

Cybersecurity Lab report – 03

14/11/2024

USN:1BM22IC044

Network Analysis with Wireshark

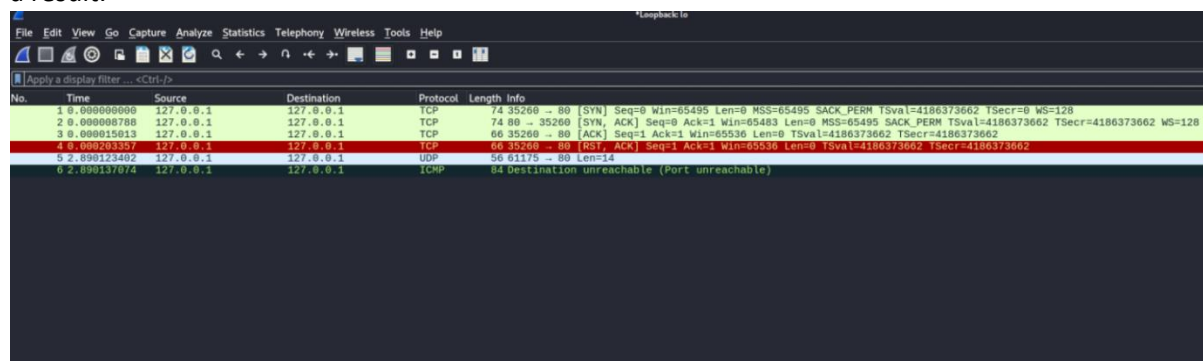
Wireshark is a packet sniffing tool that allows us to capture network traffic, examine the contents of packets, view the protocols in use, and analyze the payloads. It can even reveal sensitive information, such as user credentials, if they are being transmitted. Additionally, Wireshark lets you filter packets based on specific protocols.

Step 1: Open the terminal and run Wireshark on the desired network interface.

wireshark -i lo

```
(kali@kali)~$ wireshark -i lo
** (wireshark:9913) 04:23:22.323669 [WSUtil WARNING] ./wsutil/filter_files.c:242 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 1 doesn't have a quoted filter name.
** (wireshark:9913) 04:23:22.323710 [WSUtil WARNING] ./wsutil/filter_files.c:242 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 2 doesn't have a quoted filter name.
```

Here, I ran the command on the loopback interface to observe the color coding in Wireshark. Then, I executed Nmap commands with options like `-sU` and `-sT` to generate UDP and TCP packets, respectively. Additionally, I sent a ping to create ICMP traffic. The following output was generated as a result.



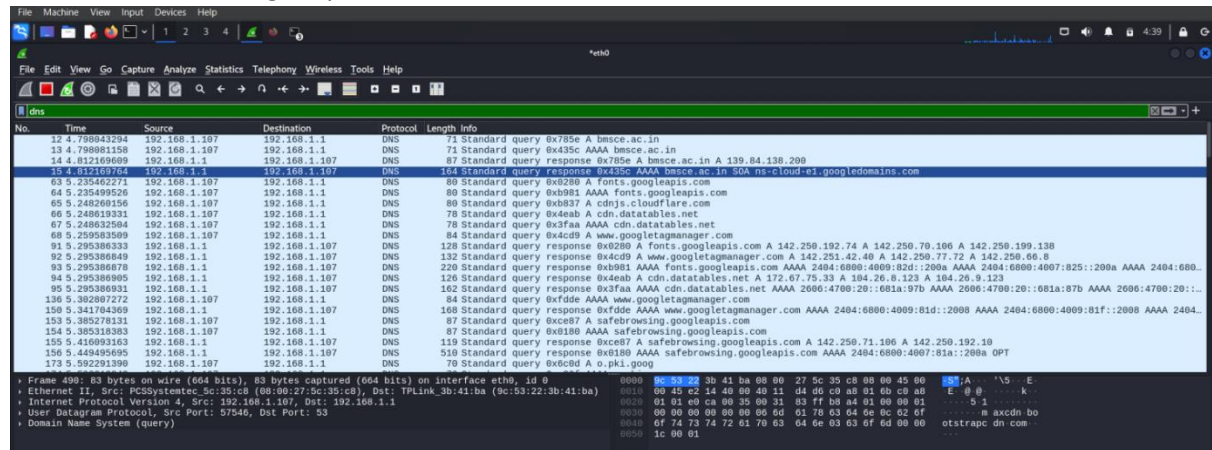
From this, I concluded that UDP packets were displayed in gray, ICMP packets were shown in green, and red packets indicated a reset (RST) or some kind of protocol violation. Additionally, I was able to observe the 3-way handshake that occurred during the communication.

Next, I ran Wireshark on the `eth0` interface using the following command:

wireshark -i eth0

```
(kali@kali)~$ wireshark -i eth0
** (wireshark:3377) 04:10:14.561748 [WSUtil WARNING] ./wsutil/filter_files.c:242 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 1 doesn't have a quoted filter name.
** (wireshark:3377) 04:10:14.561775 [WSUtil WARNING] ./wsutil/filter_files.c:242 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 2 doesn't have a quoted filter name.
```

I opened the browser and searched for the website `**bmsce.ac.in**`. Since we know that DNS queries are sent during this process, I applied a filter in Wireshark to capture DNS traffic. As a result, I obtained the following output.



