

# Task 5 : Capture and Analyze Network Traffic Using Wireshark

**Objective:** Capture live network packets and identify basic protocols and traffic types.

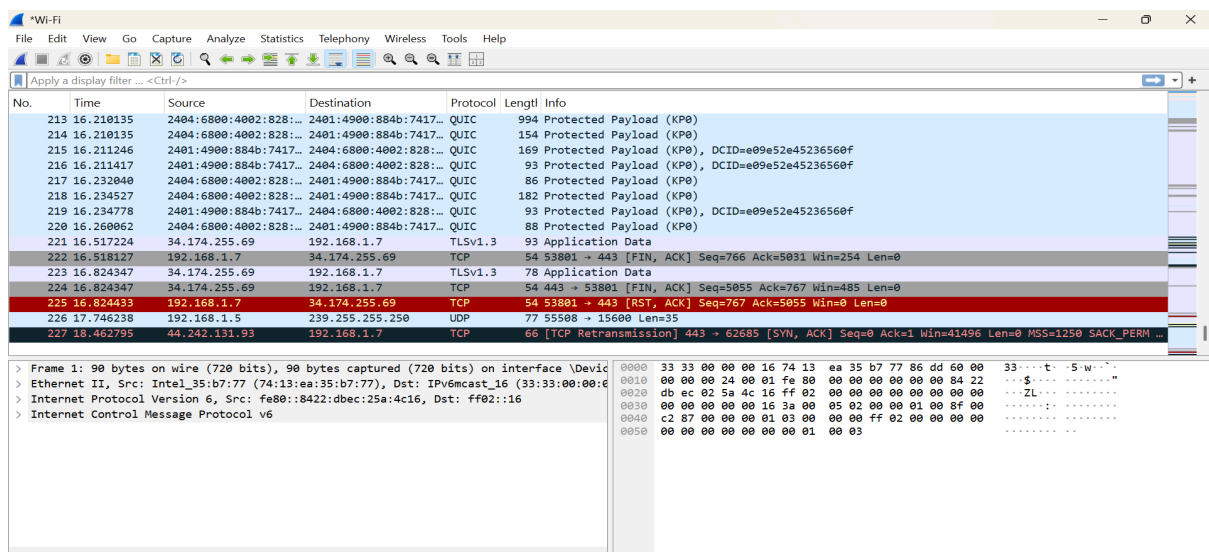
**Tools:** Wireshark (free).

**Deliverables:** A packet capture (.pcap) file and a short report of protocols identified.

Step 1 — Visited a website named wikibook to generate traffic



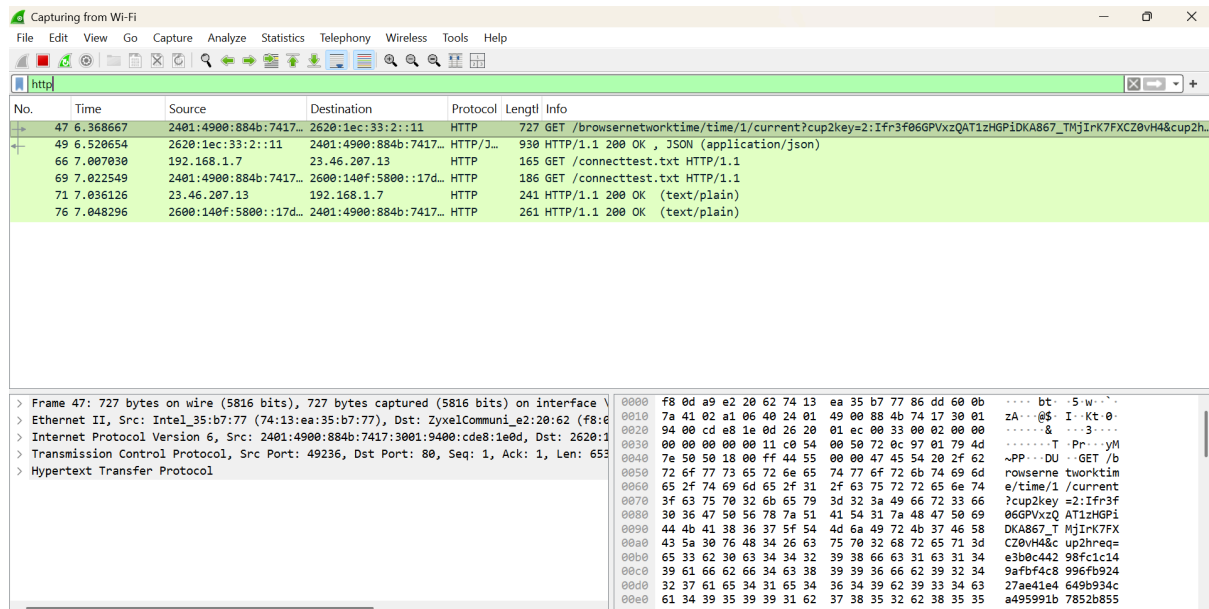
Step 2 :- Wireshark is installed in my system so I started packet capturing



