

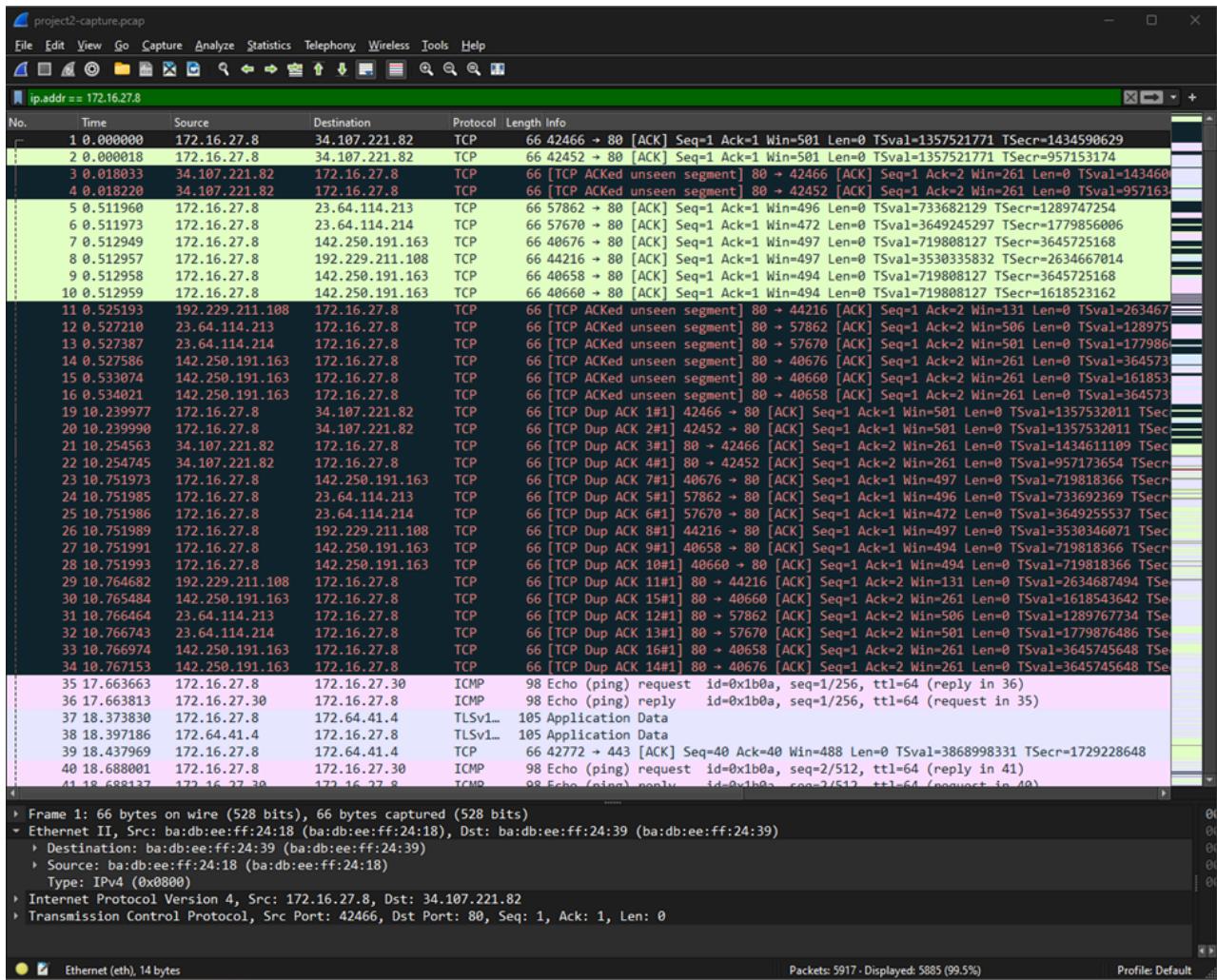
Project -2



Preparing people to lead extraordinary lives

OBAID RAZA ID: 1677274

1. What is the MAC address of the device being watched? (hex in colon-separated format)



Filter Used: ip.addr == 172.16.27.8

- ba:db:ee:ff:24:18

Packet Selection:

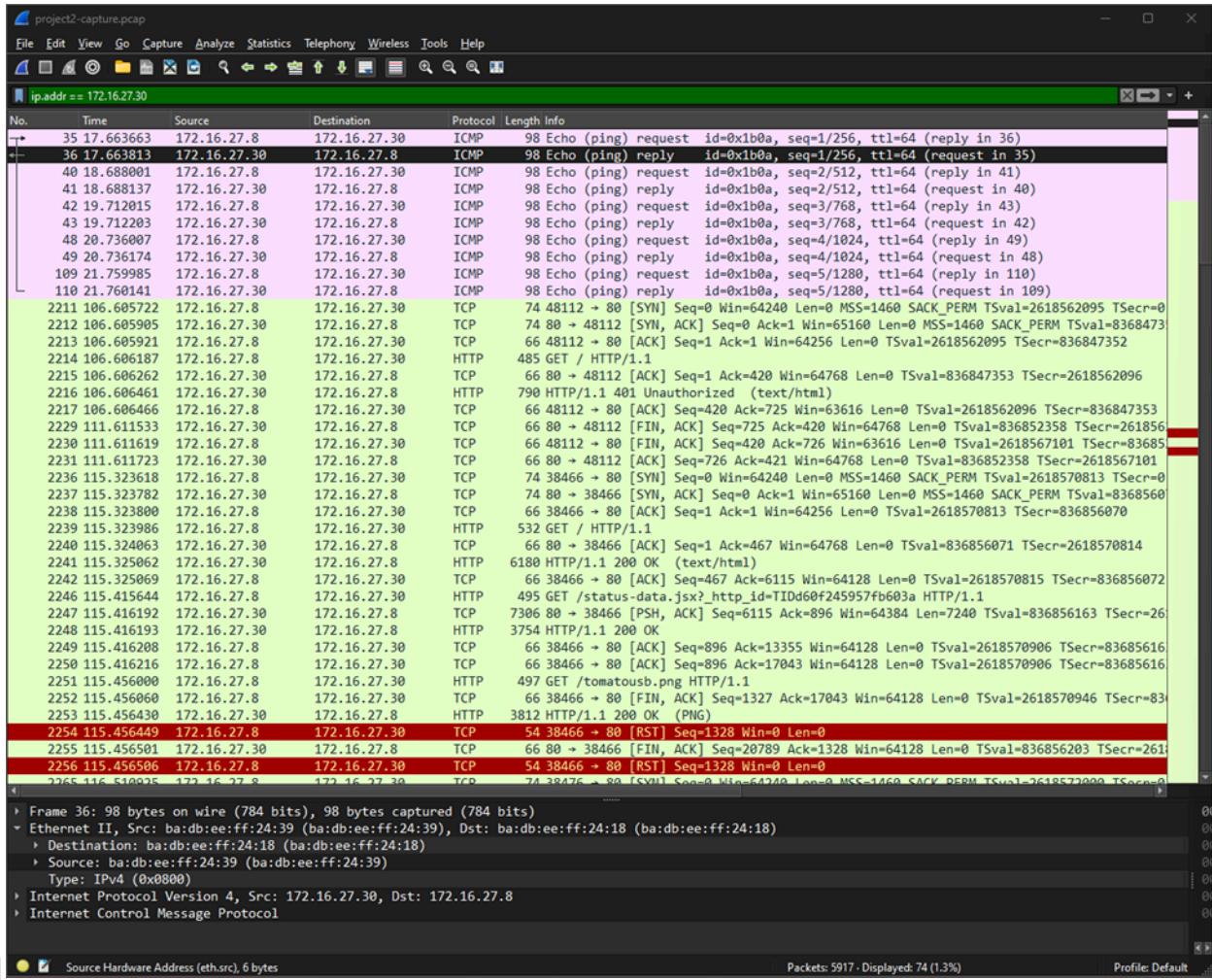
- The first packet involving the device (**172.16.27.8**) is **Frame 1**.
- This packet contains both the Source MAC Address and Destination MAC Address fields.

