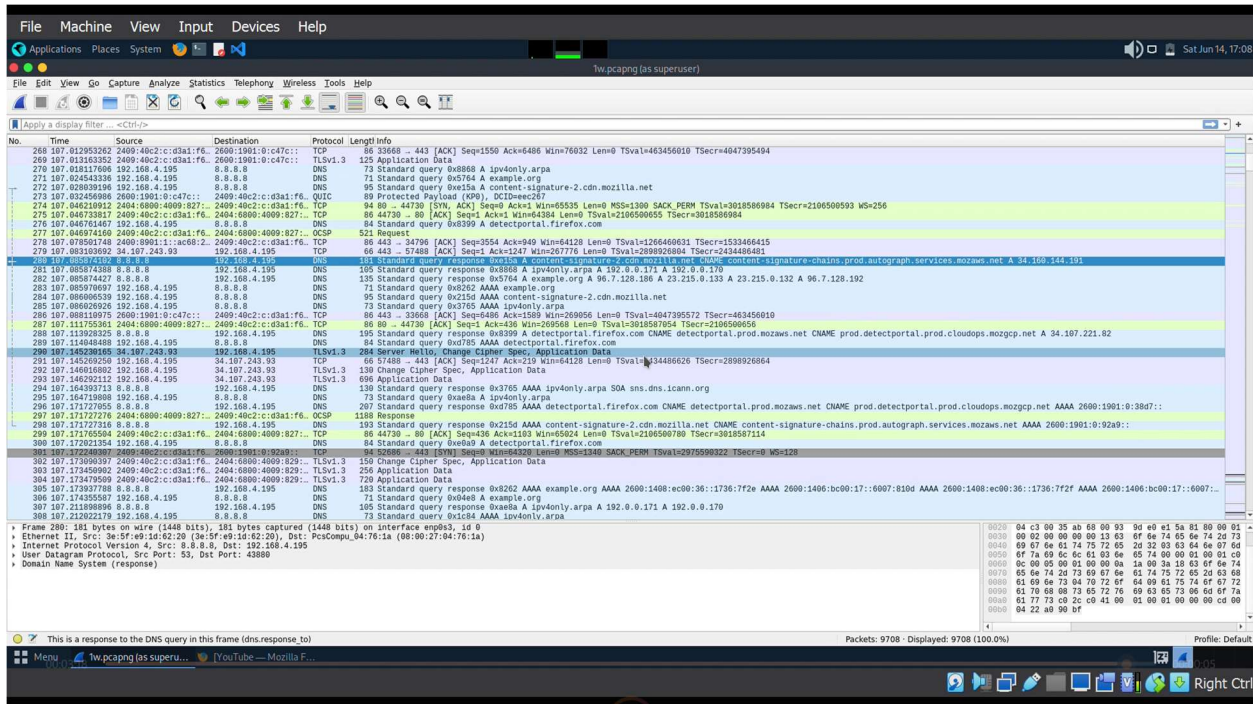


WIRESHARK REPORT



No.	Time	Source	Destination	Protocol	Length	Info
268	107.012953262	2409:40c2:c:d3a1:f6...	2600:1901:0:c47c::...	TCP	86	33668 → 443 [ACK] Seq=1550 Ack=486 Win=76032 Len=0 TSval=463456010 TSecr=4047395494
269	107.013033352	2409:40c2:c:d3a1:f6...	2600:1901:0:c47c::...	TLSv1.3	125	Application Data
270	107.013117696	192.168.4.195	8.8.8.8	DNS	73	Standard query 0x8868 A ipv4only.arpa
271	107.024433336	192.168.4.195	8.8.8.8	DNS	71	Standard query 0x5764 A example.org
272	107.028939196	192.168.4.195	8.8.8.8	DNS	95	Standard query 0xe35a A content-signature-2.cdn.mozilla.net
273	107.032456986	2600:1901:0:c47c::...	2409:40c2:c:d3a1:f6...	QUIC	89	Protected Payload (KPG) DCID=ec267
274	107.040218912	2404:6800:4009:827::...	2409:40c2:c:d3a1:f6...	TCP	84	80 → 44730 [SYN, ACK] Seq=1 Ack=1 Win=5535 Len=0 MSS=1300 SACK_PERM TSval=3018586984 TSecr=2106506055 MS=256
275	107.040733817	2409:40c2:c:d3a1:f6...	2404:6800:4009:827::...	TCP	86	44730 → 80 [ACK] Seq=1 Ack=1 Win=64384 Len=0 TSval=2106506055 TSecr=3018586984
276	107.046701467	192.168.4.195	8.8.8.8	DNS	84	Standard query 0x6399 A detectportal.firefox.com
277	107.046974169	2409:40c2:c:d3a1:f6...	2404:6800:4009:827::...	OCSP	521	Request
278	107.078901748	2409:40c2:c:d3a1:f6...	2409:40c2:c:d3a1:f6...	TCP	86	443 → 34786 [ACK] Seq=3554 Ack=90 Win=64128 Len=0 TSval=1265460801 TSecr=1533466415
279	107.083103092	34.107.243.93	192.168.4.195	TCP	66	443 → 57488 [ACK] Seq=1 Ack=1247 Win=267776 Len=0 TSval=288926804 TSecr=2434486481
280	107.085074128	8.8.8.8	192.168.4.195	DNS	111	Standard query response 0x15c4 A content-signature-2.cdn.mozilla.net CNAME content-signature-chains.prod.autograph.services.mozilla.net A 34.160.144.191
