



TASK 5: CAPTURE AND ANALYZE NETWORK TRAFFIC USING WIRESHARK

Cybersecurity Lab Report

VOONNA VENKATESH
ELEVATE LABS

Task 5:

Capture and Analyze Network Traffic Using Wireshark

Index:

- 1.Install Wireshark.
- 2.Start capturing on your active network interface.
- 3.Browse a website or ping a server to generate traffic.
- 4.Stop capture after a minute.
- 5.Filter captured packets by protocol (e.g., HTTP, DNS, TCP).
- 6.Identify at least 3 different protocols in the capture.
- 7.Export the capture as a .pcap file.
- 8.Summarize your findings and packet details.

Objective

Capture live network packets and identify basic protocols and traffic types

Tools Used

- Wireshark

Steps Performed:

Install Wireshark

Download wireshark and install it.

During the installation accept the installation of Npcap(necessary for packet capturing).

Launch wireshark after completion of installation.

Open Wireshark and select your active network interface from the list(usually **Wi-Fi or Ethernet**).

Click on the interface to begin capturing live packets.

Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.68.20.201	34.237.73.95	TLSv1.2	293	Application Data
2	0.225009	34.237.73.95	10.68.20.201	TLSv1.2	317	Application Data
3	0.270488	10.68.20.201	34.237.73.95	TCP	54	52660 → 443 [ACK] Seq=240 A
4	0.318259	10.68.20.201	104.26.11.240	TCP	55	53480 → 443 [ACK] Seq=1 Ack
5	0.328928	104.26.11.240	10.68.20.201	TCP	66	443 → 53480 [ACK] Seq=1 Ack
6	0.426663	76:72:c3:e3:4c:e9	Broadcast	ARP	56	Who has 10.68.0.1? Tell 10.
7	0.839726	104.22.55.228	10.68.20.201	TLSv1.2	79	Application Data
8	0.840676	10.68.20.201	104.22.55.228	TLSv1.2	83	Application Data
9	0.845919	104.22.55.228	10.68.20.201	TCP	60	443 → 52349 [ACK] Seq=26 Ac
10	0.869011	13.203.1.146	10.68.20.201	TCP	60	443 → 53467 [ACK] Seq=1 Ack
11	0.869063	10.68.20.201	13.203.1.146	TCP	54	[TCP ACKed unseen segment]
12	2.162460	10.68.20.201	3.233.158.25	TCP	55	53535 → 443 [ACK] Seq=1 Ack
13	2.478097	3.233.158.25	10.68.20.201	TCP	66	443 → 53535 [ACK] Seq=1 Ack
14	2.555217	AzureWaveTec_e7:0c:...	Broadcast	ARP	56	Who has 10.68.0.1? Tell 10.