Interpretation:

- In Frame 1, under the Ethernet II header, the Source MAC Address is listed as **ba:db:ee:ff:24:18**.
- This MAC address (ba:db:ee:ff:24:18) represents the device being monitored, as it is consistently the source or destination MAC in subsequent frames related to **172.16.27.8**.

Conclusion: Based on the captured data, the MAC address of the device being watched is ba:db:ee:ff:24:18, confirmed by its role as the Source MAC Address in Frame

2. What is the MAC address of the gateway device? (hex in colon-separated format)



Answer for Gateway MAC Address:

- **Gateway Device MAC Address: ba:db:ee:ff:24:39**
- 1. Filter Used: ip.addr == 172.16.27.8

Packet Selection:

- In Frame 36, we observe an ICMP (ping) reply with Source MAC Address **ba:db:ee:ff:24:39** and Destination MAC Address **ba:db:ee:ff:24:18**.
- This source MAC address **ba:db:ee:ff:24:39** is identified as the gateway since it responds to a request from the monitored device **172.16.27.8**.

2. Interpretation:

- Since ba:db:ee:ff:24:39 is the source MAC address in the response to a ping request from 172.16.27.8, it indicates that this address belongs to the gateway, which is responding to the device's requests.

3. Conclusion:

- The MAC address ba:db:ee:ff:24:39 is confirmed to be the gateway device, as seen in the ICMP reply packet in Frame 36.

3. What was the IP of the first device that was pinged (ICMP Echo Request)?

No.	Time	Source	Destination	Protocol	Length	Info
35	17.663663	172.16.27.8	172.16.27.30	ICMP	98	Echo (ping) request id=0x1b0a, seq=1/256, ttl=64 (reply in 36)
36	17.663813	172.16.27.30	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b0a, seq=1/256, ttl=64 (request in 35)
40	18.668001	172.16.27.8	172.16.27.30	ICMP	98	Echo (ping) request id=0x1b0a, seq=2/512, ttl=64 (reply in 41)
41	18.668137	172.16.27.30	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b0a, seq=2/512, ttl=64 (request in 40)
42	19.712015	172.16.27.8	172.16.27.30	ICMP	98	Echo (ping) request id=0x1b0a, seq=3/768, ttl=64 (reply in 43)
43	19.712203	172.16.27.30	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b0a, seq=3/768, ttl=64 (request in 42)
48	20.736007	172.16.27.8	172.16.27.30	ICMP	98	Echo (ping) request id=0x1b0a, seq=4/1024, ttl=64 (reply in 49)
49	20.736174	172.16.27.30	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b0a, seq=4/1024, ttl=64 (request in 48)
109	21.759985	172.16.27.8	172.16.27.30	ICMP	98	Echo (ping) request id=0x1b0a, seq=5/1280, ttl=64 (reply in 110)
110	21.760141	172.16.27.30	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b0a, seq=5/1280, ttl=64 (request in 109)
113	29.767286	172.16.27.8	8.8.4.4	ICMP	98	Echo (ping) request id=0x1b15, seq=1/256, ttl=64 (reply in 114)
114	29.784062	8.8.4.4	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b15, seq=1/256, ttl=116 (request in 113)
115	30.769158	172.16.27.8	8.8.4.4	ICMP	98	Echo (ping) request id=0x1b15, seq=2/512, ttl=64 (reply in 116)
116	30.784137	8.8.4.4	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b15, seq=2/512, ttl=116 (request in 115)
133	31.770281	172.16.27.8	8.8.4.4	ICMP	98	Echo (ping) request id=0x1b15, seq=3/768, ttl=64 (reply in 134)
134	31.785516	8.8.4.4	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b15, seq=3/768, ttl=116 (request in 133)
135	32.771624	172.16.27.8	8.8.4.4	ICMP	98	Echo (ping) request id=0x1b15, seq=4/1024, ttl=64 (reply in 136)
136	32.789048	8.8.4.4	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b15, seq=4/1024, ttl=116 (request in 135)
137	33.773180	172.16.27.8	8.8.4.4	ICMP	98	Echo (ping) request id=0x1b15, seq=5/1280, ttl=64 (reply in 138)
138	33.800553	8.8.4.4	172.16.27.8	ICMP	98	Echo (ping) reply id=0x1b15, seq=5/1280, ttl=116 (request in 137)
186	60.374822	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=1/256, ttl=1 (no response found!)
187	60.374841	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=2/512, ttl=1 (no response found!)
188	60.374850	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=3/768, ttl=1 (no response found!)
189	60.374858	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=4/1024, ttl=2 (no response found!)
190	60.374865	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=5/1280, ttl=2 (no response found!)
191	60.374869	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=6/1536, ttl=2 (no response found!)
192	60.374871	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=7/1792, ttl=3 (no response found!)
193	60.374873	172.16.27.8	40.89.244.232	ICMP	74	Echo (ping) request id=0x1b20, seq=8/2048, ttl=3 (no response found!)