281	107.085074128	8.8.8.8	192.168.4.195	DNS	109	Standard query response 0x5768 A ipv4only.arpa A 192.0.0.171 A 192.0.0.170
282	107.085074427	8.8.8.8	192.168.4.195	DNS	135	Standard query response 0x5764 A example.org A 96.7.128.106 A 23.215.0.133 A 23.215.0.132 A 96.7.128.192
283	107.085076057	192.168.4.195	8.8.8.8	DNS	71	Standard query 0x5262 AAAA example.org
284	107.086080539	192.168.4.195	8.8.8.8	DNS	95	Standard query 0x215d AAAA content-signature-2.cdn.mozilla.net
285	107.086080539	192.168.4.195	8.8.8.8	DNS	73	Standard query 0x3765 AAAA ipv4only.arpa
286	107.088110975	2600:1901:0:c47c::...	2409:40c2:c:d3a1:f6...	TCP	86	443 → 33668 [ACK] Seq=4886 Ack=1589 Win=269056 Len=0 TSval=4047395572 TSecr=463456010
287	107.111753361	2404:6800:4009:827::...	2409:40c2:c:d3a1:f6...	TCP	86	80 → 44730 [ACK] Seq=1 Ack=436 Win=269568 Len=0 TSval=3018587054 TSecr=2106506055
288	107.113023125	8.8.8.8	192.168.4.195	DNS	195	Standard query response 0x6399 A detectportal.firefox.com CNAME detectportal.prod.mozilla.net CNAME prod.detectportal.prod.cloudops.mozgcp.net A 34.107.221.82
289	107.114048489	192.168.4.195	8.8.8.8	DNS	84	Standard query 0xd785 AAAA detectportal.firefox.com
290	107.140209065	34.107.243.93	192.168.4.195	TLSv1.3	284	Server Hello: Change Cipher Spec, Application Data
291	107.145309250	192.168.4.195	34.107.243.93	TCP	66	57488 → 443 [ACK] Seq=1247 Ack=219 Win=64128 Len=0 TSval=134486626 TSecr=288926804
292	107.146016802	192.168.4.195	34.107.243.93	TLSv1.3	696	Change Cipher Spec, Application Data
293	107.146292112	192.168.4.195	34.107.243.93	TLSv1.3	696	Application Data
294	107.164393713	8.8.8.8	192.168.4.195	DNS	139	Standard query response 0x3765 AAAA ipv4only.arpa SOA sns.dns.icann.org
295	107.164718988	192.168.4.195	8.8.8.8	DNS	73	Standard query 0xa6a A ipv4only.arpa