Three different protocols have been observed in this generic packet capture:

- TCP
- UDP
- QUIC

After applying HTTP filter



## 1. HTTP Traffic Findings

The HTTP-filtered packets clearly show **unencrypted HTTP communication**, allowing full visibility into requests and responses.

**Observed details:**

- HTTP **GET requests** were observed, including:
  - Requests to **/connecttest.txt**
  - Requests to **/browsernetworktime/time/1/current**
- HTTP responses included:
  - **HTTP/1.1 200 OK**
  - Content types such as **text/plain** and **application/json**
- HTTP traffic was exchanged over **TCP port 80**

- Packet payloads were readable, highlighting the lack of encryption in HTTP traffic.

### Inference:

HTTP traffic confirms successful application-layer communication and demonstrates why HTTP is considered insecure compared to HTTPS.

After applying TCP filter

The screenshot shows a Wireshark capture of network traffic with a TCP filter applied. The packet list displays several TCP packets, including a three-way handshake and data transmission. The packet details pane for packet 42 shows an HTTP GET request.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2401:4900:884b:7417::...	2001:df2:e500:ed1a::...	TCP	75	49879 → 443 [ACK] Seq=1 Ack=1 Win=255 Len=1
2	0.494558	2001:df2:e500:ed1a::...	2401:4900:884b:7417::...	TCP	86	443 → 49879 [ACK] Seq=1 Ack=2 Win=83 Len=0 SLE=1 SRE=2
11	1.623896	2a03:2880:f349:121::...	2401:4900:884b:7417::...	SSL	113	Continuation Data
12	1.628511	2401:4900:884b:7417::...	2a03:2880:f349:121::...	SSL	103	Continuation Data
13	1.656933	2a03:2880:f349:121::...	2401:4900:884b:7417::...	TCP	74	443 → 59444 [ACK] Seq=40 Ack=30 Win=330 Len=0
31	5.012489	34.174.255.69	192.168.1.7	TCP	54	443 → 50495 [ACK] Seq=1 Ack=1 Win=489 Len=0
32	5.012550	192.168.1.7	34.174.255.69	TCP	54	[TCP ACKed unseen segment] 50495 → 443 [ACK] Seq=1 Ack=2 Win=255 Len=0
42	6.282803	2401:4900:884b:7417::...	2620:1ec:33:2::11	TCP	86	49236 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1420 WS=256 SACK_PERM
45	6.367552	2620:1ec:33:2::11	2401:4900:884b:7417::...	TCP	86	80 → 49236 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 WS=256 SACK_PERM
46	6.367656	2401:4900:884b:7417::...	2620:1ec:33:2::11	TCP	74	49236 → 80 [ACK] Seq=1 Ack=1 Win=65200 Len=0
47	6.368667	2401:4900:884b:7417::...	2620:1ec:33:2::11	HTTP	727	GET /browsernetworktime/time/1/current?cup2key=2:If3f066PVxzQAT1zHGPIDKA867_THjIrK7FXCZ0VH4&cu...
48	6.455800	2620:1ec:33:2::11	2401:4900:884b:7417::...	TCP	74	80 → 49236 [ACK] Seq=1 Ack=654 Win=4194816 Len=0
49	6.520654	2620:1ec:33:2::11	2401:4900:884b:7417::...	HTTP/JSON	930	HTTP/1.1 200 OK, JSON (application/json)
50	6.544611	2401:4900:884b:7417::...	2620:1ec:33:2::11	TCP	74	49236 → 80 [FIN, ACK] Seq=654 Ack=857 Win=64512 Len=0
51	6.634966	2620:1ec:33:2::11	2401:4900:884b:7417::...	TCP	74	80 → 49236 [ACK] Seq=857 Ack=655 Win=4194816 Len=0
52	6.638679	2620:1ec:33:2::11	2401:4900:884b:7417::...	TCP	74	80 → 49236 [FIN, ACK] Seq=857 Ack=655 Win=4194816 Len=0