Frame 35: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: ba:db:ee:ff:24:18 (ba:db:ee:ff:24:18), Dst: ba:db:ee:ff:24:39 (ba:db:ee:ff:24:39)
Internet Protocol Version 4, Src: 172.16.27.8, Dst: 172.16.27.30
Internet Control Message Protocol

Answer:

- **First Device Pinged IP Address:172.16.27.30**

1. Filter Used: icmp

2. Packet Selection: 35

3. Interpretation:

- Since this is the first ICMP Echo Request in the capture, the destination IP 40.89.244.232 is identified as the first device pinged by the monitored device.

4. Conclusion:

- **The IP address of the first device that was pinged (ICMP Echo Request) by 172.16.27.30, confirmed by examining packet 35 and filter ICMP.**

4. What DNS server(s) is/are being used to resolve names to IPs?

Time	Source	Destination	Protocol	Length	Info
266 60.464584	8.8.8.8	172.16.27.8	DNS	160	Standard query response 0xe590 PTR 101.28.216.96.in-addr.arpa PTR po-326-346-rur30
256 60.437536	8.8.8.8	172.16.27.8	DNS	96	Standard query response 0xa0df No such name PTR 3.162.112.10.in-addr.arpa OPT
234 60.489483	8.8.8.8	172.16.27.8	DNS	95	Standard query response 0x9fc2 No such name PTR 1.27.16.172.in-addr.arpa OPT
185 60.374658	8.8.8.8	172.16.27.8	DNS	150	Standard query response 0x692e AAAA duckduckgo.com SOA dns1.p05.nsone.net OPT
162 45.510203	8.8.8.8	172.16.27.8	DNS	150	Standard query response 0x295f AAAA duckduckgo.com SOA dns1.p05.nsone.net OPT
160 45.489043	8.8.8.8	172.16.27.8	DNS	168	Standard query response 0xeaac AAAA www.duckduckgo.com CNAME duckduckgo.com SOA dn
159 45.475295	8.8.8.8	172.16.27.8	DNS	119	Standard query response 0x6011 A www.duckduckgo.com CNAME duckduckgo.com A 48.89.2
140 39.118870	8.8.8.8	172.16.27.8	DNS	436	Standard query response 0x6980 AAAA connectivity-check.ubuntu.com AAAA 2620:2d:400
96 20.847847	8.8.8.8	172.16.27.8	DNS	141	Standard query response 0x5dee AAAA ipv4only.arpa SOA sns.dns.icann.org OPT
92 20.828881	8.8.8.8	172.16.27.8	DNS	141	Standard query response 0x9555 AAAA ipv4only.arpa SOA sns.dns.icann.org OPT
79 20.792082	8.8.8.8	172.16.27.8	DNS	218	Standard query response 0x2c08 AAAA detectportal.firefox.com CNAME detectportal.pr
78 20.781862	8.8.8.8	172.16.27.8	DNS	206	Standard query response 0x9848 A detectportal.firefox.com CNAME detectportal.prod.
5861 464.949826	172.16.27.8	8.8.8.8	DNS	116	Standard query 0x31fa AAAA telemetry-incoming.r53-2.services.mozilla.com OPT
5828 464.755959	172.16.27.8	8.8.8.8	DNS	116	Standard query 0x98aa AAAA telemetry-incoming.r53-2.services.mozilla.com OPT
5825 464.728038	172.16.27.8	8.8.8.8	DNS	101	Standard query 0x273e AAAA incoming.telemetry.mozilla.org OPT
5824 464.727789	172.16.27.8	8.8.8.8	DNS	101	Standard query 0x6660 A incoming.telemetry.mozilla.org OPT
4696 429.088847	172.16.27.8	8.8.8.8	DNS	100	Standard query 0x1e80 A connectivity-check.ubuntu.com OPT
3997 375.423049	172.16.27.8	8.8.8.8	DNS	97	Standard query 0x9380 AAAA mozilla.cloudflare-dns.com OPT
3996 375.422990	172.16.27.8	8.8.8.8	DNS	97	Standard query 0x4345 A mozilla.cloudflare-dns.com OPT
3871 339.116340	172.16.27.8	8.8.8.8	DNS	100	Standard query 0x9f6f AAAA connectivity-check.ubuntu.com OPT
2328 129.123759	172.16.27.8	8.8.8.8	DNS	100	Standard query 0xe4eb A connectivity-check.ubuntu.com OPT
337 69.207665	172.16.27.8	8.8.8.8	DNS	85	Standard query 0xfb05 MX duckduckgo.com OPT
335 69.180529	172.16.27.8	8.8.8.8	DNS	85	Standard query 0x0ed9 AAAA duckduckgo.com OPT
317 60.975307	172.16.27.8	8.8.8.8	DNS	97	Standard query 0x24e3 PTR 232.244.89.40.in-addr.arpa OPT
308 60.711210	172.16.27.8	8.8.8.8	DNS	97	Standard query 0xc231 PTR 246.54.44.104.in-addr.arpa OPT
302 60.691831	172.16.27.8	8.8.8.8	DNS	96	Standard query 0xd3f0 PTR 145.16.10.51.in-addr.arpa OPT
300 60.672231	172.16.27.8	8.8.8.8	DNS	97	Standard query 0x47a5 PTR 222.28.44.104.in-addr.arpa OPT
295 60.634977	172.16.27.8	8.8.8.8	DNS	96	Standard query 0x808c PTR 18.11.44.104.in-addr.arpa OPT
285 60.604215	172.16.27.8	8.8.8.8	DNS	97	Standard query 0x457f PTR 23.237.44.104.in-addr.arpa OPT
279 60.588022	172.16.27.8	8.8.8.8	DNS	98	Standard query 0x4c97 PTR 178.117.248.50.in-addr.arpa OPT
276 60.560089	172.16.27.8	8.8.8.8	DNS	97	Standard query 0x158a PTR 210.33.110.96.in-addr.arpa OPT
273 60.543995	172.16.27.8	8.8.8.8	DNS	96	Standard query 0x1214 PTR 49.40.110.96.in-addr.arpa OPT
271 60.520124	172.16.27.8	8.8.8.8	DNS	96	Standard query 0x5d60 PTR 85.177.85.68.in-addr.arpa OPT
269 60.491867	172.16.27.8	8.8.8.8	DNS	96	Standard query 0x832f PTR 57.197.86.68.in-addr.arpa OPT
267 60.465150	172.16.27.8	8.8.8.8	DNS	96	Standard query 0x1d30 PTR 37.235.87.68.in-addr.arpa OPT
257 60.439088	172.16.27.8	8.8.8.8	DNS	97	Standard query 0xe590 PTR 101.28.216.96.in-addr.arpa OPT
249 60.422379	172.16.27.8	8.8.8.8	DNS	96	Standard query 0xa0df PTR 3.162.112.10.in-addr.arpa OPT
224 60.391889	172.16.27.8	8.8.8.8	DNS	95	Standard query 0x9fc2 PTR 1.27.16.172.in-addr.arpa OPT
184 60.359232	172.16.27.8	8.8.8.8	DNS	85	Standard query 0x692e AAAA duckduckgo.com OPT