296	107.171727855	8.8.8.8	192.168.4.195	DNS	267	Standard query response 0xd785 AAAA detectportal.firefox.com CNAME detectportal.prod.mozilla.net CNAME prod.detectportal.prod.cloudops.mozgcp.net AAAA 2600:1901:0:38d7::
297	107.171727276	2404:6800:4009:827::...	2409:40c2:c:d3a1:f6...	OCSP	1268	Response
298	107.171727216	8.8.8.8	192.168.4.195	DNS	193	Standard query response 0x215d AAAA content-signature-2.cdn.mozilla.net CNAME content-signature-chains.prod.autograph.services.mozilla.net AAAA 2600:1901:0:92a9::
299	107.171705584	2409:40c2:c:d3a1:f6...	2404:6800:4009:827::...	TCP	86	44730 → 80 [ACK] Seq=436 Ack=183 Win=55024 Len=0 TSval=2106506070 TSecr=3018587114
300	107.172021354	192.168.4.195	8.8.8.8	DNS	84	Standard query 0xe0a9 A detectportal.firefox.com
301	107.172248807	2409:40c2:c:d3a1:f6...	2600:1901:0:92a9::...	TCP	84	52686 → 443 [RST] Seq=0 Win=0 Len=0 MSS=1440 SACK_PERM TSval=2975599322 TSecr=0 MS=128
302	107.173090337	2409:40c2:c:d3a1:f6...	2404:6800:4009:827::...	TLSv1.3	159	Change Cipher Spec, Application Data
303	107.173456982	2409:40c2:c:d3a1:f6...	2404:6800:4009:827::...	TLSv1.3	256	Application Data
304	107.173479569	2409:40c2:c:d3a1:f6...	2404:6800:4009:827::...	TLSv1.3	720	Application Data
305	107.173937788	8.8.8.8	192.168.4.195	DNS	183	Standard query response 0x8262 AAAA example.org AAAA 2600:1400:ec00:36::1736:7f2e AAAA 2600:1400:bc00:17::6007:810d AAAA 2600:1400:ec00:36::1736:7f2f AAAA 2600:1400:bc00:17::6007::
306	107.174259587	192.168.4.195	8.8.8.8	DNS	71	Standard query 0x04e4 A example.org
307	107.21188896	8.8.8.8	192.168.4.195	DNS	105	Standard query response 0xa6a A ipv4only.arpa A 192.0.0.171 A 192.0.0.170
308	107.21022179	192.168.4.195	8.8.8.8	DNS	73	Standard query 0x1c84 AAAA ipv4only.arpa