Frame 1: 293 bytes on wire (2344 bits), 293 bytes

Ethernet II, Src: Intel_f8:32:39 (3c:21:9c:f8:32:39), Dst: 08:00:00:00:00:00

Internet Protocol Version 4, Src: 10.68.20.201, Dst: 34.237.73.95

Transmission Control Protocol, Src Port: 52660, Dst Port: 443

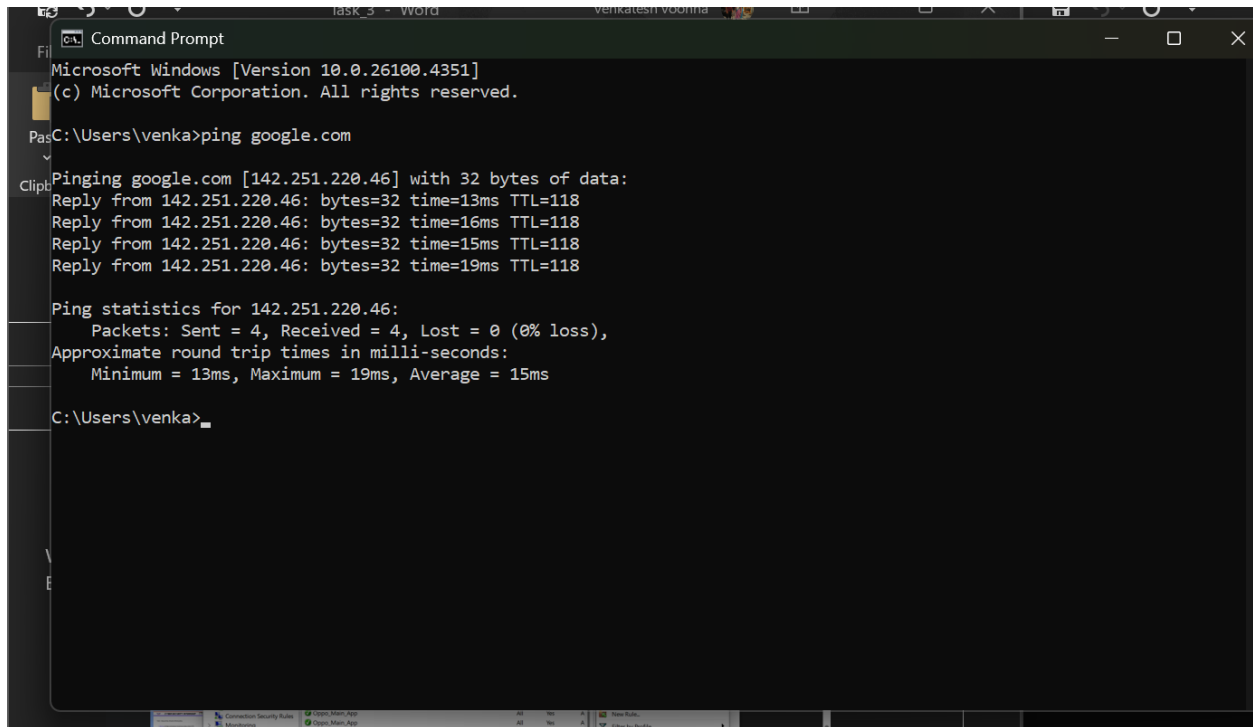
Transport Layer Security

```

0000  b4 0c 25 e2 80 58 3c 21  9c f8 32 39 08 00 45
0010  01 17 2b 89 40 00 80 06  00 00 0a 44 14 c9 22
0020  49 5f cd b4 01 bb 3e 81  57 4d b6 f9 70 2e 50
0030  00 fb 8c 62 00 00 17 03  03 00 ea 10 e8 2d e9
0040  fb d4 90 98 25 14 82 51  be 5b b3 c2 d3 82 e0
0050  a8 d3 70 8f 2d 3d b7 40  18 84 59 1b 9c 5e e3
0060  d7 20 0a 68 d8 b1 89 ed  9e f0 b8 8f fb 73 68
0070  bf ce 87 1d 1e 64 d4 f7  3b 50 40 a1 ac 24 89
0080  d8 aa 26 94 b9 62 cf 93  45 2c 45 b2 db 13 df
0090  47 3b cc 39 4d 97 bf 58  7f a7 1b 10 ec 33 70
00a0  2e 91 37 08 c5 7b cb d8  51 ad 7c 56 53 5e d6
00b0  1a ad b1 8c 93 58 fc 5c  61 48 7e 7d be 30 ef
00c0  ef 7d c5 bc 65 a5 85 6d  9c 86 e8 b4 94 aa 00
00d0  d3 35 3c 1d 39 47 aa 4f  a6 71 76 5a 05 36 7d
00e0  59 39 2f 75 82 ef 5f 42  46 d3 71 d9 a2 fb da
00f0  dd bb 16 c7 69 69 9d e3  46 5d 18 f3 58 d0 3c
0100  09 3d 3e 3a c3 a4 16 93  87 76 1a 64 c3 0a 1a
0110  a3 d6 e2 37 6a 6a 85 03  18 ea 44 a4 90 ae 92
0120  35 fd 23 0c 5f
  
```

While wireshark is running in background open a web browser and visit any website