Frame 5861: 116 bytes on wire (928 bits), 116 bytes captured (928 bits)
 Ethernet II, Src: ba:db:ee:ff:24:18 (ba:db:ee:ff:24:18), Dst: ba:db:ee:ff:24:39 (ba:db:ee:ff:24:39)
 Internet Protocol Version 4, Src: 172.16.27.8, Dst: 8.8.8.8
 User Datagram Protocol, Src Port: 35665, Dst Port: 53
 Domain Name System (query)
 Transaction ID: 0x31fa
 Flags: 0x0100 Standard query
 Questions: 1
 Answer RRs: 0

Packets: 5917 - Displayed: 70 (1.2%) Profile: Default

Filter Used: dns

Answer: The monitored device is using Google's DNS server 8.8.8.8

5 SOA records:

- sns.dns.icann.org
- dns1.p05.nsone.net
- dns101.comcast.net
- ns1-06.azure-dns.com
- ns1-06.azure-dns.com

Packet Inspection:

- In Frame 5861, a DNS Standard Query is sent from the monitored device 172.16.27.8 to 8.8.8.8 with a request for telemetry-incoming.r53-2.services.mozilla.com.
- Observing other DNS query packets, we consistently see that the DNS server IP 8.8.8.8 is used as the destination for these queries.

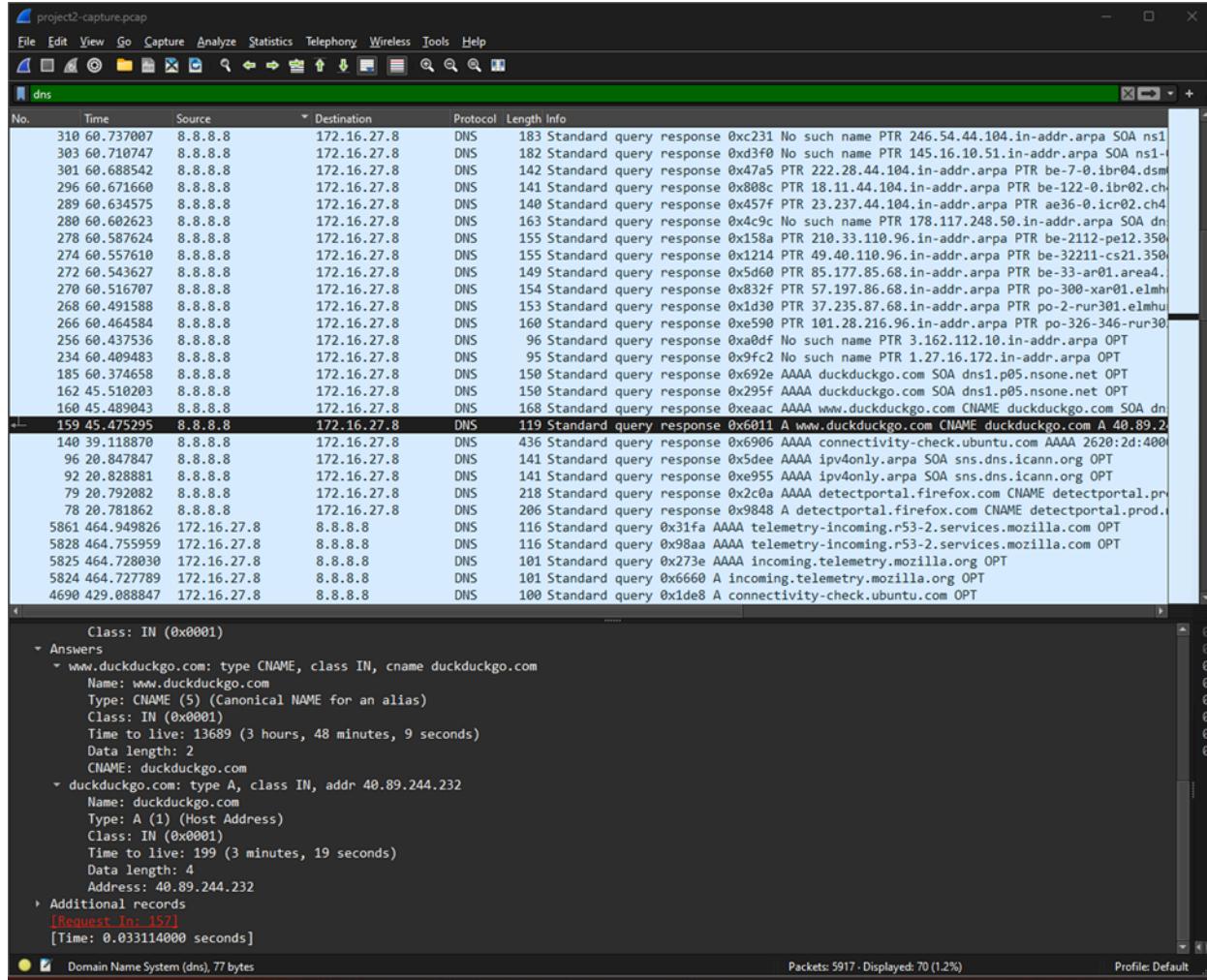
Interpretation:

- The repeated appearance of 8.8.8.8 as the destination IP for DNS queries indicates that this is the primary DNS server used by the monitored device for name resolution.