Frame 280: 181 bytes on wire (1448 bits), 181 bytes captured (1448 bits) on interface enp0s3, id 0

Ethernet II, Src: 3e:5f:a9:1d:62:20 (3e:5f:a9:1d:62:20), Dst: PcsCompu_04:76:1a (08:00:27:04:76:1a)

Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.4.195

User Datagram Protocol, Src Port: 53, Dst Port: 43880

Domain Name System (response)

Packets: 9708 · Displayed: 9708 (100.0%) Profile: Default

I filtered for packets sent to my own machine.

It means I was observing responses or traffic where my IP was the destination — which is normal during:

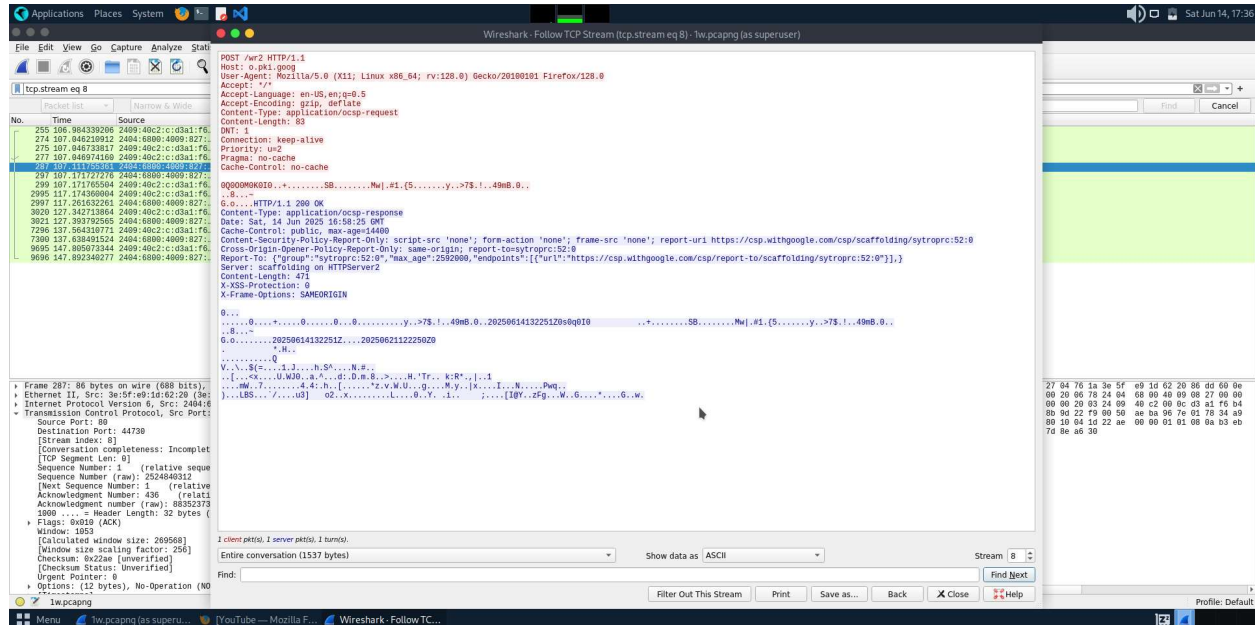
- **Browsing (HTTP/HTTPS)** — your machine receives data
- **DNS replies** — your system gets the resolved IP
- **TCP/UDP handshakes** — server responds to you
- **ICMP (ping)** — if someone pings you

The screenshot displays the Wireshark interface with a packet capture of a TCP connection. The top bar shows the application is running as 'tcpdump (as superuser)'. The packet list on the left shows a series of packets, with the selected packet being a TCP Reset (RST) packet from the server to the client. The packet details pane on the right shows the structure of the TCP segment, including the header and options. The packet bytes pane at the bottom shows the raw data of the packet.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
254	106.98340240	8.8.8.8	192.168.4.135	TCP	133	Standard query response 0x1b62 AAAA 0.pki.goog CNAME pki.goog L google.com AAAA 2484:6800:4009:827::2003
255	106.98349330	2484:6800:4009:827::2003	192.168.4.135	TCP	60	(SYN) Seq=4320 Len=0 Window=65535 SACK PERM TSval=210506053 TSecr=0 WS=120
256	106.98349875	192.168.4.135	2484:6800:4009:827::2003	TCP	74	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
257	106.99176080	192.168.4.135	2484:6800:4009:827::2003	TCP	60	57488 -> 443 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
258	107.00793981	192.168.4.135	2484:6800:4009:827::2003	TCP	1312	Client Hello [V1]
259	107.00809482	2484:6800:4009:827::2003	192.168.4.135	QUIC	1399	Protected Payload (KXP), CID=dec267
260	107.00809515	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
261	107.00809550	2484:6800:4009:827::2003	192.168.4.135	TCP	185	Protected Payload (KXP), CID=dec267
262	107.00809585	2484:6800:4009:827::2003	192.168.4.135	TCP	83	Protected Payload (KXP), CID=dec267
263	107.00809620	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
264	107.00809655	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
265	107.00809690	2484:6800:4009:827::2003	192.168.4.135	TCP	128	Application Data
266	107.00809725	2484:6800:4009:827::2003	192.168.4.135	TCP	128	Application Data
267	107.00809760	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
268	107.00809795	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
269	107.00809830	2484:6800:4009:827::2003	192.168.4.135	TCP	128	Application Data
270	107.00809865	2484:6800:4009:827::2003	192.168.4.135	TCP	128	Application Data
271	107.00809900	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
272	107.00809935	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
273	107.00809970	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
274	107.00810005	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
275	107.00810040	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
276	107.00810075	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
277	107.00810110	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
278	107.00810145	2484:6800:4009:827::2003	192.168.4.135	TCP	86	443 -> 57488 [ACK] Seq=6083 Ack=1559 Win=26956 Len=0 TSval=243448641 TSecr=243448641 WS=256
279	107.00810180	2484:6800:4009:827::2003	192.168.4.135	TCP</		

