

## **Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark**

### **5<sup>th</sup> Task SUBMISSION REPORT OF ELEVATE LABS CYBERSECURITY INTERNSHIP**

<b>NAME</b>	<b>ADARSH SHARMA</b>
<b>Submitted to:</b>	<b>Elevate Labs</b>
<b>Name of the Academic Institute</b>	<b>Ganpat University</b>

## **REPORT SUBMITTED TO**

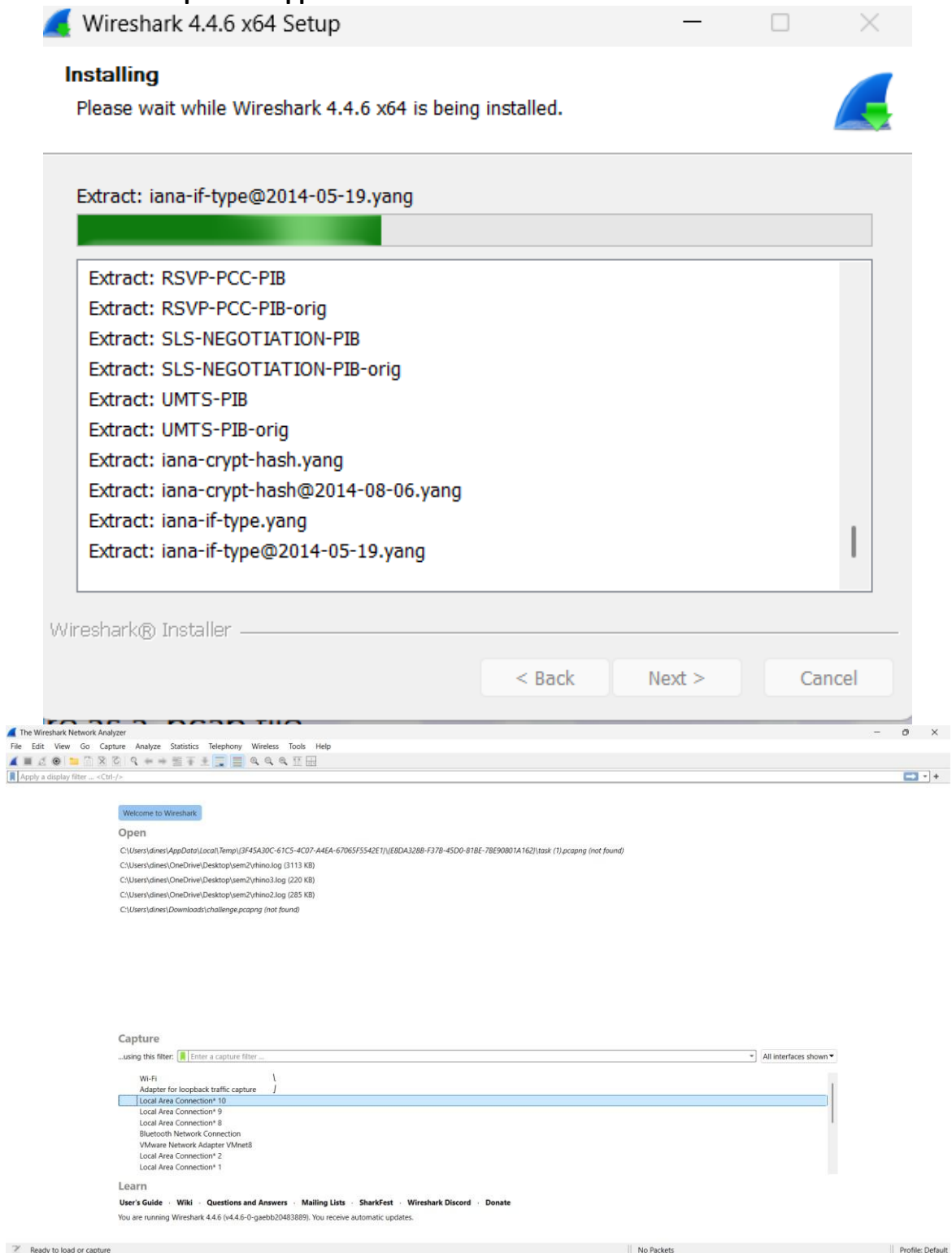


**As part of the Cyber Security Internship, I have completed "Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark." by following all steps as instructed. Below is a detailed summary of each step followed during the task:**

## Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark

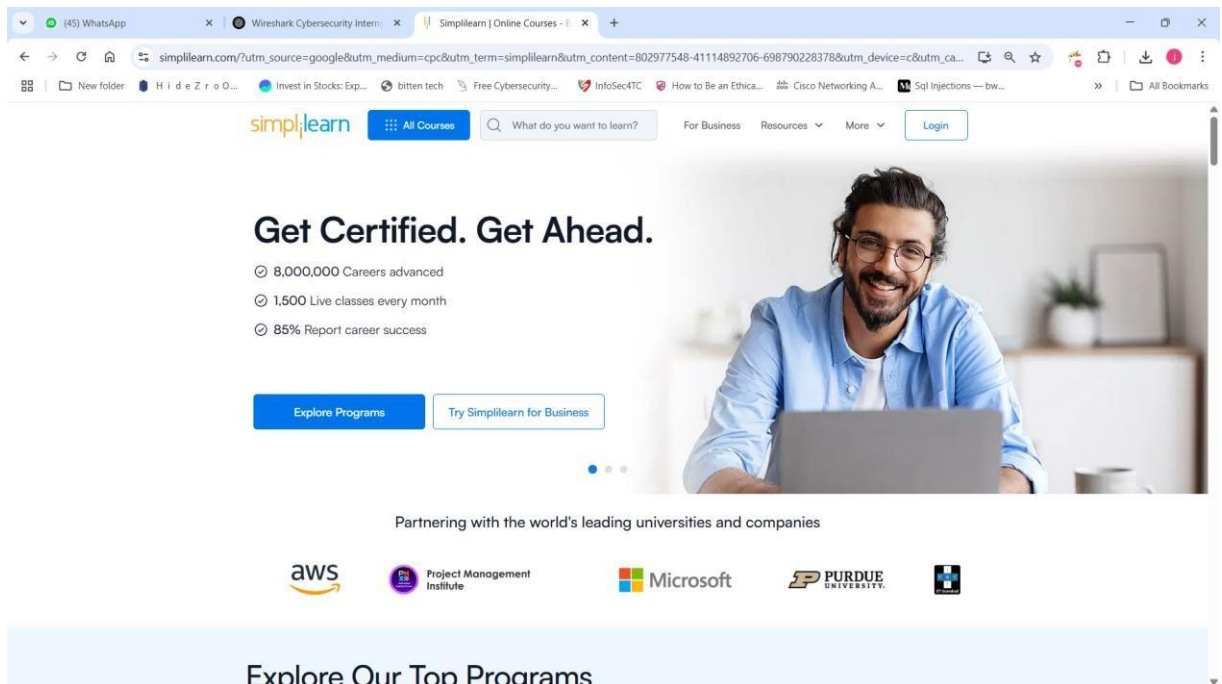
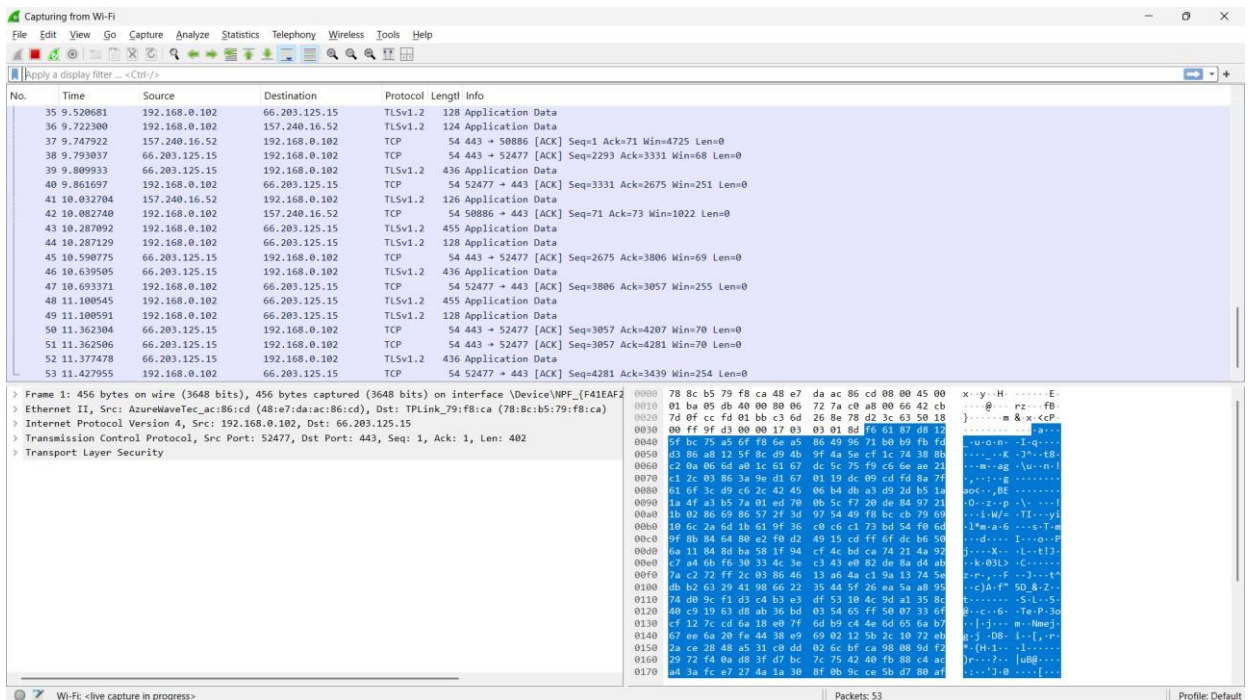
### 1. Install Wireshark.

