	192.168.1.11	192.168.1.1	HTTP	376	GET /success.txt?ipv4 HTTP/1.1
461	1.584859954	34.107.221.82	192.168.1.11	HTTP	282	HTTP/1.1 200 OK (text/plain)
508	10.518099538	192.168.1.11	192.168.1.1	HTTP	739	GET / HTTP/1.1
592	10.913137593	142.250.70.110	192.168.1.11	HTTP	1031	HTTP/1.1 304 Not Modified (text/html)
626	11.113869088	192.168.1.11	192.168.1.1	OSPF	493	Request
632	11.172489787	192.168.1.11	192.168.1.1	OSPF	494	Request
636	11.197747219	142.251.221.227	192.168.1.11	OSPF	1169	Response
640	11.258691891	142.251.221.227	192.168.1.11	OSPF	1169	Response
1021	11.894833558	192.168.1.11	192.168.1.1	OSPF	494	Request
1059	11.928896275	192.168.1.11	192.168.1.1	OSPF	494	Request
1088	11.999664802	192.168.1.11	192.168.1.1	OSPF	493	Request
1188	12.011435887	142.251.221.227	192.168.1.11	OSPF	1169	Response
1112	12.012644780	192.168.1.11	192.168.1.1	OSPF	494	Request
1113	12.020133111	142.251.221.227	192.168.1.11	OSPF	1169	Response
1188	12.154769373	192.168.1.11	192.168.1.1	OSPF	493	Request
1233	12.176688934	142.251.221.227	192.168.1.11	OSPF	1169	Response
1248	12.311472821	192.168.1.11	192.168.1.1	OSPF	493	Request
1253	12.336647888	192.168.1.11	192.168.1.1	OSPF	494	Request
1261	12.352618313	142.251.221.227	192.168.1.11	OSPF	1169	Response
1274	12.378451111	142.251.221.227	192.168.1.11	OSPF	1169	Response
1288	12.402482509	142.251.221.227	192.168.1.11	OSPF	1169	Response
1333	12.434926831	142.251.221.227	192.168.1.11	OSPF	1169	Response
1698	13.007533354	192.168.1.11	192.168.1.1	OSPF	494	Request
1691	13.018728797	192.168.1.11	192.168.1.1	OSPF	494	Request
1692	13.018905342	192.168.1.11	192.168.1.1	OSPF	494	Request
1700	13.127606262	142.251.221.227	192.168.1.11	OSPF	1169	Response
1701	13.127606297	142.251.221.227	192.168.1.11	OSPF	1169	Response
1702	13.127606322	142.251.221.227	192.168.1.11	OSPF	1169	Response

c) TCP (Transmission Control Protocol)

TCP handshake packets (SYN, SYN-ACK, ACK) were captured.

Example: Connection to 34.160.144.191 using port 443 (HTTPS).

Also observed TLS handshake packets, showing encrypted traffic.



5. Conclusion

Through this task, I gained practical experience in:

- Capturing live packets with Wireshark.
- Applying filters to focus on specific protocols.
- Understanding how DNS resolves domains, how HTTP transmits data, and how TCP establishes connections.

This strengthened my protocol analysis and troubleshooting skills.