- **Protocol:** TCP
- **Layer:** Transport
- **Details:** SYN + ACK indicates a successful TCP handshake step
- **Insight:** Establishment of encrypted session possibly over HTTPS



- **Protocol:** HTTP
- **Layer:** Application
- **Details:** POST request to Google's OCSP server
- **Insight:** Shows certificate validation over HTTP

Applications Places System

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

fw.pcapng (as superuser)

Apply a display filter... <Ctrl>F

Display filter: ip.addr == 192.168.4.195

Find Cancel

No.	Time	Source	Destination	Protocol	Length	Info
270	107.018117696	192.168.4.195	8.8.8.8	DNS	73	Standard query 0x8868 A ipv4only.arp
271	107.024543398	192.168.4.195	8.8.8.8	DNS	71	Standard query 0x5764 A example.org
272	107.030991396	192.168.4.195	8.8.8.8	DNS	95	Standard query 0xe15a A content-signature-2.cdn.mozilla.net
273	107.032456986	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	QUIC	89	Protected Payload (KPo), DCID=ec267
274	107.040210912	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	TCP	89	44730 [SYN, ACK] Seq=0 Acl=1 Win=5525 Len=0 MSS=1380 SACK_PERM TSval=3018558984 TSecr=2105050593 WS=256
275	107.040133817	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	TCP	89	44730 -> 80 [ACK] Seq=1 Acl=1 Win=4384 Len=0 TSval=2105050655 TSecr=3018558984
276	107.040761467	192.168.4.195	8.8.8.8	DNS	84	Standard query 0x8399 A detectportal.firefox.com
277	107.040741509	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	OCSP	321	Request
278	107.078501748	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	TCP	86	443 -> 34796 [ACK] Seq=3554 Acl=949 Win=64128 Len=0 TSval=1266406831 TSecr=1533466415
279	107.083103692	34.107.243.93	192.168.4.195	TCP	66	443 -> 57488 [ACK] Seq=1 Acl=1247 Min=20776 Len=0 TSval=1269926884 TSecr=2434486481
280	107.085874102	8.8.8.8	192.168.4.195	DNS	181	Standard query response 0xe15a A content-signature-2.cdn.mozilla.net CNAME content-signature-chains.prod.autograph.services.mozilla.net A 34.160.144.191
281	107.085874388	8.8.8.8	192.168.4.195	DNS	195	Standard query response 0xe15a A ipv4only.arp A 192.0.0.171 A 192.0.0.170
282	107.085874427	8.8.8.8	192.168.4.195	DNS	130	Standard query response 0x5764 A example.org A 96.7.128.186 A 23.215.0.132 A 96.7.128.192
283	107.085978097	192.168.4.195	8.8.8.8	DNS	71	Standard query 0x8262 AAAA example.org
284	107.08600539	192.168.4.195	8.8.8.8	DNS	95	Standard query 0x215d AAAA content-signature-2.cdn.mozilla.net
285	107.086026926	192.168.4.195	8.8.8.8	DNS	73	Standard query 0x3765 AAAA ipv4only.arp
286	107.088119175	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	TCP	86	443 -> 33608 [ACK] Seq=606 Acl=1589 Min=26956 Len=0 TSval=4047395572 TSecr=463456819
287	107.111753361	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	TCP	86	80 -> 44730 [ACK] Seq=1 Acl=436 Min=26956 Len=0 TSval=3018587054 TSecr=2105050655
288	107.113028225	8.8.8.8	192.168.4.195	DNS	195	Standard query response 0x399 A detectportal.firefox.com CNAME detectportal.prod.cloudops.mozgcp.net CNAME prod.detectportal.prod.cloudops.mozgcp.net A 34.107.221.82
289	107.114048488	192.168.4.195	8.8.8.8	DNS	84	Standard query 0x785 AAAA detectportal.firefox.com
290	107.145230165	34.107.243.93	192.168.4.195	TLSv1.3	284	Server Hello, Change Cipher Spec, Application Data
291	107.145230259	192.168.4.195	34.107.243.93	TCP	66	57488 -> 443 [ACK] Seq=1247 Acl=218 Min=64128 Len=0 TSval=2434486626 TSecr=2898266864
292	107.146015802	192.168.4.195	34.107.243.93	TLSv1.3	130	Change Cipher Spec, Application Data
293	107.146012112	192.168.4.195	34.107.243.93	TLSv1.3	666	Application Data
294	107.164393713	8.8.8.8	192.168.4.195	DNS	130	Standard query response 0x3765 AAAA ipv4only.arp SOA sns.dns.icann.org
295	107.164719867	192.168.4.195	8.8.8.8	DNS	73	Standard query 0xe15a A ipv4only.arp
296	107.171777776	2000:1901::c:d3a1:f6	2000:1901::c:d3a1:f6	OCSP	1168	Response