- Download from: <https://www.wireshark.org/download.html>
- Install and open the application.

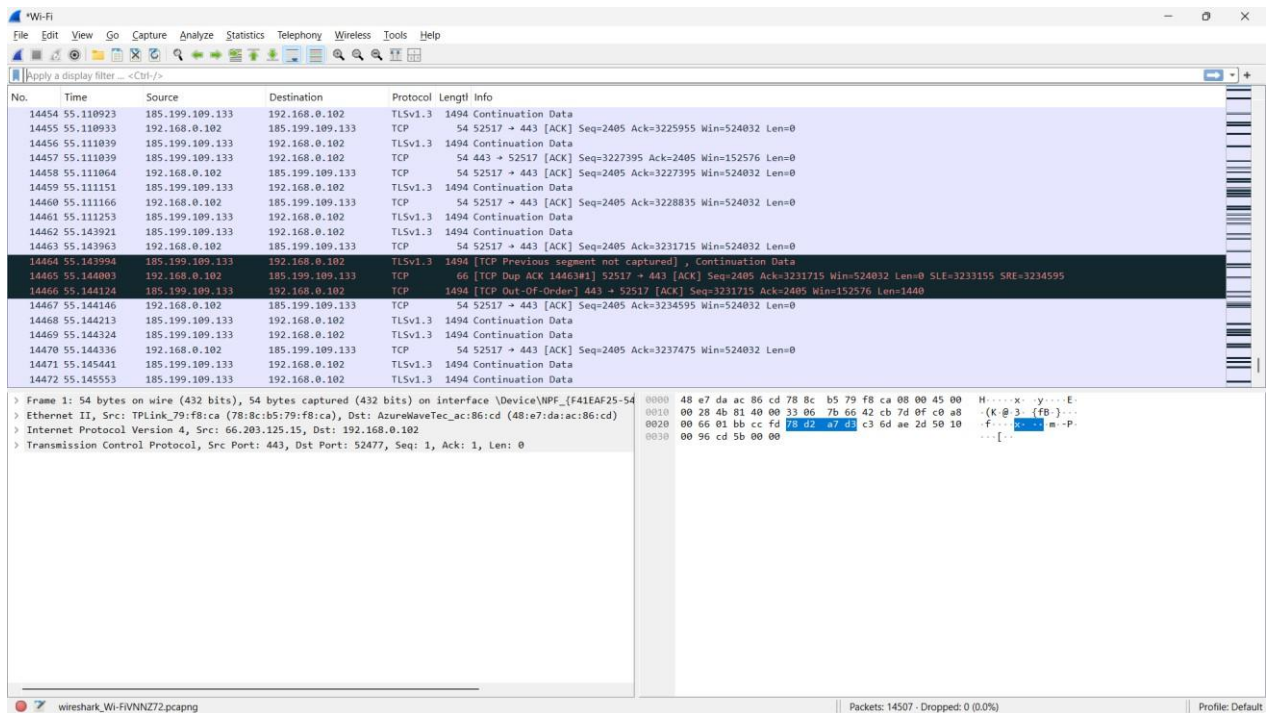


## Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark

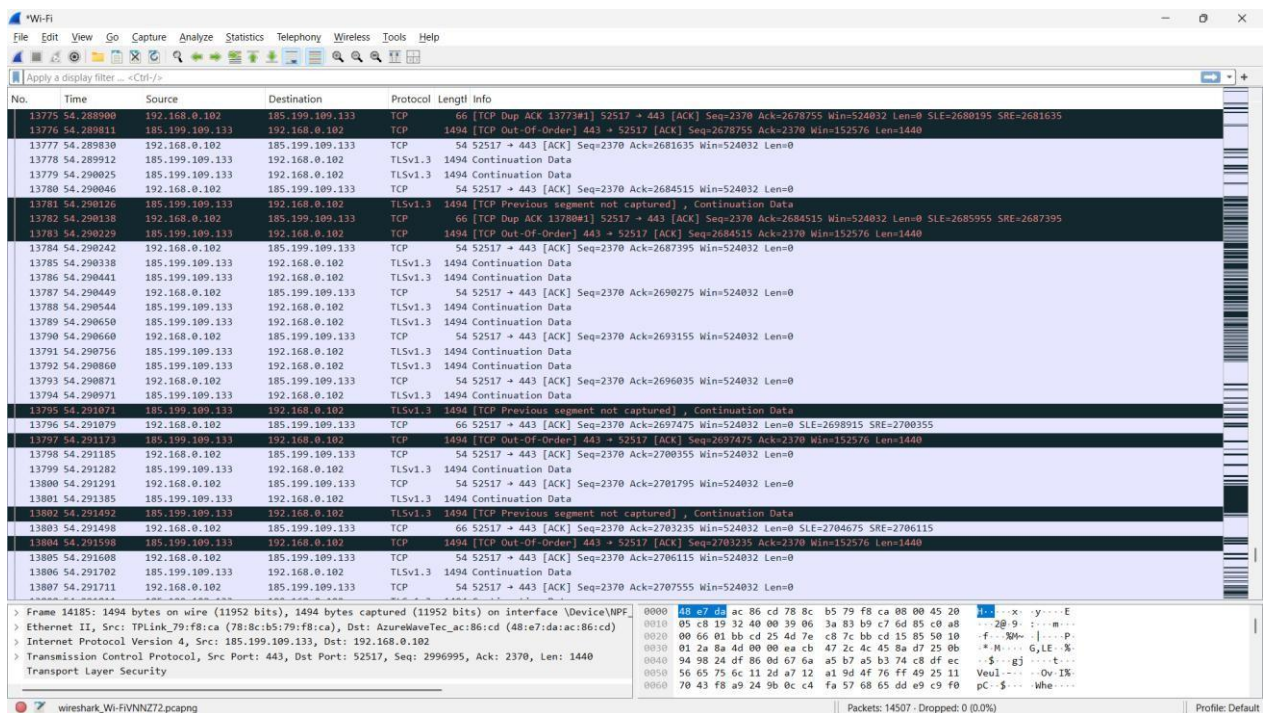
### 2. Start capturing on your active network interface.



## Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark



### 4. Stop capture after a minute.





# Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark

## 5. Filter captured packets by protocol (e.g., HTTP, DNS, TCP).

The image displays two screenshots of the Wireshark network protocol analyzer. The top screenshot shows a list of captured packets filtered by the HTTP protocol. The bottom screenshot shows a list of captured packets filtered by the DNS protocol.

**Top Screenshot: HTTP Filter**

No.	Time	Source	Destination	Protocol	Length	Info
3115	49.095887	192.168.0.102	216.58.203.35	HTTP	256	GET /r/gsr1.cr1 HTTP/1.1
3117	49.112751	216.58.203.35	192.168.0.102	HTTP	277	HTTP/1.1 304 Not Modified
3118	49.118449	192.168.0.102	216.58.203.35	HTTP	254	GET /r/r4.cr1 HTTP/1.1
3119	49.134550	216.58.203.35	192.168.0.102	HTTP	277	HTTP/1.1 304 Not Modified

**Bottom Screenshot: DNS Filter**

No.	Time	Source	Destination	Protocol	Length	Info
2638	19.975856	192.168.0.1	192.168.0.102	DNS	176	Standard query response 0x3bac A survey.webengage.com CNAME survey-alb-new-1082488714.us-east-1.elb.amazonaws.com A 34.198.198...
2936	33.382261	192.168.0.102	192.168.0.1	DNS	87	Standard query 0xae87 A merino.services.mozilla.com
2937	33.399776	192.168.0.1	192.168.0.102	DNS	103	Standard query response 0xae87 A merino.services.mozilla.com A 34.110.138.217
2938	33.400621	192.168.0.102	192.168.0.1	DNS	87	Standard query 0xb997 A merino.services.mozilla.com
2942	33.416825	192.168.0.1	192.168.0.102	DNS	103	Standard query response 0xb997 A merino.services.mozilla.com A 34.110.138.217
2943	33.417061	192.168.0.102	192.168.0.1	DNS	87	Standard query 0xb8d1 AAAA merino.services.mozilla.com
2948	33.433386	192.168.0.1	192.168.0.102	DNS	168	Standard query response 0xb8d1 AAAA merino.services.mozilla.com SOA ns-679.audnsd-20.net
3110	49.051147	192.168.0.102	192.168.0.1	DNS	70	Standard query 0xb18d A c.pki.goog CNAME pki-goog-1.google.com A 216.58.203.35
3111	49.074637	192.168.0.1	192.168.0.102	DNS	121	Standard query response 0xb18d A c.pki.goog CNAME pki-goog-1.google.com A 216.58.203.35
3208	56.392219	192.168.0.102	192.168.0.1	DNS	75	Standard query 0xbd33 A ssl.gstatic.com
3209	56.426300	192.168.0.102	192.168.0.1	DNS	75	Standard query 0xbd33 A ssl.gstatic.com
3210	56.468321	192.168.0.1	192.168.0.102	DNS	91	Standard query response 0xbd33 A ssl.gstatic.com A 142.251.220.3
3211	56.468321	192.168.0.1	192.168.0.102	DNS	91	Standard query response 0xbd33 A ssl.gstatic.com A 142.251.220.3
3257	57.427161	192.168.0.102	192.168.0.1	DNS	87	Standard query 0x54a7 A w3-reporting-nel.reddit.com
3258	57.449560	192.168.0.1	192.168.0.102	DNS	186	Standard query response 0x54a7 A w3-reporting-nel.reddit.com CNAME reddit.map.fastly.net A 151.101.193.140 A 151.101.1.140 A 15...

## Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark

The image displays two screenshots of the Wireshark network traffic analysis tool. The top screenshot shows a capture of TCP traffic on interface \Device\NPF\_{F41EAF25...}. The bottom screenshot shows a capture of ARP traffic on the same interface.