Ex: <https://example.com>. Open the command prompt and run command: **ping google.com**



```

Microsoft Windows [Version 10.0.26100.4351]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venka>ping google.com

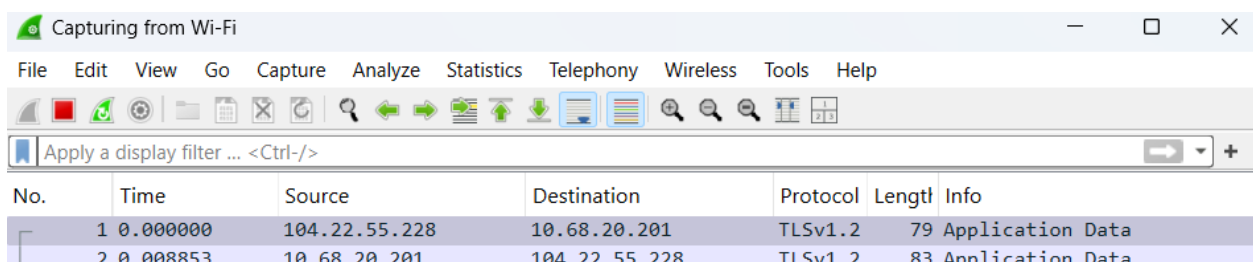
Pinging google.com [142.251.220.46] with 32 bytes of data:
Reply from 142.251.220.46: bytes=32 time=13ms TTL=118
Reply from 142.251.220.46: bytes=32 time=16ms TTL=118
Reply from 142.251.220.46: bytes=32 time=15ms TTL=118
Reply from 142.251.220.46: bytes=32 time=19ms TTL=118

Ping statistics for 142.251.220.46:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 13ms, Maximum = 19ms, Average = 15ms

C:\Users\venka>
  
```

This will create HTTP/HTTPS, DNS, and ICMP traffic.

After a minute, click the red square (stop) button on the left of top toolbar.



Use the **filter bar** to isolate specific protocols:

Protocol	Filter
DNS	dns
TCP	tcp
ICMP	icmp
ARP	arp

DNS(domain name system): name resolution (e.g., A or AAAA record queries).

capture.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

No.	Time	Source	Destination	Protocol	Length	Info
259	33.471493	10.68.20.201	49.45.0.4	DNS	100	Standard query 0x148f A browser-intake-datadoghq.com
261	33.471598	10.68.20.201	49.45.0.4	DNS	100	Standard query 0xf773 HTTPS browser-intake-datadoghq.com
264	33.478162	49.45.0.4	10.68.20.201	DNS	186	Standard query response 0xf773 HTTPS browser-intake-datadoghq.com SOA ns-1907.awsdns-46.co.uk
268	33.480730	49.45.0.4	10.68.20.201	DNS	150	Standard query response 0x148f A browser-intake-datadoghq.com A 3.233.158.26 A 3.233.158.25 A 3.233.158.24
285	33.558720	10.68.20.201	49.45.0.4	DNS	86	Standard query 0x1300 A www.google.com
287	33.558805	10.68.20.201	49.45.0.4	DNS	86	Standard query 0xdb9b HTTPS www.google.com
291	33.564013	49.45.0.4	10.68.20.201	DNS	113	Standard query response 0xdb9b HTTPS www.google.com HTTPS
293	33.564013	49.45.0.4	10.68.20.201	DNS	104	Standard query response 0x1300 A www.google.com A 142.250.77.36
422	33.809150	10.68.20.201	49.45.0.4	DNS	86	Standard query 0xec29 A t0.gstatic.com
426	33.810502	10.68.20.201	49.45.0.4	DNS	86	Standard query 0x19a9 HTTPS t0.gstatic.com
432	33.812694	10.68.20.201	49.45.0.4	DNS	86	Standard query 0x8b36 HTTPS lh3.google.com
434	33.812776	10.68.20.201	49.45.0.4	DNS	86	Standard query 0x0ba4 A lh3.google.com
437	33.813563	49.45.0.4	10.68.20.201	DNS	104	Standard query response 0xec29 A t0.gstatic.com A 142.250.192.36
443	33.817702	49.45.0.4	10.68.20.201	DNS	124	Standard query response 0x0ba4 A lh3.google.com CNAME lh2.l.google.com A 142.251.42.14
451	33.820156	49.45.0.4	10.68.20.201	DNS	158	Standard query response 0x8b36 HTTPS lh3.google.com CNAME lh2.l.google.com SOA ns1.google.com
496	33.864939	10.68.20.201	49.45.0.4	DNS	100	Standard query 0xa815 A ogads-pa.clients6.google.com
501	33.866204	10.68.20.201	49.45.0.4	DNS	100	Standard query 0xbff4 HTTPS ogads-pa.clients6.google.com
509	33.868548	10.68.20.201	49.45.0.4	DNS	87	Standard query 0xe39c A apis.google.com