Frame 296: 207 bytes on wire (1656 bits), 207 bytes captured (1656 bits) on interface enp0s3, id 0
Ethernet II, Src: 3a:5f:e9:1d:62:28 (3a:5f:e9:1d:62:28), Dst: PcsCompu_84:76:1a (08:00:27:84:76:1a)
Internet Protocol Version 6, Src: 8.8.8.8, Dst: 192.168.4.195
User Datagram Protocol, Src Port: 53, Dst Port: 47229
Domain Name System (response)
Transaction ID: 0x785
Flags: 0x180 Standard query response, No error
Questions: 1
Answer RRs: 3
Authority RRs: 0
Additional RRs: 0
Queries
Answers
[Request in: 289]
[Time: 0.0576557 seconds]

Ready to load or capture

Packets: 9708 Displayed: 9708 (100.0%)

Profile: Default

- Protocol: DNS
- Layer: Application
- Details: Response to DNS queries shows resolved IPv6 addresses
- Insight: Connection attempted with IPv6-capable services like Firefox/Cloud services

Applications Places System

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

fw.pcapng (as superuser)

Wireshark - Protocol Hierarchy Statistics - fw.pcapng (as superuser)

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDU's
Frame	100.0	9708	100.0	9618024	516 k	0	0	0	9708
Ethernet	100.0	9708	1.4	136122	7156	0	0	0	9708
Internet Protocol Version 6	73.7	7150	1.9	286000	15 k	0	0	0	7150
User Datagram Protocol	61.6	5977	0.5	47816	2514	0	0	0	5977
QUIC, IETF	61.6	5963	65.4	6420063	337 k	5963	6389917	335 k	6006
Network Time Protocol	0.1	8	0.0	384	20	8	384	20	8
Multicast Domain Name System	0.1	6	0.0	748	39	6	748	39	6
Transmission Control Protocol	11.9	1153	6.7	635961	34 k	639	48000	2523	1153
Transport Layer Security	4.7	456	6.7	658929	34 k	456	563165	29 k	532
Hypertext Transfer Protocol	0.4	16	0.4	51844	2270	1	319	16	56
Online Certificate Status Protocol	0.6	56	0.2	15612	820	56	15612	820	56
Line-based text data	0.0	1	0.0	8	0	1	8	0	1
Internet Control Message Protocol v6	0.2	20	0.0	708	37	20	708	37	20
Internet Protocol Version 4	26.2	2544	0.5	50880	2675	0	0	0	2544
User Datagram Protocol	2.4	234	0.0	1872	98	0	0	0	234
Network Time Protocol	0.1	8	0.0	384	20	8	384	20	8
NetBIOS Name Service	0.1	6	0.0	300	15	6	300	15	6
Multicast Domain Name System	0.1	6	0.0	746	39	6	746	39	6
Dynamic Host Configuration Protocol	0.0	4	0.0	1200	63	4	1200	63	4
Domain Name System	2.2	210	0.2	15424	810	210	15424	810	210
Transmission Control Protocol	23.8	2310	22.6	2214027	116 k	1774	1713347	90 k	2310
Transport Layer Security	5.5	532	22.3	2190765	115 k	532	1969057	103 k	559
Hypertext Transfer Protocol	0.0	4	0.0	1068	56	2	636	33	4
Line-based text data	0.0	2	0.0	16	0	2	16	0	2
Address Resolution Protocol	0.1	14	0.0	536	28	14	536	28	14