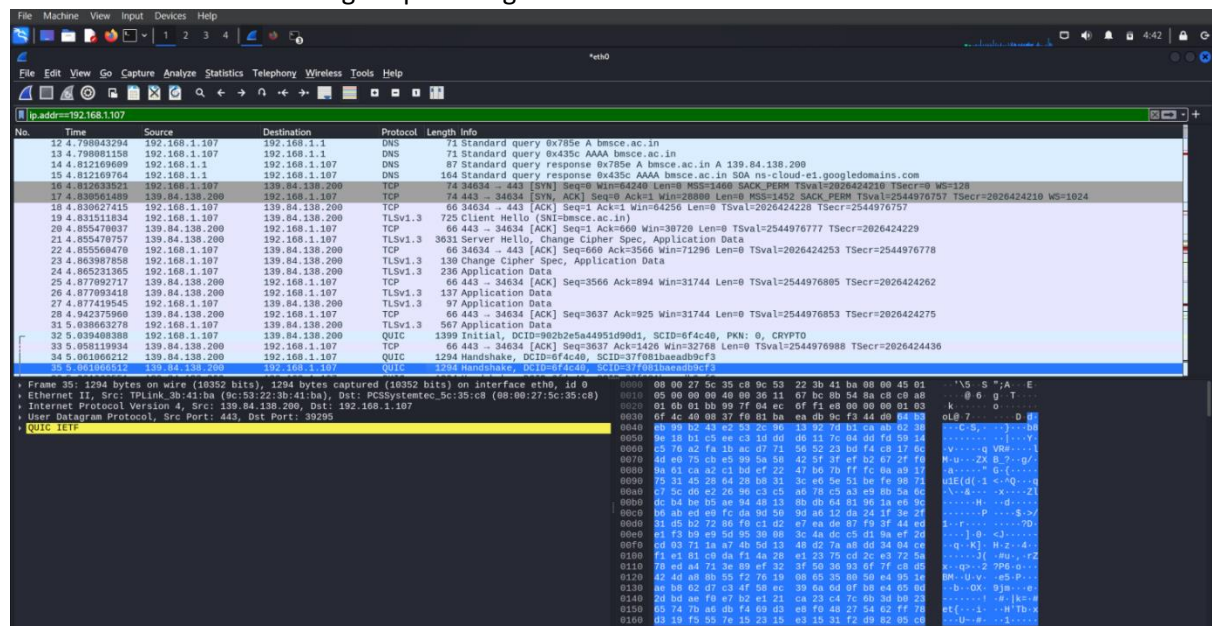
The image shows a Wireshark packet capture of DNS traffic. The packet list on the left shows several DNS packets, with packet 164 highlighted in blue. The packet details pane on the right shows the structure of the DNS query and response. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
12.4	789843294	192.168.1.107	192.168.1.1	DNS	71	Standard query 0x785e A bmsce.ac.in
13.4	789881158	192.168.1.107	192.168.1.1	DNS	71	Standard query 0x435c AAAA bmsce.ac.in
14.4	812169609	192.168.1.1	192.168.1.107	DNS	87	Standard query response 0x785e A bmsce.ac.in A 139.84.138.200
15.4	812169764	192.168.1.1	192.168.1.107	DNS	164	Standard query response 0x435c AAAA bmsce.ac.in SOA ns-cloud-e1.googledomains.com
63.5	235462271	192.168.1.107	192.168.1.1	DNS	80	Standard query 0xb280 A Fonts.googleapis.com
64.5	235489526	192.168.1.107	192.168.1.1	DNS	80	Standard query response 0xb280 A Fonts.googleapis.com
65.5	248269156	192.168.1.107	192.168.1.1	DNS	80	Standard query 0xb837 A cdnjs.cloudflare.com
66.5	248619331	192.168.1.107	192.168.1.1	DNS	78	Standard query 0x4eab A cdn.datatables.net
67.5	248632564	192.168.1.107	192.168.1.1	DNS	78	Standard query response 0x4eab A cdn.datatables.net
68.5	259583509	192.168.1.107	192.168.1.1	DNS	84	Standard query 0xcdc9 A www.googletagmanager.com
91.5	295386333	192.168.1.1	192.168.1.107	DNS	128	Standard query response 0xb280 A Fonts.googleapis.com A 142.250.192.74 A 142.250.70.100 A 142.250.109.138
92.5	295386849	192.168.1.1	192.168.1.107	DNS	132	Standard query response 0xb837 A cdnjs.cloudflare.com A 142.251.42.40 A 142.250.77.72 A 142.250.66.8
93.5	295386678	192.168.1.1	192.168.1.107	DNS	220	Standard query response 0xb981 AAAA Fonts.googleapis.com AAAA 2484:6880:4009:82d::200a AAAA 2484:6880:4007:825::200a AAAA 2484:6880:4009:81f::200a AAAA 2484:6880:4009:81f::200a
94.5	295386905	192.168.1.1	192.168.1.107	DNS	126	Standard query 0x4eab A cdn.datatables.net A 172.67.75.33 A 184.26.8.123 A 184.26.9.123
95.5	295386931	192.168.1.1	192.168.1.107	DNS	162	Standard query response 0x4eab A cdn.datatables.net AAAA 2060:4700:20::681a:97b AAAA 2060:4700:20::681a:97b AAAA 2060:4700:20::681a:97b AAAA 2060:4700:20::681a:97b
136.5	362887272	192.168.1.107	192.168.1.1	DNS	84	Standard query 0xfdde AAAA www.googletagmanager.com
150.5	341784369	192.168.1.1	192.168.1.107	DNS	168	Standard query response 0xfdde AAAA www.googletagmanager.com AAAA 2484:6880:4009:81d::2008 AAAA 2484:6880:4009:81f::2008 AAAA 2484:6880:4009:81f::2008 AAAA 2484:6880:4009:81f::2008
151.5	380278131	192.168.1.107	192.168.1.1	DNS	87	Standard query 0xc8e7 A safefrrowing.googleapis.com
154.5	385318383	192.168.1.107	192.168.1.1	DNS	87	Standard query 0xb180 AAAA safefrrowing.googleapis.com
155.5	416093163	192.168.1.1	192.168.1.107	DNS	119	Standard query response 0xc8e7 A safefrrowing.googleapis.com A 142.250.71.106 A 142.250.192.10
156.5	444849605	192.168.1.1	192.168.1.107	DNS	510	Standard query response 0xb180 AAAA safefrrowing.googleapis.com AAAA 2484:6880:4007:83a:200a OPT
173.5	592291390	192.168.1.107	192.168.1.1	DNS	70	Standard query 0xc8d0 A o.pki.goog