> Frame 259: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface \Device\NPF_{8A5FDC...} 0000 b4 0c 25 e2 80 58 3c 21 9c f8 32 39 08 00 45 00 ...%..Xc! ..29..E..

> Ethernet II, Src: Intel_f8:32:39 (3c:21:9c:f8:32:39), Dst: PaloAltoNetw_e2:80:58 (b4:0c:25:e2:80:58) 0010 00 56 88 9b 40 00 80 06 00 00 0a 44 14 c9 31 2d ...V..@...D..1-..

> Internet Protocol Version 4, Src: 10.68.20.201, Dst: 49.45.0.4 0020 00 04 cf a5 00 35 38 33 a3 85 cb 5c 40 3c 50 18 ...583 ...@P... ..

> Transmission Control Protocol, Src Port: 53157, Dst Port: 53, Seq: 3, Ack: 1, Len: 46 0030 00 00 18 c3 73 73 73 73 73 73 73 73 73 73 73 73 ...P... ..

TCP(Transfer Control Protocol): used as transport layer for HTTP.

Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
35455	128.093634	10.68.20.201	20.42.65.94	TCP	54	53712 → 443 [ACK] Seq=564
35456	128.093665	10.68.20.201	20.42.65.94	TCP	54	53712 → 443 [ACK] Seq=564
35457	128.093694	10.68.20.201	20.42.65.94	TCP	54	53712 → 443 [ACK] Seq=564
35458	128.093721	10.68.20.201	20.42.65.94	TCP	54	53712 → 443 [ACK] Seq=564
35459	128.096150	10.68.20.201	20.42.65.94	TLSv1.3	134	Change Cipher Spec, Appli
35460	128.096326	10.68.20.201	20.42.65.94	TLSv1.3	1207	Application Data
35461	128.096374	10.68.20.201	20.42.65.94	TLSv1.3	2072	Application Data
35464	128.299657	35.174.127.31	10.68.20.201	TLSv1.2	317	Application Data
35465	128.299657	20.42.65.94	10.68.20.201	TLSv1.3	157	Application Data
35466	128.299657	20.42.65.94	10.68.20.201	TCP	60	443 → 53712 [ACK] Seq=651
35467	128.299657	20.42.65.94	10.68.20.201	TCP	60	443 → 53712 [ACK] Seq=651
35468	128.299657	20.42.65.94	10.68.20.201	TCP	60	443 → 53712 [ACK] Seq=651
35469	128.299657	20.42.65.94	10.68.20.201	TLSv1.3	499	Application Data
35470	128.299935	10.68.20.201	20.42.65.94	TCP	54	53712 → 443 [ACK] Seq=381
35471	128.300030	10.68.20.201	20.42.65.94	TCP	54	53712 → 443 [ACK] Seq=381
35472	128.340448	10.68.20.201	35.174.127.31	TCP	54	52957 → 443 [ACK] Seq=243