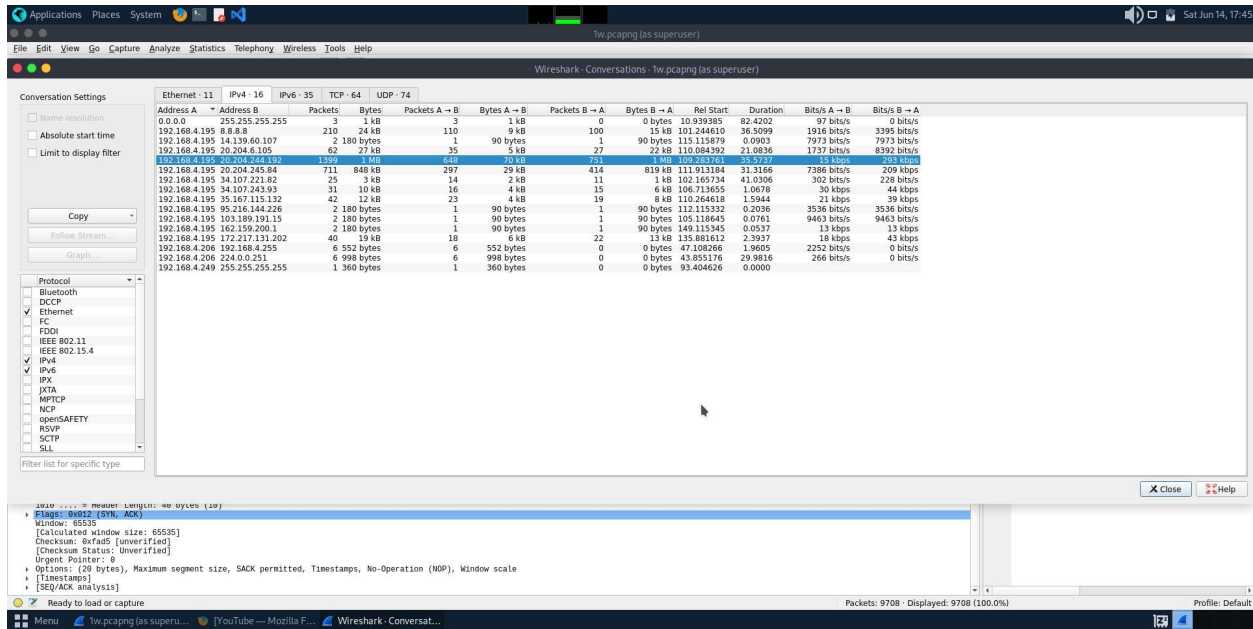
Close Copy Help

Ready to load or capture

Packets: 9708 Displayed: 9708 (100.0%)

Profile: Default

- **Protocol:** Summary (All captured protocols)
- **Layer:** All layers
- **Details:** Shows percentage of each protocol observed (e.g., 65% TCP, 22% DNS)
- **Insight:** Helps understand traffic behavior and protocol usage distribution



Conversations View

Go to: Statistics → Conversations → IPv4

🔍 We'll see:

- All IPs your machine talked to
- How much data was sent/received
- Useful for identifying **which site or server** had most activity

OSI MODEL

Layer	Name	Function	Protocols/Examples	Data Unit	Key Devices
7	Application	User interfaces, network services	HTTP, FTP, SMTP, DNS, SSH	Data	Gateways, Firewalls