This showed that DNS packets were displayed in blue, which concluded the color coding part.

Packet Filtering:

Next, I applied the filter `ip.addr==192.168.1.107` to capture packets related to the IP address 192.168.1.107. The following output was generated as a result.



The image shows a Wireshark packet capture filtered by the IP address 192.168.1.107. The packet list on the left shows several packets, with packet 35 highlighted in blue. The packet details pane on the right shows the structure of the QUIC packet. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
12.4	789843294	192.168.1.107	192.168.1.1	DNS	71	Standard query 0x785e A bmsce.ac.in
13.4	789881158	192.168.1.107	192.168.1.1	DNS	71	Standard query 0x435c AAAA bmsce.ac.in
14.4	812169609	192.168.1.1	192.168.1.107	DNS	87	Standard query response 0x785e A bmsce.ac.in A 139.84.138.200
15.4	812169764	192.168.1.1	192.168.1.107	DNS	164	Standard query response 0x435c AAAA bmsce.ac.in SOA ns-cloud-e1.googledomains.com
16.4	812693521	192.168.1.107	139.84.138.200	TCP	74	34634 -> 443 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM TSval=2826424228 TSecr=0 WS=128
17.4	839561489	139.84.138.200	192.168.1.107	TCP	74	443 -> 34634 [SYN, ACK] Seq=0 Ack=1 Win=2880 Len=0 MSS=1452 SACK_PERM TSval=2544976757 TSecr=2826424228 WS=1024
18.4	839627415	192.168.1.107	139.84.138.200	TCP	66	34634 -> 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2826424228 TSecr=2544976757
19.4	831511634	192.168.1.107	139.84.138.200	TLVSV1.3	725	Client Hello (SHA1-bmsce.ac.in)
20.4	855479037	139.84.138.200	192.168.1.107	TCP	66	443 -> 34634 [ACK] Seq=1 Ack=600 Win=38720 Len=0 TSval=2544976777 TSecr=2826424229
21.4	855479757	139.84.138.200	192.168.1.107	TLVSV1.3	3631	Server Hello, Change Cipher Spec, Application Data
22.4	855568478	192.168.1.107	139.84.138.200	TCP	66	34634 -> 443 [ACK] Seq=600 Ack=3566 Win=71296 Len=0 TSval=2826424253 TSecr=2544976778
23.4	863897858	192.168.1.107	139.84.138.200	TLVSV1.3	138	Change Cipher Spec, Application Data
24.4	865231365	192.168.1.107	139.84.138.200	TLVSV1.3	236	Application Data
25.4	877892717	139.84.138.200	192.168.1.107	TCP	66	443 -> 34634 [ACK] Seq=3566 Ack=894 Win=31744 Len=0 TSval=2544976853 TSecr=2826424275
26.4	877893418	139.84.138.200	192.168.1.107	TLVSV1.3	137	Application Data
27.4	877419545	192.168.1.107	139.84.138.200	TLVSV1.3	97	Application Data
28.4	942375960	139.84.138.200	192.168.1.107	TCP	66	443 -> 34634 [ACK] Seq=3637 Ack=925 Win=31744 Len=0 TSval=2544976988 TSecr=2826424436
31.5	938663278	192.168.1.107	139.84.138.200	TLVSV1.3	567	Application Data
32.5	939488388	192.168.1.107	139.84.138.200	QUIC	1399	Initial, DCID=962b2e5a44951d90d1, SCID=0f4c4b, PKN: 0, CRYPTO
33.5	958119334	139.84.138.200	192.168.1.107	TCP	66	443 -> 34634 [ACK] Seq=3637 Ack=1326 Win=32768 Len=0 TSval=2544976988 TSecr=2826424436
34.5	961066212	139.84.138.200	192.168.1.107	QUIC	1294	Handshake, DCID=0f4c4b, SCID=37f681bae4dbcf3
35.5	961066512	139.84.138.200	192.168.1.107	QUIC	1294	Handshake, DCID=0f4c4b, SCID=37f681bae4dbcf3

I entered the filter **tcp.port==80** to get the tcp packets consists of port 80 either in source or destination