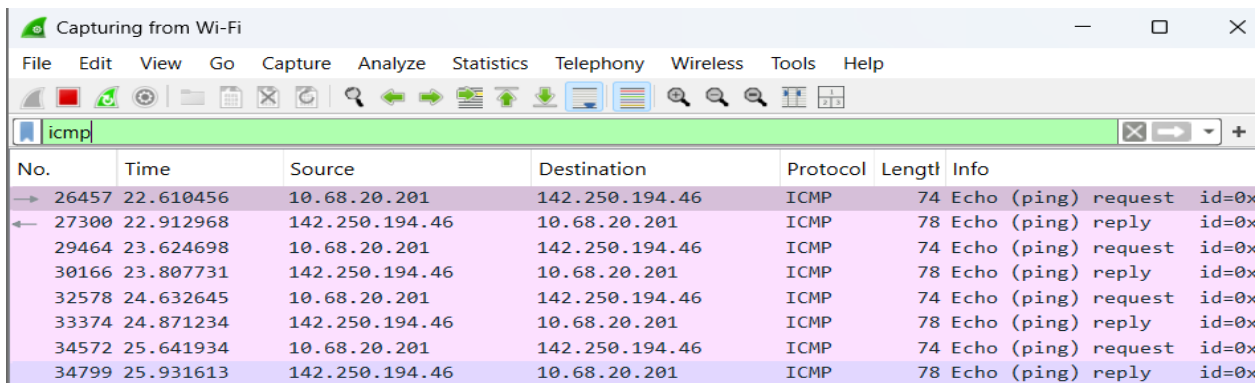
> Frame 26456: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{8A5FDC...} 0000 b4 0c 25 e2 80 58 3c 21 9c f8 32 39 08 00 45 00 ...%..Xc! ..29..E..

> Ethernet II, Src: Intel_f8:32:39 (3c:21:9c:f8:32:39), Dst: PaloAltoNetw_e2:80:58 (b4:0c:25:e2:80:58) 0010 00 28 6f 9d 40 00 80 06 00 00 0a 44 14 c9 0d ...V..@...D..1-..

> Internet Protocol Version 4, Src: 10.68.20.201, Dst: 35.174.127.31 0020 f6 48 d1 b9 01 bb fc 1e ea f9 d1 1b bd c7 50 ...583 ...@P... ..

> Transmission Control Protocol, Src Port: 53689, Dst Port: 443, Seq: 52957, Ack: 443, Len: 54 0030 1f ff 22 db 00 00 ...P... ..

ICMP(Internet Control Message Protocol): ping request and reply.