Frame 42: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF{...} Ethernet II, Src: Intel\_35:b7:77 (74:13:ea:35:b7:77), Dst: ZyxelCommuni\_e2:20:62 (f8:6d:8c:20:62:f8:6d) Internet Protocol Version 6, Src: 2401:4900:884b:7417:3001:9400:cde8:1e0d, Dst: 2620:1ec:33:2::11 Transmission Control Protocol, Src Port: 49236, Dst Port: 80, Seq: 0, Len: 0

## 2. TCP Traffic Findings

The TCP-filtered packets reveal both **HTTP** and **HTTPS-related TCP sessions**, showing reliable, connection-oriented communication.

### Observed details:

- TCP **three-way handshake** was clearly visible:
  - SYN → SYN, ACK → ACK
- TCP connections were established on:
  - **Port 80** for HTTP traffic
  - **Port 443** for encrypted (TLS/SSL) traffic

- TCP session lifecycle was observed:
  - Connection establishment
  - Data transfer
  - Graceful connection termination using **FIN**, **ACK**
- An instance of “**TCP ACKed unseen segment**” was observed, which can occur due to:
  - Packet loss
  - Capture starting mid-session
  - Out-of-order packet arrival

### Inference:

TCP ensured reliable data delivery and session management for application-layer protocols such as HTTP and HTTPS.

After applying dns filter

The screenshot shows the Wireshark interface with the DNS filter applied. The packet list displays several DNS packets, including queries and responses. The packet details pane shows the structure of a DNS response packet, including the header, question, answer, and authority sections. The packet bytes pane shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
36	6.242222	2401:4900:884b:7417...	2401:4900:50:9::7c1	DNS	98	Standard query 0x65fb A edge.microsoft.com
37	6.265471	2401:4900:884b:7417...	2401:4900:50:9::7c1	DNS	96	Standard query 0xaba3 A wpad.domain.name
38	6.266213	2401:4900:884b:7417...	2401:4900:50:9::7c1	DNS	96	Standard query 0x43b1 AAAA wpad.domain.name
39	6.278551	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	219	Standard query response 0x65fb A edge.microsoft.com CNAME edge-microsoft-com.ax-0002.ax-msedge...
40	6.280263	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	243	Standard query response 0xd247 AAAA edge.microsoft.com CNAME edge-microsoft-com.ax-0002.ax-msed...
41	6.281525	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	234	Standard query response 0x7577 HTTPS edge.microsoft.com CNAME edge-microsoft-com.ax-0002.ax-mse...
43	6.295867	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	163	Standard query response 0xaba3 No such name A wpad.domain.name SOA ac1.nstld.com
44	6.299620	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	163	Standard query response 0x43b1 No such name AAAA wpad.domain.name SOA ac1.nstld.com
54	6.941837	2401:4900:884b:7417...	2401:4900:50:9::7c1	DNS	103	Standard query 0x1056 A www.msftconnecttest.com
55	6.942034	2401:4900:884b:7417...	2401:4900:50:9::7c1	DNS	103	Standard query 0x6578 AAAA www.msftconnecttest.com
56	6.961719	2401:4900:884b:7417...	2401:4900:50:9::7c1	DNS	104	Standard query 0xf408 A ipv6.msftconnecttest.com
57	6.961945	2401:4900:884b:7417...	2401:4900:50:9::7c1	DNS	104	Standard query 0x76fc AAAA ipv6.msftconnecttest.com
58	6.975035	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	274	Standard query response 0x6578 AAAA www.msftconnecttest.com CNAME ncsi-geo.trafficmanager.net C...
59	6.979566	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	359	Standard query response 0x1056 A www.msftconnecttest.com CNAME ncsi-geo.trafficmanager.net CNAM...
61	6.993662	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	289	Standard query response 0xf408 A ipv6.msftconnecttest.com CNAME ncsiv6-geo.trafficmanager.net C...
62	6.996178	2401:4900:50:9::7c1	2401:4900:884b:7417...	DNS	284	Standard query response 0x76fc AAAA ipv6.msftconnecttest.com CNAME ncsiv6-geo.trafficmanager.ne...