No.	Time	Source	Destination	Protocol	Length	Info
187	0.0174439	192.168.1.107	142.250.183.131	TCP	74	51178 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM TSval=1460989826 TSecr=0 WS=128
262	5.636174346	142.250.183.131	192.168.1.107	TCP	74	80 → 51178 [ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM TSval=1460989806 TSecr=256
263	6.036215570	192.168.1.107	142.250.183.131	TCP	60	51178 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 MSS=1460 SACK_PERM TSval=1460989826 TSecr=2428778019
264	6.039134728	192.168.1.107	142.250.183.131	OCSP	494	Request
265	6.049412799	142.250.183.131	192.168.1.107	TCP	60	80 → 51178 [ACK] Seq=1 Ack=429 Win=132096 Len=0 TSval=2420778019 TSecr=1460989826
230	7.22867899	192.168.1.107	172.64.149.23	TCP	74	39216 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM TSval=2799152273 TSecr=0 WS=128
231	7.27679250	192.168.1.107	173.223.235.16	TCP	66	42084 → 80 [ACK] Seq=1 Ack=1 Win=496 Len=0 TSval=251830611 TSecr=3466657908
232	7.33123262	142.250.183.131	192.168.1.107	OCSP	768	Response
233	7.42117240	142.250.183.131	192.168.1.107	TCP	66	41878 → 443 [ACK] Seq=1397 Ack=5110 Win=75904 Len=0 TSval=3439781280 TSecr=3917226009
234	7.53170646	192.168.1.107	142.250.183.131	TCP	66	51178 → 80 [ACK] Seq=429 Ack=793 Win=63360 Len=0 TSval=1460989723 TSecr=2420778019
238	7.743918990	172.64.149.23	192.168.1.107	TCP	74	80 → 39216 [SYN] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=3726732389 TSecr=2799152273 WS=8192
239	7.743959883	192.168.1.107	172.64.149.23	TCP	66	39216 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2799152293 TSecr=3726732389
240	7.745483599	192.168.1.107	172.64.149.23	OCSP	497	Request
247	7.754287589	172.64.149.23	192.168.1.107	TCP	66	80 → 39216 [ACK] Seq=1 Ack=432 Win=131072 Len=0 TSval=3726732399 TSecr=2799152294
252	7.799421352	192.168.1.107	172.64.149.23	TCP	66	39216 → 80 [ACK] Seq=432 Ack=1018 Win=63360 Len=0 TSval=2799152348 TSecr=3726732439
278	8.89129553	192.168.1.107	142.250.183.131	TCP	74	51182 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM TSval=1460989879 TSecr=0 WS=128
280	9.908392926	142.250.183.131	192.168.1.107	TCP	74	80 → 51182 [SYN] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM TSval=2077458756 TSecr=1460989879 WS=256
290	9.908392463	192.168.1.107	142.250.183.131	TCP	66	51182 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1460989898 TSecr=2077458756
292	9.908979872	192.168.1.107	142.250.183.131	OCSP	494	Request
293	9.915396302	142.250.183.131	192.168.1.107	TCP	66	80 → 51182 [ACK] Seq=1 Ack=429 Win=132096 Len=0 TSval=2077458756 TSecr=1460989898
307	9.957859508	192.168.1.107	173.223.235.9	TCP	66	52356 → 80 [ACK] Seq=1 Ack=1 Win=496 Len=0 TSval=1288070646 TSecr=1027718134

Frame 187: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec, Sc:35:c8 (08:00:27:5c:35:c8), Dst: TPLink_3b:41:ba (9c:53:22:3b:41:ba)
Internet Protocol Version 4, Src: 192.168.1.107, Dst: 142.250.183.131
Transmission Control Protocol, Src Port: 51178, Dst Port: 80, Seq: 0, Len: 0

to filter the packets which has 192.168.1.107 has source I applied the filter **ip.src==192.168.1.107**

No.	Time	Source	Destination	Protocol	Length	Info
226	5.715964541	192.168.1.107	192.168.1.1	DNS	76	Standard query 0x18d0 AAAA ocp.sectigo.com
227	5.716018643	192.168.1.107	192.168.1.1	DNS	76	Standard query response 0x18d0 AAAA ocp.sectigo.com
230	5.723867899	192.168.1.107	172.64.149.23	TCP	74	39216 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM TSval=2799152273 TSecr=0 WS=128
231	5.727679250	192.168.1.107	173.223.235.16	TCP	66	42084 → 80 [ACK] Seq=1 Ack=1 Win=496 Len=0 TSval=251830611 TSecr=3466657908
232	5.73123262	192.168.1.107	142.250.183.131	TCP	66	41878 → 443 [ACK] Seq=1397 Ack=5110 Win=75904 Len=0 TSval=3439781280 TSecr=3917226009
233	5.73454747	192.168.1.107	142.250.171.106	TLSv1.3	130	Change Cipher Spec, Application Data
236	5.735249435	192.168.1.107	142.250.171.106	TLSv1.3	236	Application Data
237	5.735533966	192.168.1.107	142.250.171.106	TLSv1.3	553	Application Data
239	5.743959883	192.168.1.107	172.64.149.23	TCP	66	39216 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2799152293 TSecr=3726732389
240	5.745483599	192.168.1.107	172.64.149.23	OCSP	497	Request
242	5.753605517	192.168.1.107	142.250.171.106	TCP	66	41878 → 443 [ACK] Seq=1397 Ack=5110 Win=75904 Len=0 TSval=3439781280 TSecr=3917226009
246	5.754939227	192.168.1.107	142.250.171.106	TLSv1.3	97	Application Data
251	5.760834762	192.168.1.107	142.251.42.40	QUIC	1399	Initial, DCID=f8327f, SCID=eaab5386dc09e7ce, PKN: 0, CRYPTO
253	5.769446825	192.168.1.107	172.64.149.23	TCP	66	39216 → 80 [ACK] Seq=432 Ack=1018 Win=63360 Len=0 TSval=2799152348 TSecr=3726732439
255	5.801976990	192.168.1.107	142.250.171.106	TCP	66	41878 → 443 [ACK] Seq=1428 Ack=5141 Win=75904 Len=0 TSval=3439781329 TSecr=3917226011
256	5.802025636	192.168.1.107	13.234.159.72	TCP	66	69998 → 443 [ACK] Seq=792 Ack=6305 Win=79322 Len=0 TSval=3747154124 TSecr=2784199031
257	5.806184508	192.168.1.107	142.251.42.40	QUIC	82	Handshake, DCID=eaab5386dc09e7ce, SCID=f8327f
260	5.827666306	192.168.1.107	142.250.171.106	TCP	66	41878 → 443 [ACK] Seq=1428 Ack=5652 Win=78720 Len=0 TSval=3439781354 TSecr=3917226079
261	5.827679637	192.168.1.107	142.250.171.106	TCP	66	41878 → 443 [ACK] Seq=1428 Ack=5722 Win=78720 Len=0 TSval=3439781354 TSecr=3917226082
262	5.827835443	192.168.1.107	142.250.171.106	TLSv1.3	136	Application Data
263	5.828087295	192.168.1.107	142.250.171.106	TCP	66	41878 → 443 [ACK] Seq=1492 Ack=5723 Win=78720 Len=0 TSval=3439781374 TSecr=3917226101
266	5.847604242	192.168.1.107	142.250.171.106	TCP	66	41878 → 443 [ACK] Seq=5722 Ack=1467 Win=268032 Len=0 TSval=3917226101 TSecr=3439781355