Conclusion:

- The monitored device is using Google's DNS server 8.8.8.8 to resolve domain names to IP addresses, as evidenced by the destination address in DNS query packets, such as in Frame 5861.

5. According to DNS in the capture, what IPv4 address hosts the duckduckgo.com website (A record)?



Answer:

- IPv4 Address for **duckduckgo.com: 40.89.244.232**
- Filter Used: dns**
 - Packet Inspection:** In Frame 159, a DNS response for the query www.duckduckgo.com contains two answers:
 - The first answer provides a CNAME (Canonical Name) record, indicating that www.duckduckgo.com is an alias for duckduckgo.com.
 - The second answer provides an A (Address) record for duckduckgo.com, giving the IPv4 address as 40.89.244.232.

3. Interpretation:

- Since the DNS response includes the A record for duckduckgo.com, we can conclude that 40.89.244.232 is the IP address hosting duckduckgo.com.

4. Conclusion:

- The IPv4 address **40.89.244.232** is confirmed as the host IP for duckduckgo.com, as shown in Frame 159 of the DNS response.

6. What account was used to log into the router device via HTTP? (username and password)

The screenshot shows the NetworkMiner interface with the following details:

Project Information: project2-capture.pcap

File Menu: File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Toolbar: Includes icons for File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help, and various search and filter functions.

Selected Filter: http

Table Headers: No., Time, Source, Destination, Protocol, Length, Info

Captured Packets List:

No.	Time	Source	Destination	Protocol	Length	Info
80	20.794928	34.107.221.82	172.16.27.8	HTTP	363	HTTP/1.1 200 OK (text/html)
2334	129.369955	185.125.190.48	172.16.27.8	HTTP	255	HTTP/1.1 204 No Content
4735	439.356001	172.16.27.8	172.16.27.6	HTTP	547	GET / HTTP/1.1
4695	429.147531	172.16.27.8	91.189.91.97	HTTP	154	GET / HTTP/1.1
3998	375.434629	172.16.27.8	172.16.27.6	HTTP	407	GET /favicon.ico HTTP/1.1
3989	375.361028	172.16.27.8	172.16.27.6	HTTP	456	GET / HTTP/1.1
3971	367.503418	172.16.27.8	172.16.27.6	HTTP	409	GET / HTTP/1.1
2638	134.066016	172.16.27.8	142.250.191.227	OCSP	486	Request
2612	134.036182	172.16.27.8	142.250.191.227	OCSP	486	Request
2435	133.664134	172.16.27.8	142.250.191.227	OCSP	487	Request
2495	133.257511	172.16.27.8	142.250.191.227	OCSP	486	Request
2390	133.141637	172.16.27.8	142.250.191.133	HTTP	411	GET / HTTP/1.1
2333	129.251139	172.16.27.8	185.125.190.48	HTTP	154	GET / HTTP/1.1
2318	126.626641	172.16.27.8	172.16.27.30	HTTP	613	POST /status-data.jsx?_http_id=TIId60f245957fb603a HTTP/1.1 (text/plain)
2302	121.622418	172.16.27.8	172.16.27.30	HTTP	613	POST /status-data.jsx?_http_id=TIId60f245957fb603a HTTP/1.1 (text/plain)
2268	116.511229	172.16.27.8	172.16.27.30	HTTP	613	POST /status-data.jsx?_http_id=TIId60f245957fb603a HTTP/1.1 (text/plain)
2251	115.456000	172.16.27.8	172.16.27.30	HTTP	497	GET /tomatousb.png HTTP/1.1
2246	115.415644	172.16.27.8	172.16.27.30	HTTP	495	GET /status-data.jsx?_http_id=TIId60f245957fb603a HTTP/1.1
2239	115.323986	172.16.27.8	172.16.27.30	HTTP	532	GET / HTTP/1.1
2214	106.606187	172.16.27.8	172.16.27.30	HTTP	485	GET / HTTP/1.1
557	81.988821	172.16.27.8	40.89.244.232	HTTP	416	GET / HTTP/1.1
486	77.686572	172.16.27.8	142.250.191.227	OCSP	487	Request
457	77.286111	172.16.27.8	142.250.191.227	OCSP	487	Request
412	76.963993	172.16.27.8	142.250.191.227	OCSP	487	Request
83	20.797014	172.16.27.8	34.107.221.82	HTTP	369	[TCP Previous segment not captured] GET /success.txt?ipv4 HTTP/1.1
61	20.760417	172.16.27.8	34.107.221.82	HTTP	367	[TCP Previous segment not captured] GET /canonical.html HTTP/1.1
4737	439.356758	172.16.27.6	172.16.27.8	HTTP	948	HTTP/1.1 200 OK (text/html)
4000	375.435175	172.16.27.6	172.16.27.8	HTTP	555	HTTP/1.1 404 Not Found (text/html)

Selected Packet Details: Frame 2318: 613 bytes on wire (4904 bits), 613 bytes captured (4904 bits)

Selected Headers:

Field	Value
Protocol	Hypertext Transfer Protocol
Method	POST
Path	/status-data.jsx?_http_id=TIId60f245957fb603a
Version	HTTP/1.1
Content-Type	text/plain
Content-Length	613
Host	172.16.27.30
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/116.0
Accept	*/*
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Content-Type	text/plain;charset=UTF-8
Content-Length	28
Origin	http://172.16.27.30
Authorization	Basic c3N0ZXZlbmNvbjphYmMxMjM=
Connection	keep-alive
Referer	http://172.16.27.30
Cookie	tomato_menu_status=overview.asp; tomato_status_overview_refresh=5

Bottom Status Bar: Hypertext Transfer Protocol (http), 519 bytes | Packets: 5917 - Displayed: 48 (0.8%) | Profile: Default

Answer:

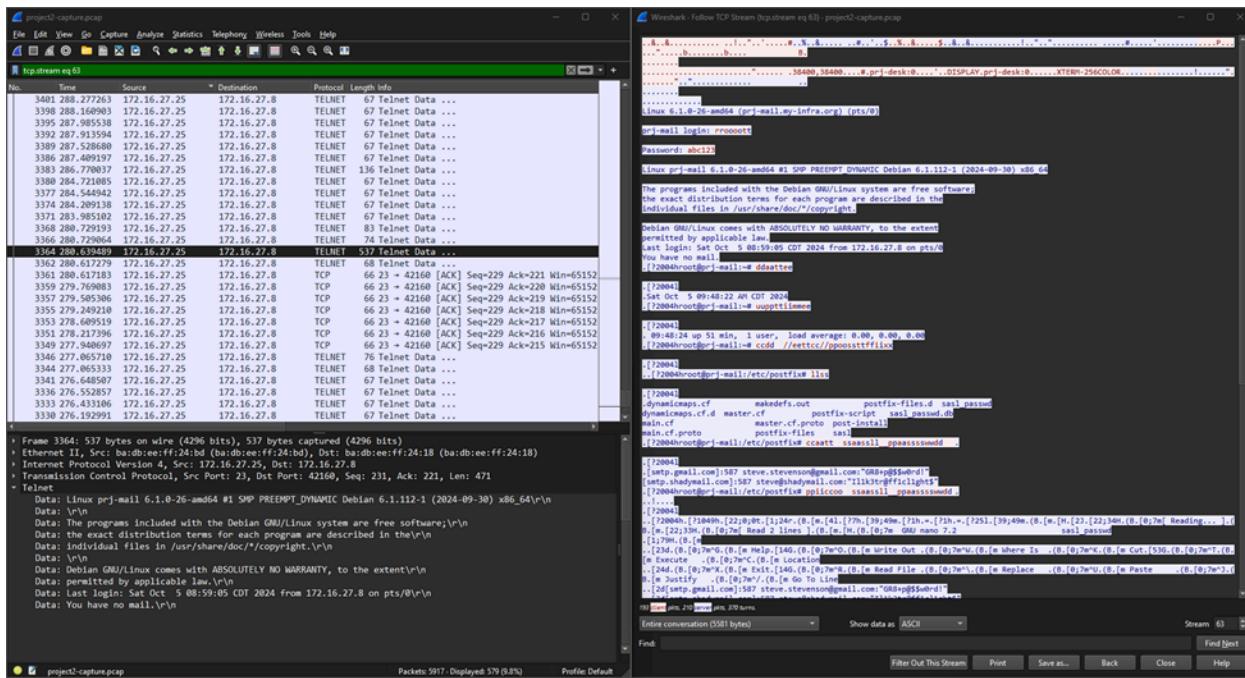
- Username: sstevenson
 - Password: abc123

Packet Number: Frame 2302

Filter Used: http && ip.addr == 172.16.27.30

Explanation: In Frame 2302, the Authorization header is valid with credentials in the Basic Authorization format. The decoded string for the encoded c3N0ZXZlbnNvbjhYmMxMjM= is sstevenson:abc123. This will provide the credentials used to log into the router. It is sent in a POST to <http://172.16.27.30/status-data.jsx>.

7. What account was used to log into the local mail server via TELNET? (username and password)



Answer:

- Username: root
- Password: abc123

1. Filter Used: telnet && ip.addr == 172.16.27.25

2. Packet Inspection:

- In Frame 3364 within the TELNET session (stream 63), we can observe the login attempt.
- The session data includes a prompt for login: where the username root is entered.
- Shortly afterward, we see Password: followed by the password entry abc123.

3. Interpretation:

- The root user logs in with the password abc123, as shown clearly in the TELNET session data.

4. Conclusion:

- Username: root
- Password: abc123

8. At the time of the capture, how long had the local mail server been up (uptime found in TELNET)?

Answer:

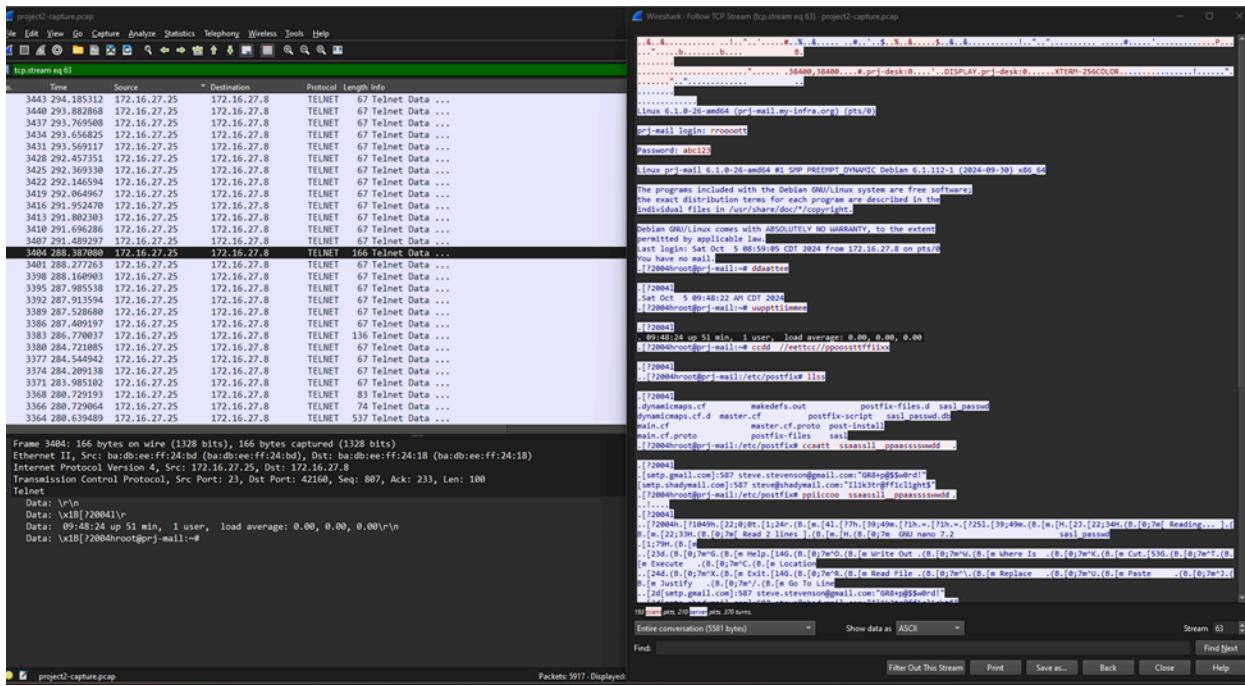
- Uptime: 51 minutes

1. Filter Used: telnet && ip.addr == 172.16.27.25

2. Packet Inspection:

- In Frame 3364 within the TELNET stream (stream 63), we observe the user executing the uptime command.