**Top Screenshot: TCP Traffic**

No.	Time	Source	Destination	Protocol	Length	Info
3100	46.956168	192.168.0.102	66.203.125.13	TCP	54	52641 → 443 [ACK] Seq=14269 Ack=11461 Win=255 Len=0
3101	47.006207	51.8.71.184	192.168.0.102	TCP	54	443 → 52679 [ACK] Seq=5868 Ack=5369 Win=64128 Len=0
3102	47.006207	51.8.71.184	192.168.0.102	TLSv1.2	366	Application Data
3103	47.048001	192.168.0.102	51.8.71.184	TCP	54	52679 → 443 [ACK] Seq=5369 Ack=6180 Win=65024 Len=0
3104	47.488195	192.168.0.102	23.193.114.57	TLSv1.2	1162	Application Data
3105	47.503676	23.193.114.57	192.168.0.102	TCP	54	443 → 52633 [ACK] Seq=4179 Ack=21139 Win=604 Len=0
3106	47.704517	23.193.114.57	192.168.0.102	TLSv1.2	225	Application Data
3107	47.704517	23.193.114.57	192.168.0.102	TLSv1.2	128	Application Data
3108	47.704566	192.168.0.102	23.193.114.57	TCP	54	52633 → 443 [ACK] Seq=21139 Ack=4424 Win=254 Len=0
3112	49.077698	192.168.0.102	216.58.203.35	TCP	66	52680 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3113	49.095699	216.58.203.35	192.168.0.102	TCP	66	80 → 52680 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
3114	49.095767	192.168.0.102	216.58.203.35	TCP	54	52680 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
3115	49.095887	192.168.0.102	216.58.203.35	HTTP	256	GET /r/sgsr1.cr1 HTTP/1.1
3116	49.112344	216.58.203.35	192.168.0.102	TCP	54	80 → 52680 [ACK] Seq=1 Ack=203 Win=269568 Len=0
3117	49.112751	216.58.203.35	192.168.0.102	HTTP	277	HTTP/1.1 304 Not Modified
3118	49.118449	192.168.0.102	216.58.203.35	HTTP	254	GET /r/r4.cr1 HTTP/1.1

**Bottom Screenshot: ARP Traffic**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	TPLink_79:f8:ca	AzureWaveTec_ac:86:cd	ARP	42	Who has 192.168.0.102? Tell 192.168.0.1
2	0.000032	AzureWaveTec_ac:86:cd	TPLink_79:f8:ca	ARP	42	192.168.0.102 is at 48:e7:da:ac:86:cd
2858	25.834397	TPLink_79:f8:ca	AzureWaveTec_ac:86:cd	ARP	42	Who has 192.168.0.102? Tell 192.168.0.1
2859	25.834414	AzureWaveTec_ac:86:cd	TPLink_79:f8:ca	ARP	42	192.168.0.102 is at 48:e7:da:ac:86:cd
3109	48.678625	d2:89:80:0e:92:c6	Broadcast	ARP	42	Who has 192.168.0.1? Tell 192.168.0.103
3137	51.852957	TPLink_79:f8:ca	AzureWaveTec_ac:86:cd	ARP	42	Who has 192.168.0.102? Tell 192.168.0.1
3138	51.852974	AzureWaveTec_ac:86:cd	TPLink_79:f8:ca	ARP	42	192.168.0.102 is at 48:e7:da:ac:86:cd

# Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark

## 6. Identify at least 3 different protocols in the capture.

The image displays two screenshots of the Wireshark network traffic analysis tool. Both screenshots show a packet list on the left and a detailed packet pane on the right.

**Top Screenshot:** The packet list shows various network protocols. Key protocols identified include:

- ARP:** Multiple entries for ARP requests and responses between local interfaces.
- DNS:** Numerous standard query requests and responses to various domains like assets.msn.com and sb1.google.com.
- HTTP:** A few entries are visible, including a GET request for a resource from mozilla.com.

**Bottom Screenshot:** This capture shows a wider variety of protocols. Key protocols identified include:

- HTTP:** Multiple GET requests and responses, including one for a resource from mozilla.com.
- MDNS:** Multiple entries for Multicast DNS queries and responses.
- QUIC:** Several entries for QUIC (Quick UDP Internet Connections) traffic, including initial packets and data transfers.