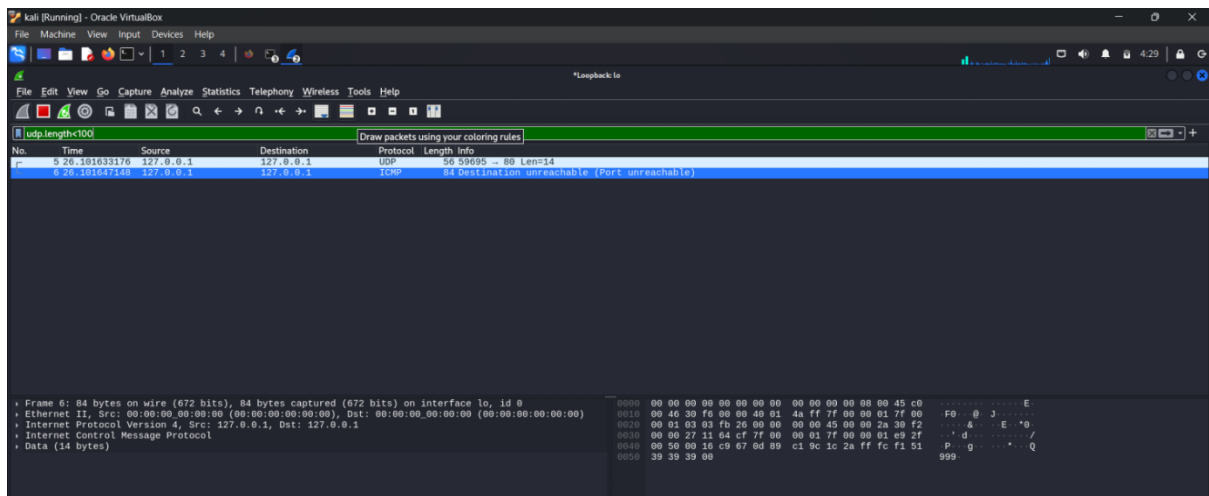
Frame 246: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec, Sc:35:c8 (08:00:27:5c:35:c8), Dst: TPLink_3b:41:ba (9c:53:22:3b:41:ba)
Internet Protocol Version 4, Src: 192.168.1.107, Dst: 142.250.171.106
Transmission Control Protocol, Src Port: 41878, Dst Port: 443, Seq: 1397, Ack: 5110, Len: 31
Transport Layer Security

To identify the packets which has 192.168.1.107 has destination I applied the filter **ip.dst==192.168.1.107**