- The output in this frame indicates up 51 min, which reflects how long the server had been running at the time of capture.



9. Who did the target send e-mail as and to during the interaction via SMTP and what was the content of the message? (provide the from and to email addresses)

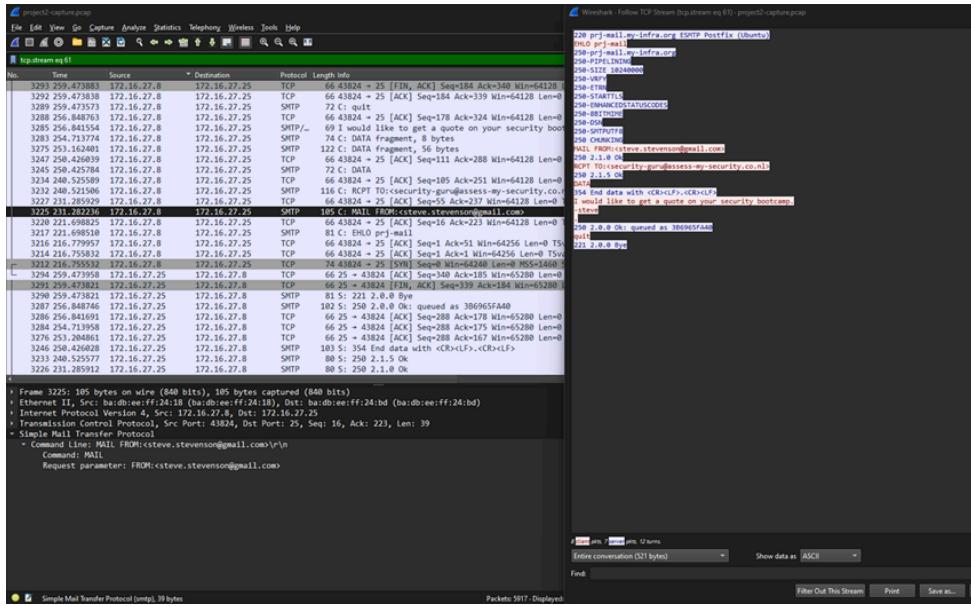
Answer:

- From: steve.stevenson@gmail.com
- To: security-guru@assess-my-security.co.nl
- Message Content: "I would like to get a quote on your security bootcamp. -steve"

1. Filter Used: smtp && ip.addr == 172.16.27.25

2. Packet Inspection:

- In Frame 3299 and subsequent frames within the SMTP session (stream 61), the email transaction is visible:
 - The MAIL FROM command indicates the sender: steve.stevenson@gmail.com.
 - The RCPT TO command reveals the recipient: security-guru@assess-my-security.co.nl.
- The DATA section of the SMTP conversation shows the message content: **I would like to get a quote on your security bootcamp. -steve**



10. What version of the Linux kernel is the local mail server running, which can be found via TELNET interactions? (it is of the format “Linux 6.x.y-z-amd64”)

Answer:

- **Linux Kernel Version: Linux 6.1.0-26-amd64**

1. **Filter Used: telnet && ip.addr == 172.16.27.25**

2. **Packet Inspection:**

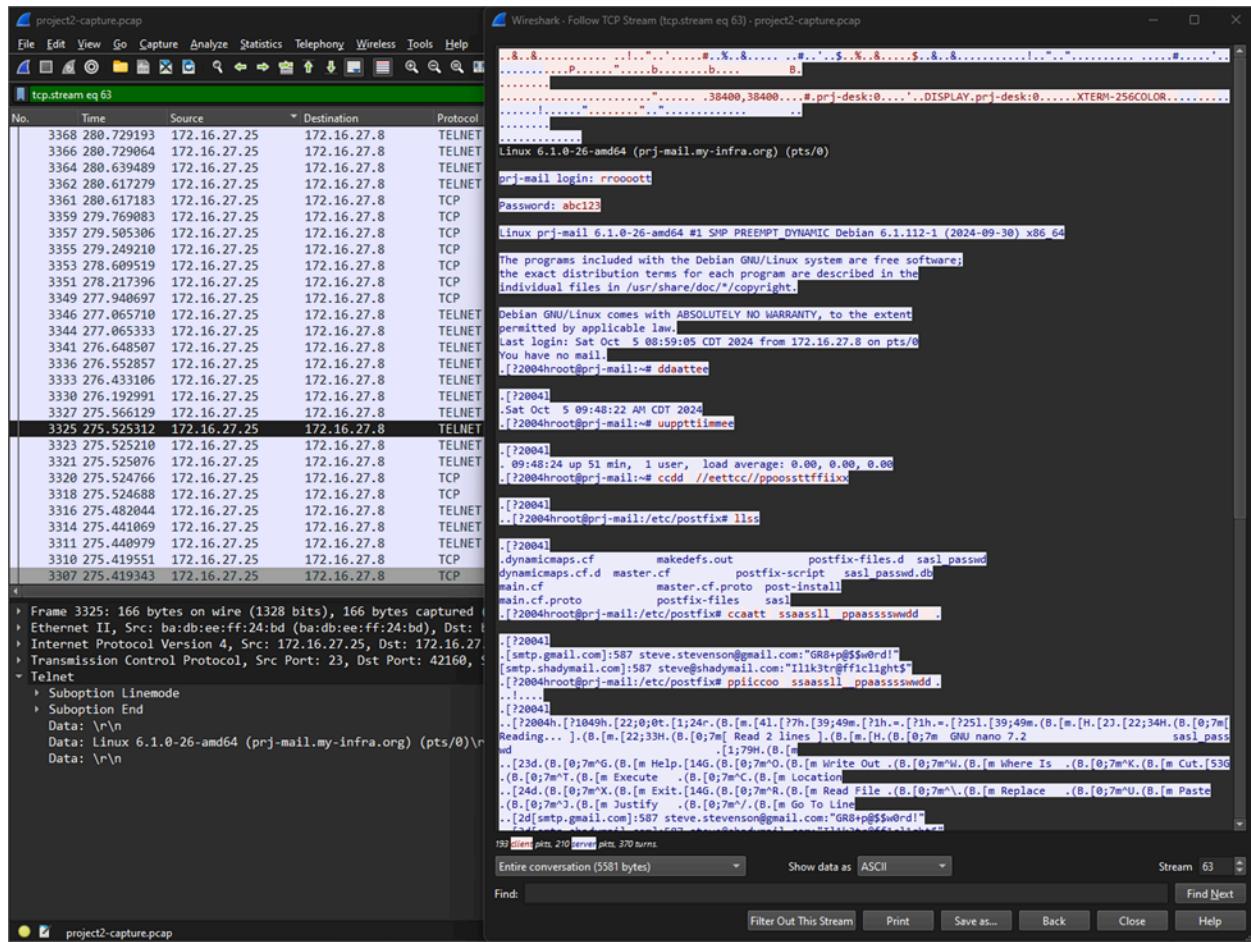
- In Frame 3325, the TELNET session output displays system information after login.