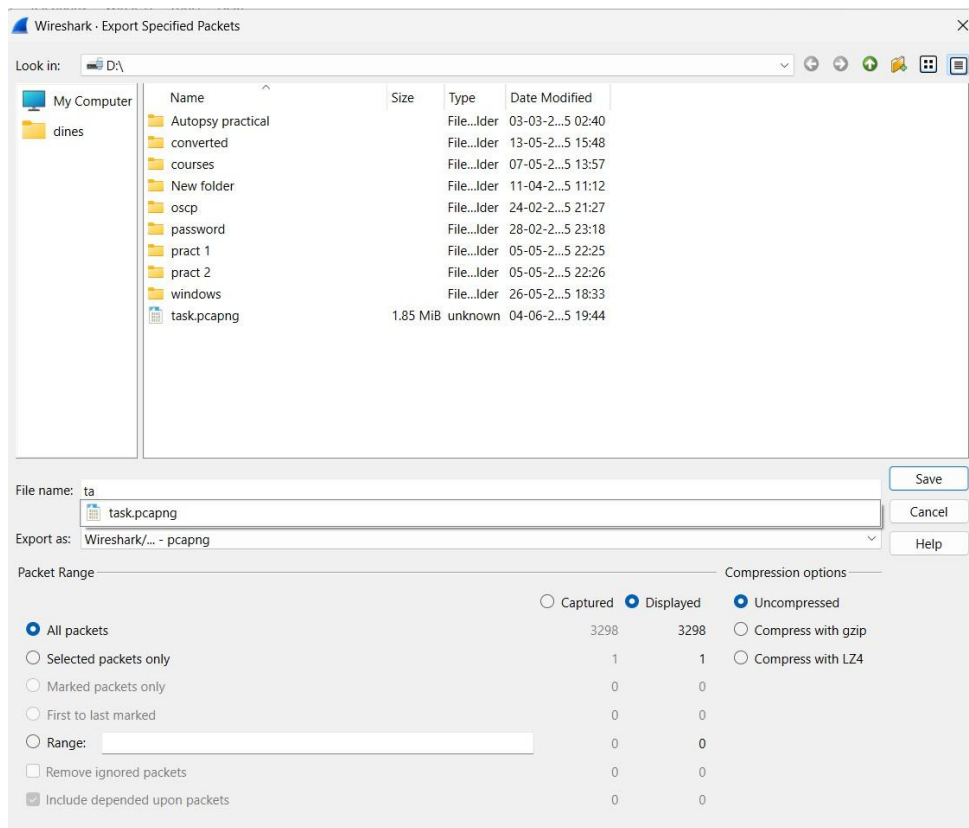
Both screenshots show the packet details pane on the right, providing a hex and ASCII view of the selected packet's raw data.



## Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark

### 7. Export the capture as a .pcap file.

- Go to File > Export Specified Packets or File > Save As.
- Save the file as .pcap  
(e.g.,internship\_capture.pcap).



### 8. Summarize your findings and packet details.

#### Cyber Security Wireshark Capture Summary

**Date/Time of Capture: June 4, 2025**

**Duration: 1 minute**

**Interface: Wi-Fi (wlan0)**

#### Protocols Observed:

1. TCP – Core protocol used for reliable transport between devices.
  - Example: Connection established to 142.250.190.78 (Google IP).



## **Task 5<sup>th</sup>: Capture and Analyze Network Traffic Using Wireshark**

### **2. DNS – Resolved domain names to IP addresses.**

- **Example: `www.google.com` resolved to `142.250.190.78`.**

### **3. HTTP – Unencrypted web traffic.**

- **Example: GET request to `http://example.com/index.html`.**

#### **Interesting Observations:**

- **Several TCP handshakes (SYN, SYN-ACK, ACK) visible.**
- **Multiple DNS queries and responses, showing hostname resolution.**
- **HTTP GET requests and responses with headers and HTML data.**
- **Minimal ICMP traffic from ping tests.**