Frame 44: 163 bytes on wire (1304 bits), 163 bytes captured (1304 bits) on interface V...  
 Ethernet II, Src: ZyxelCommuni\_e2:20:62 (f8:0d:a9:e2:20:62), Dst: Intel\_35:b7:77 (74:1...  
 Internet Protocol Version 6, Src: 2401:4900:50:9::7c1, Dst: 2401:4900:884b:7417:3001:5...  
 User Datagram Protocol, Src Port: 53, Dst Port: 51217  
 Domain Name System (response)

Domain Name System: Protocol

Packets: 436 · Displayed: 72 (16.5%)

Profile: Default

## 3. DNS Traffic Findings

The DNS-filtered view shows multiple **standard DNS queries and responses**, primarily related to Microsoft connectivity and service checks.

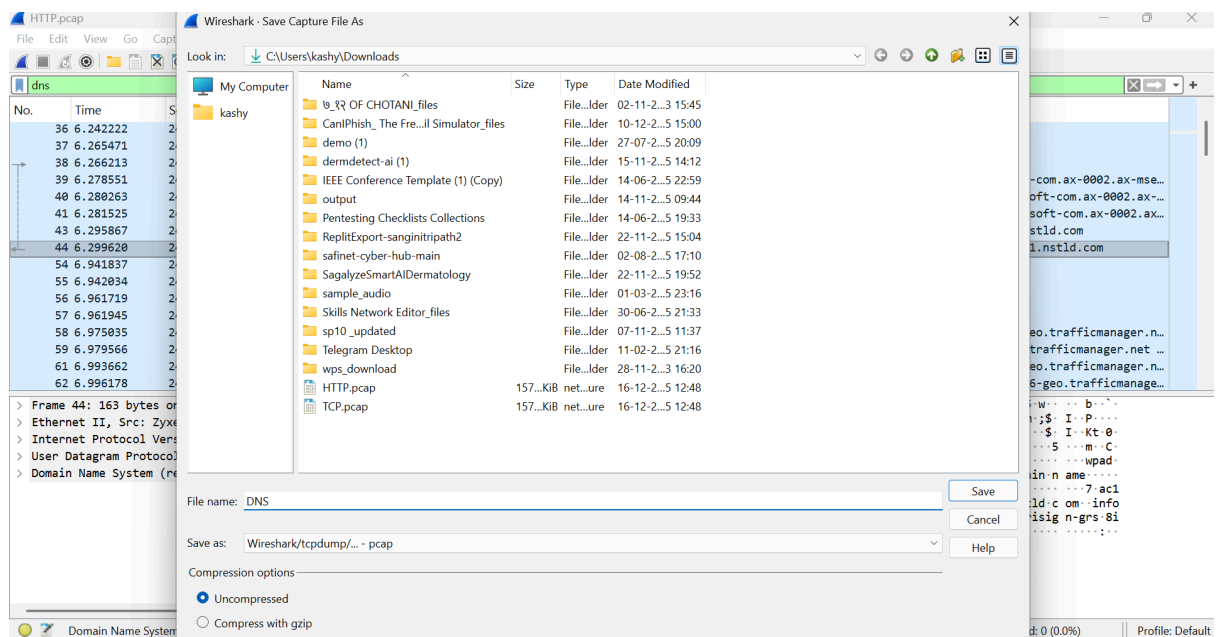
## Observed details:

- DNS queries for domains such as:
  - `edge.microsoft.com`
  - `www.msftconnecttest.com`
  - `ipv6.msftconnecttest.com`
  - `wpad.domain.name`
- Both **A (IPv4)** and **AAAA (IPv6)** record queries were observed.
- DNS responses include:
  - Successful resolutions using **CNAME records** (e.g., traffic manager domains)
  - **No such name** responses for `wpad.domain.name`, which is a common and normal behavior when WPAD is not configured.
- DNS communication occurred over **UDP port 53**, confirming standard DNS resolution behavior.

## Inference:

DNS traffic confirms that the system was actively resolving domain names before initiating HTTP/TCP connections, which is the first step in web communication

And Lastly all the files were saved as .pcap file.



## Summary —

### Overall Analysis and Conclusion

- The captured traffic accurately demonstrates the **end-to-end flow of network communication**:
  1. **DNS** resolves domain names to IP addresses
  2. **TCP** establishes reliable connections
  3. **HTTP** transfers application data
- Both **IPv4 and IPv6 traffic** were observed, indicating a dual-stack network environment.
- The traffic captured represents **normal, legitimate system and browser activity**, including connectivity checks and web requests.
- Wireshark filtering proved effective in isolating protocols and analyzing packet behavior.