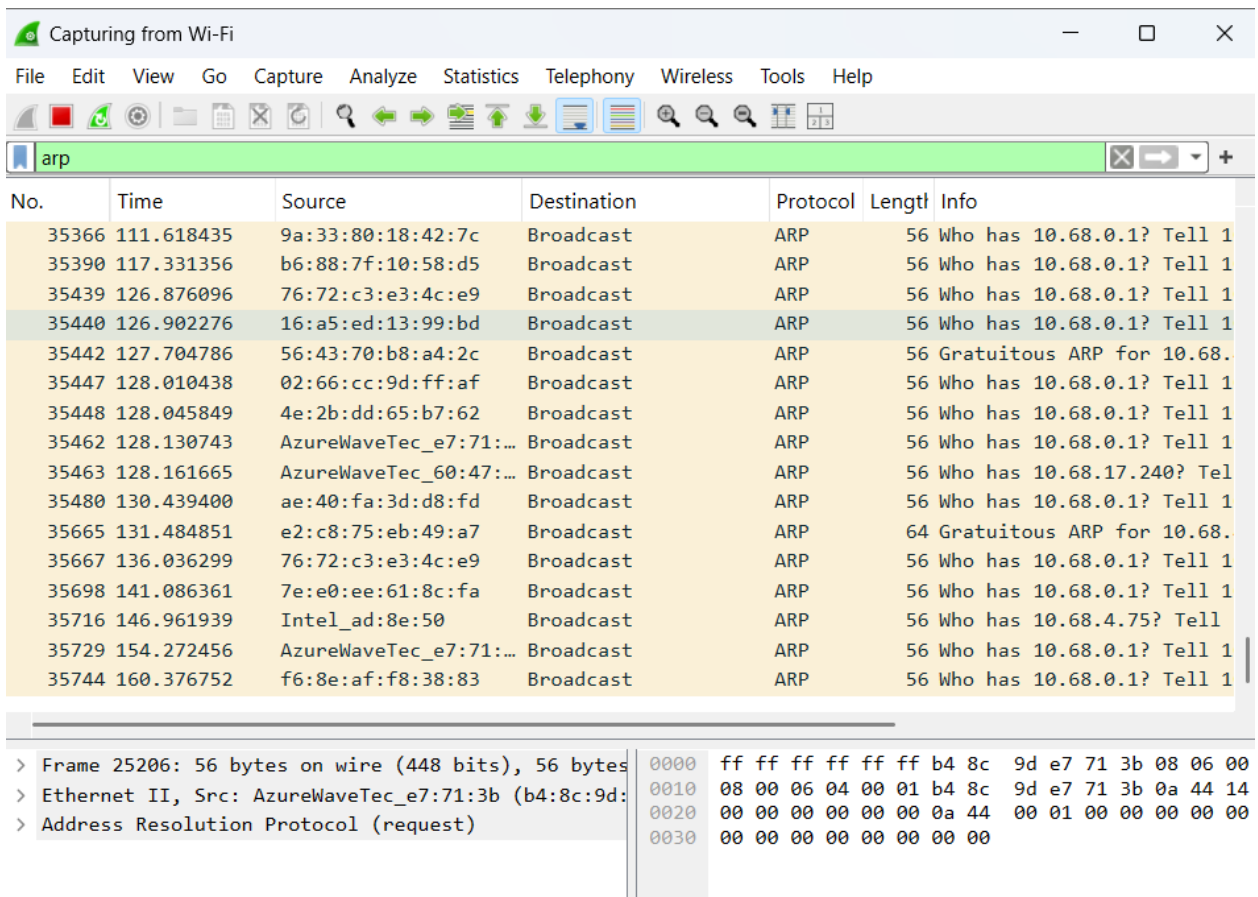
Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
26457	22.610456	10.68.20.201	142.250.194.46	ICMP	74	Echo (ping) request id=0x...
27300	22.912968	142.250.194.46	10.68.20.201	ICMP	78	Echo (ping) reply id=0x...
29464	23.624698	10.68.20.201	142.250.194.46	ICMP	74	Echo (ping) request id=0x...
30166	23.807731	142.250.194.46	10.68.20.201	ICMP	78	Echo (ping) reply id=0x...
32578	24.632645	10.68.20.201	142.250.194.46	ICMP	74	Echo (ping) request id=0x...
33374	24.871234	142.250.194.46	10.68.20.201	ICMP	78	Echo (ping) reply id=0x...
34572	25.641934	10.68.20.201	142.250.194.46	ICMP	74	Echo (ping) request id=0x...
34799	25.931613	142.250.194.46	10.68.20.201	ICMP	78	Echo (ping) reply id=0x...

ARP(Address Resolution Protocol): address resolution between MAC and IP.



Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
35366	111.618435	9a:33:80:18:42:7c	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35390	117.331356	b6:88:7f:10:58:d5	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35439	126.876096	76:72:c3:e3:4c:e9	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35440	126.902276	16:a5:ed:13:99:bd	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35442	127.704786	56:43:70:b8:a4:2c	Broadcast	ARP	56	Gratuitous ARP for 10.68.
35447	128.010438	02:66:cc:9d:ff:af	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35448	128.045849	4e:2b:dd:65:b7:62	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35462	128.130743	AzureWaveTec_e7:71:...	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35463	128.161665	AzureWaveTec_60:47:...	Broadcast	ARP	56	Who has 10.68.17.240? Tel
35480	130.439400	ae:40:fa:3d:d8:fd	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35665	131.484851	e2:c8:75:eb:49:a7	Broadcast	ARP	64	Gratuitous ARP for 10.68.
35667	136.036299	76:72:c3:e3:4c:e9	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35698	141.086361	7e:e0:ee:61:8c:fa	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35716	146.961939	Intel_ad:8e:50	Broadcast	ARP	56	Who has 10.68.4.75? Tell
35729	154.272456	AzureWaveTec_e7:71:...	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1
35744	160.376752	f6:8e:af:f8:38:83	Broadcast	ARP	56	Who has 10.68.0.1? Tell 1