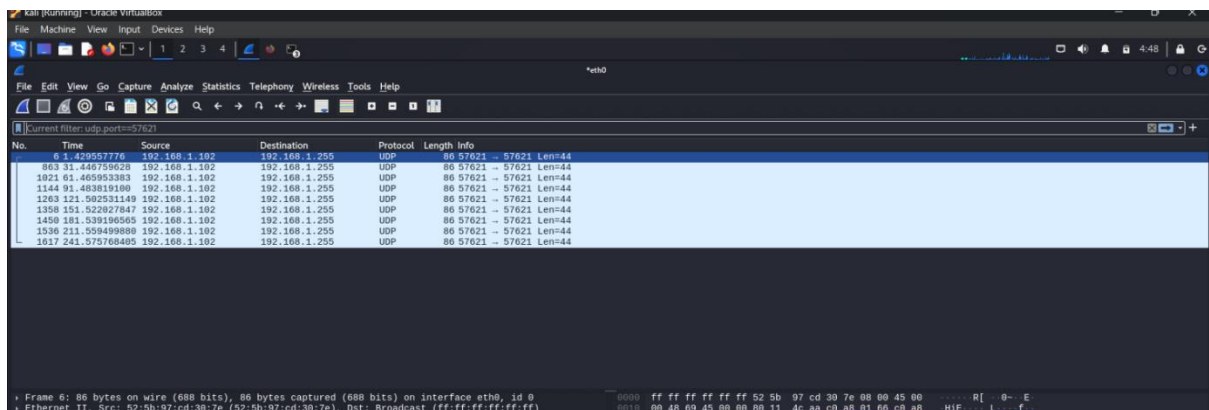
No.	Time	Source	Destination	Protocol	Length	Info
226	5.715964541	142.234.159.72	192.168.1.107	TCP	66	443 → 69998 [ACK] Seq=1 Ack=666 Win=28288 Len=0 TSval=1270419919 TSecr=3747153986
221	5.769398632	13.234.159.72	192.168.1.107	TLSv1.2	4162	Server Hello
223	5.789789742	13.234.159.72	192.168.1.107	TLSv1.2	2060	Certificate, Server Key Exchange, Server Hello Done
228	5.722825031	192.168.1.1	192.168.1.107	DNS	158	Standard query response 0x18d0 A ocp.sectigo.com CNAME ocp.cloudflare.com cdn.cloudflare.net A 172.64.149.23 A 194.18.38.233
229	5.722825399	192.168.1.1	192.168.1.107	DNS	182	Standard query response 0x18d0 AAAA ocp.sectigo.com CNAME ocp.cloudflare.com cdn.cloudflare.net AAAA 2606:4700:4a00:ac49:9517::
232	5.733123262	142.250.183.131	192.168.1.107	OCSP	768	Response
233	5.73454747	192.168.1.107	142.250.171.106	TCP	66	41878 → 443 [ACK] Seq=1397 Ack=5110 Win=75904 Len=0 TSval=3439781280 TSecr=3917226009
236	5.735249435	192.168.1.107	142.250.171.106	TLSv1.3	130	Change Cipher Spec, Application Data
237	5.735533966	192.168.1.107	142.250.171.106	TLSv1.3	236	Application Data
239	5.743959883	192.168.1.107	172.64.149.23	TCP	74	80 → 39216 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM TSval=2799152273 TSecr=0 WS=128
241	5.751885953	142.250.171.106	192.168.1.107	TCP	66	443 → 41878 [ACK] Seq=4490 Ack=740 Win=268288 Len=0 TSval=3917226080 TSecr=3439781262
242	5.751885439	142.250.171.106	192.168.1.107	TCP	66	443 → 41878 [ACK] Seq=4490 Ack=910 Win=268288 Len=0 TSval=3917226080 TSecr=3439781262
243	5.752679324	142.250.171.106	192.168.1.107	TCP	66	443 → 41878 [ACK] Seq=4490 Ack=1397 Win=268032 Len=0 TSval=3917226089 TSecr=3439781262
244	5.753531135	142.250.171.106	192.168.1.107	TLSv1.3	586	Application Data, Application Data
247	5.754287589	172.64.149.23	192.168.1.107	TCP	74	80 → 39216 [ACK] Seq=1 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=3726732389 TSecr=2799152273 WS=8192
248	5.755051299	13.234.159.72	192.168.1.107	TLSv1.2	340	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
249	5.75566424	142.250.171.106	192.168.1.107	TLSv1.3	97	Application Data
250	5.770529231	142.250.171.106	192.168.1.107	TCP	66	443 → 41878 [ACK] Seq=5141 Ack=1428 Win=268032 Len=0 TSval=3917226092 TSecr=3439781281
252	5.799421352	172.64.149.23	192.168.1.107	OCSP	1083	Response
254	5.80215534	142.251.42.40	192.168.1.107	QUIC	1399	Initial, DCID=f8327f, SCID=eaab5386dc09e7ce, PKN: 1, ACK, CRYPTO, PADDING
259	5.827634474	142.250.171.106	192.168.1.107	TLSv1.3	577	Application Data, Application Data
259	5.827634474	142.250.171.106	192.168.1.107	TCP	66	443 → 41878 [ACK] Seq=5722 Ack=1467 Win=268032 Len=0 TSval=3917226101 TSecr=3439781355
264	5.846929614	142.250.171.106	192.168.1.107	TCP	66	443 → 41878 [ACK] Seq=5722 Ack=1467 Win=268032 Len=0 TSval=3917226101 TSecr=3439781355
265	5.847604242	142.250.171.106	192.168.1.107	TCP	66	443 → 41878 [ACK] Seq=5722 Ack=1467 Win=268032 Len=0 TSval=3917226101 TSecr=3439781355

Frame 244: 688 bytes on wire (5504 bits), 688 bytes captured (5504 bits) on interface eth0, id 0
Ethernet II, Src: TPLink_3b:41:ba (9c:53:22:3b:41:ba), Dst: PCSSystemtec, Sc:35:c8 (08:00:27:5c:35:c8)
Internet Protocol Version 4, Src: 142.250.171.106, Dst: 192.168.1.107
Transmission Control Protocol, Src Port: 443, Dst Port: 41878, Seq: 4490, Ack: 1397, Len: 614
Transport Layer Security