Layer	Name	Function	Protocols/Examples	Data Unit	Key Devices
		(HTTP, email, file transfers).			
6	Presentation	Data translation, encryption, compression (e.g., JPEG, SSL/TLS).	SSL/TLS, JPEG, MPEG, ASCII	Data	-
5	Session	Manages connections (setup, maintenance, termination).	NetBIOS, RPC, SIP	Data	-
4	Transport	End-to-end communication (reliability, flow control, error correction).	TCP, UDP, SCTP	Segment (TCP)	Firewalls, Load Balancers
3	Network	Logical addressing and routing (IP, routers).	IP, ICMP, OSPF, BGP, IPv4/IPv6	Packet	Routers, L3 Switches
2	Data Link	Physical addressing (MAC), error detection (switches, bridges).	Ethernet, PPP, VLANs, MAC	Frame	Switches, NICs
1	Physical	Transmits raw bitstream over physical media (cables, wireless).	USB, DSL, Fiber, IEEE 802.11 (Wi-Fi)	Bit	Hubs, Repeater

Phishing Incident Response Playbook

1. Preparation

- Conduct awareness training for all employees.
- Deploy anti-phishing email filters.
- Ensure MFA (Multi-Factor Authentication) is enabled.
- Maintain updated contact info for SOC/IT teams.

2. Identification

- **Source:** Report from user or automatic detection tool (e.g., email flagged).
- **Signs of phishing:**
 - Suspicious sender email
 - Urgent request for credentials
 - Malicious link or attachment
- **Tools to use:**
 - Email header analyzer
 - VirusTotal (for attachment or URL)
 - SIEM alert dashboard

3. Containment

- Instruct user **not to interact** with the email.
- Block sender domain at email gateway.
- Quarantine or delete the email.
- Revoke access tokens or reset password if credentials were entered.

4. Eradication

- Delete phishing email from all affected inboxes.
- Clean infected endpoints (if link/attachment executed malware).
- Block phishing domain/IP at firewall or DNS level.

5. Recovery

- Restore clean email backups (if needed).
- Re-enable user access (if locked).
- Monitor the system and user activity closely for 24–72 hours.

6. Lessons Learned

- Document the timeline of the phishing incident.
- Review what detection/prevention worked and what failed.
- Update playbooks, detection rules, and training material.
- Share anonymized case with team for future awareness.

LOG PARSING SCRIPTS

1. Linux SSH Failed Login Parser (journal or auth.log)

Python:-

```
import re

with open("ssh_failures.log", "r") as file:
    for line in file:
        match = re.search(r"Failed password for( invalid user)? (\w+) from ([\d.]+) port (\d+)", line)
        if match:
            user = match.group(2)
            ip = match.group(3)
            port = match.group(4)
            print(f"User: {user} | IP: {ip} | Port: {port}")
```

Use case: After running

Bash:-

```
journalctl _COMM=sshd | grep "Failed password" > ssh_failures.log
```

2. Apache Access Log Parser

For /var/log/apache2/access.log:

Python:-

```
import re

log_file = "access.log"

with open(log_file, "r") as file:
    for line in file:
```



```

        match = re.search(r'(\d+\.\d+\.\d+\.\d+) - - \[(.*?)\] "GET (.*?)
HTTP/.*?" (\d+)', line)
    if match:
        ip = match.group(1)
        datetime = match.group(2)
        page = match.group(3)
        status = match.group(4)
        print(f"[{datetime}] {ip} requested {page} → Status: {status}")

```

Use case: Tracks who visited which page and what HTTP status they received.

3. Generic Keyword Alert System (e.g., for detecting 'error')

Python:-

```

keywords = ["error", "unauthorized", "denied", "failed"]
log_file = "syslog.log" # Use any log file

with open(log_file, "r") as file:
    for line in file:
        if any(keyword.lower() in line.lower() for keyword in keywords):
            print(f"[ALERT] {line.strip()}")

```

Use case: Can be pointed at any log and used to find critical alerts.