> Frame 25206: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0

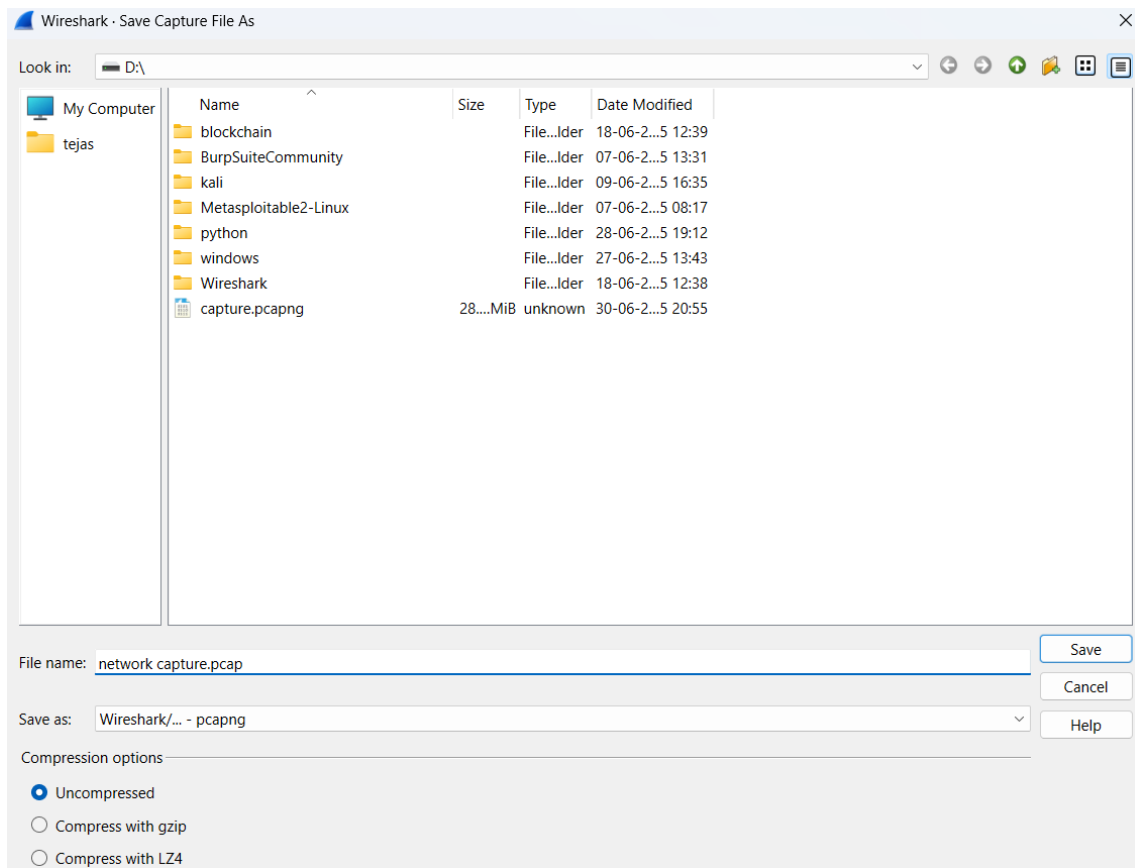
> Ethernet II, Src: AzureWaveTec_e7:71:3b (b4:8c:9d:08:00:06), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Address Resolution Protocol (request)

```

0000  ff ff ff ff ff ff b4 8c 9d e7 71 3b 08 06 00
0010  08 00 06 04 00 01 b4 8c 9d e7 71 3b 0a 44 14
0020  00 00 00 00 00 00 0a 44 00 01 00 00 00 00 00
0030  00 00 00 00 00 00 00 00
  
```

Got to **File>Save as** and Name your file: **network capture.pcap** and click **save**(makesure the file type is **.pcap**).



Conclusion:

This task provide valuable hands-on experience in capturing and analyzing live network traffic using Wireshark. By monitoring real-time packet flow, I was able to identify and filter multiple network protocols such as DNS, TCP, ICMP, and ARP. The exercise enhanced my understanding of how data is transmitted over a network, how different protocols function, and how packet-level analysis can reveal important details about system communication.