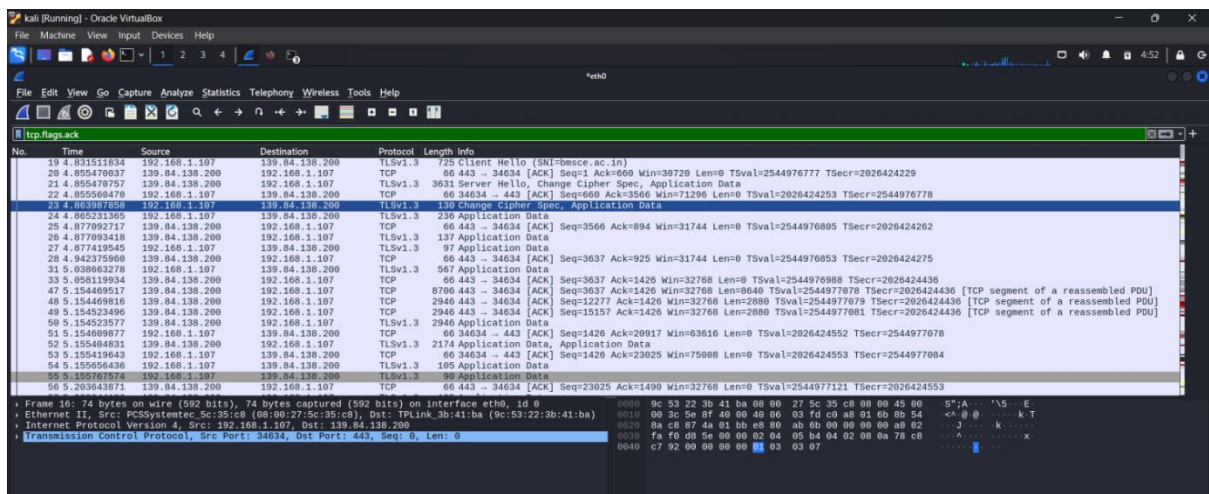
To filter the packets which has udp length less than particular value say 100 i ran the following filter **udp.length<100**



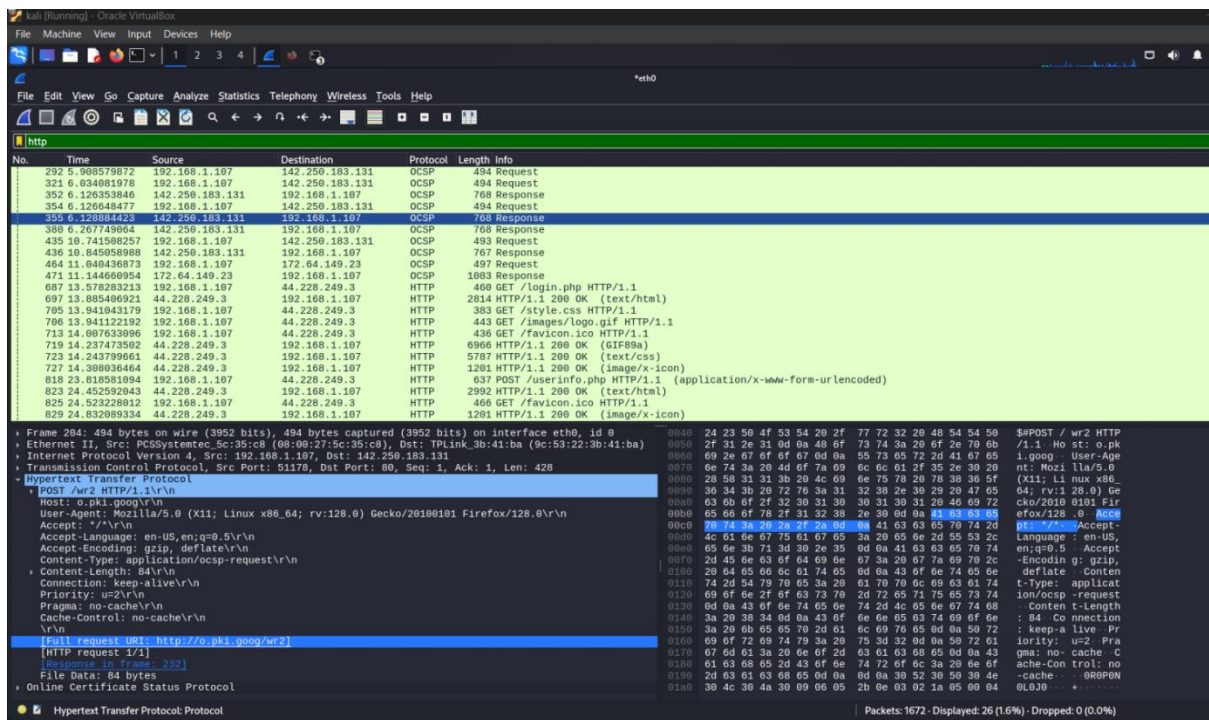
To get udp packet associated with a particular port i ran the following filter **udp.port==<port>**



To find packets with the ack flag set i ran the following filter **tcp.flags.ack**



To filter http packets i ran the following filter **http**

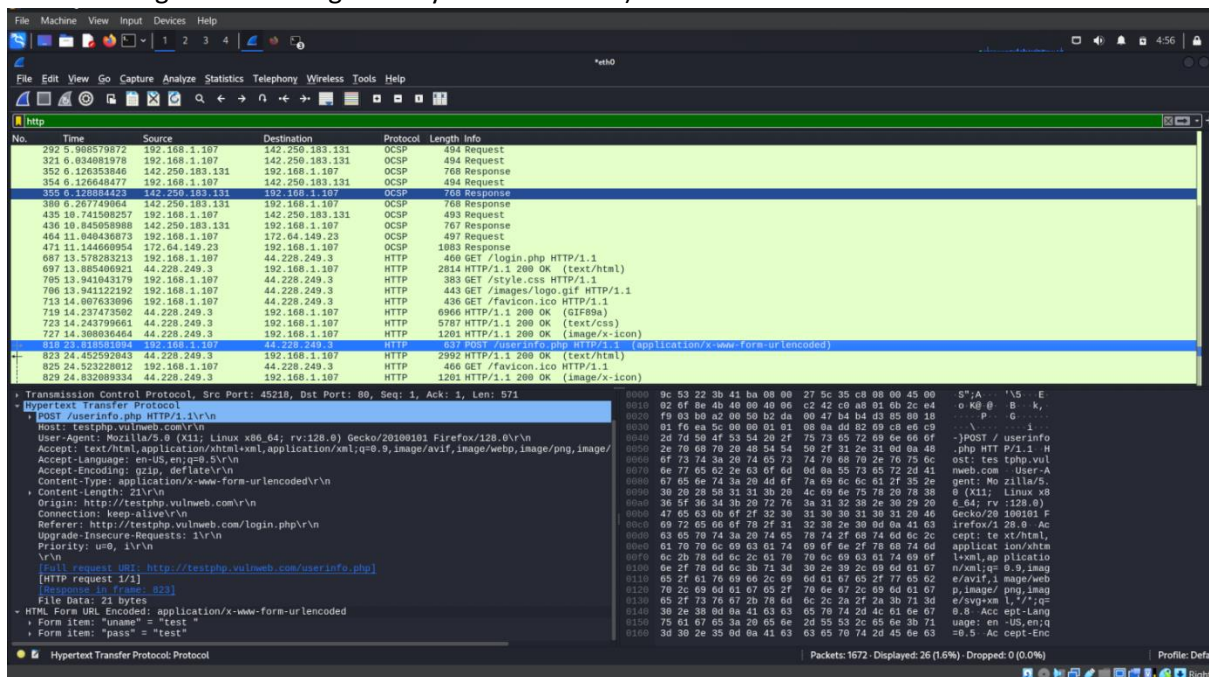


This concludes the filtering section

Packet analysis

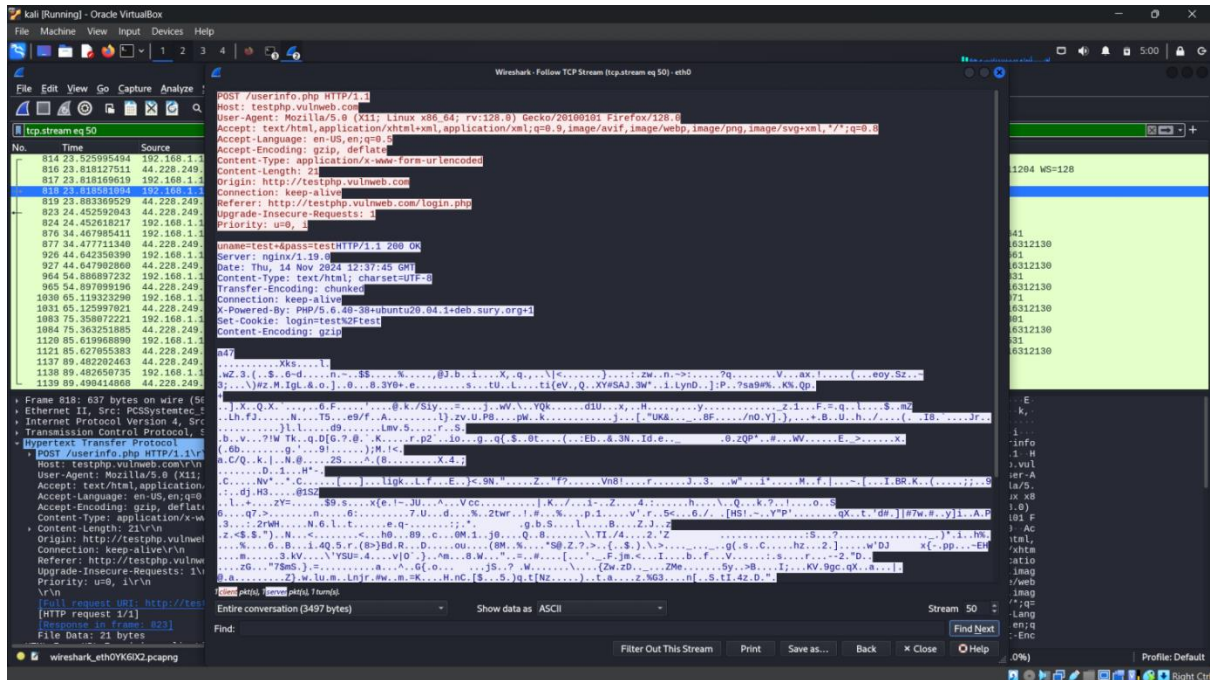
In the final image, I noticed a packet containing a POST request (the 4th one from the end), so I decided to investigate it further. I examined the application layer of the protocol and found that the user was attempting to submit data to the /userinfo.php page. Upon inspecting the details, I discovered that the login parameters were listed as `uname="test"` and `pass="test"`.

(Note: This test was performed on <http://testphp.vulnweb.com/login.php>, a purposely vulnerable website designed for testing security vulnerabilities.)



I right-clicked on that specific packet and selected the “Follow” option. This revealed two additional choices: Follow TCP Stream and Follow HTTP Stream. I analyzed both of these streams, and here are the outputs I obtained from each.

TCPSTREAM



HTTPSTREAM