- The Linux kernel version is indicated in the line:

Linux 6.1.0-26-amd64 (prj-mail.my-infra.org) (pts/0)

3. **Interpretation:**

- This line in the TELNET session provides the exact kernel version 6.1.0-26-amd64.



11. According to the HTTP2 GET requests, embedded in the HEADERS, what was the first, complete search request sent by the user, via www.duckduckgo.com (they require the use of the ssl keys provided)?

Frame Number: 1464 and 1471

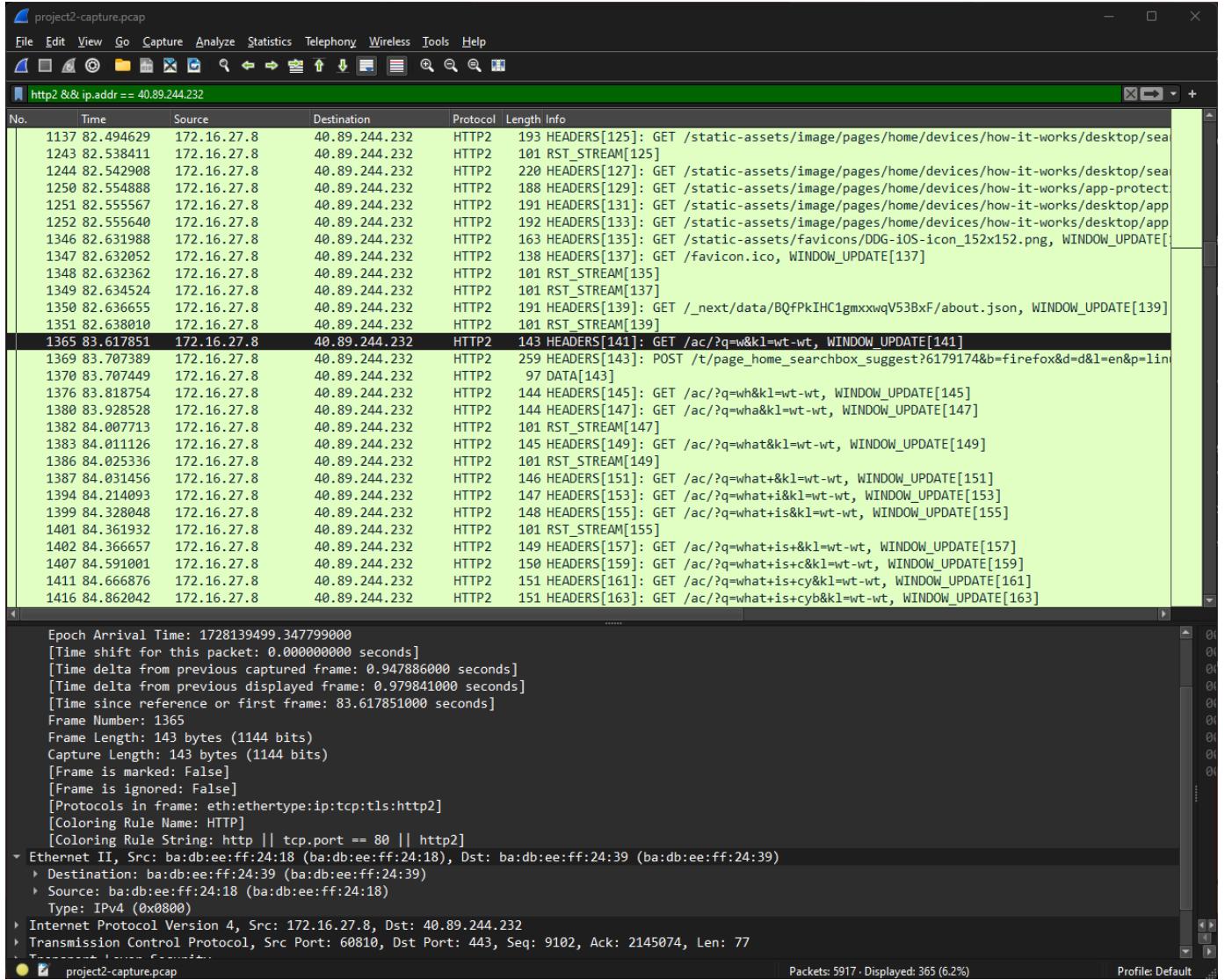
Filter Applied: http2 && ip.addr == 40.89.244.232

Search Query: q=what is cyber security

Interpretation: The HTTP2 GET request in Frame 1365 includes the search query parameter q=w in the path /ac/?q=what is cyber security &kl=wt-wt, identifying it as the first search request sent by the user to DuckDuckGo. The packet captures the full details of the request, including HTTP2 headers that specify :method: GET and :authority: duckduckgo.com. This frame is the earliest in the capture with a complete search query.

Answer:

- First Complete Search Request: q=what is cyber security
- Found in Frame: 1464 and 1471



12. According to the HTTP2 GET requests, embedded in the HEADERS what was the last, complete search request sent by the user, via www.duckduckgo.com (they require the use of the ssl keys provided)?

Answer:

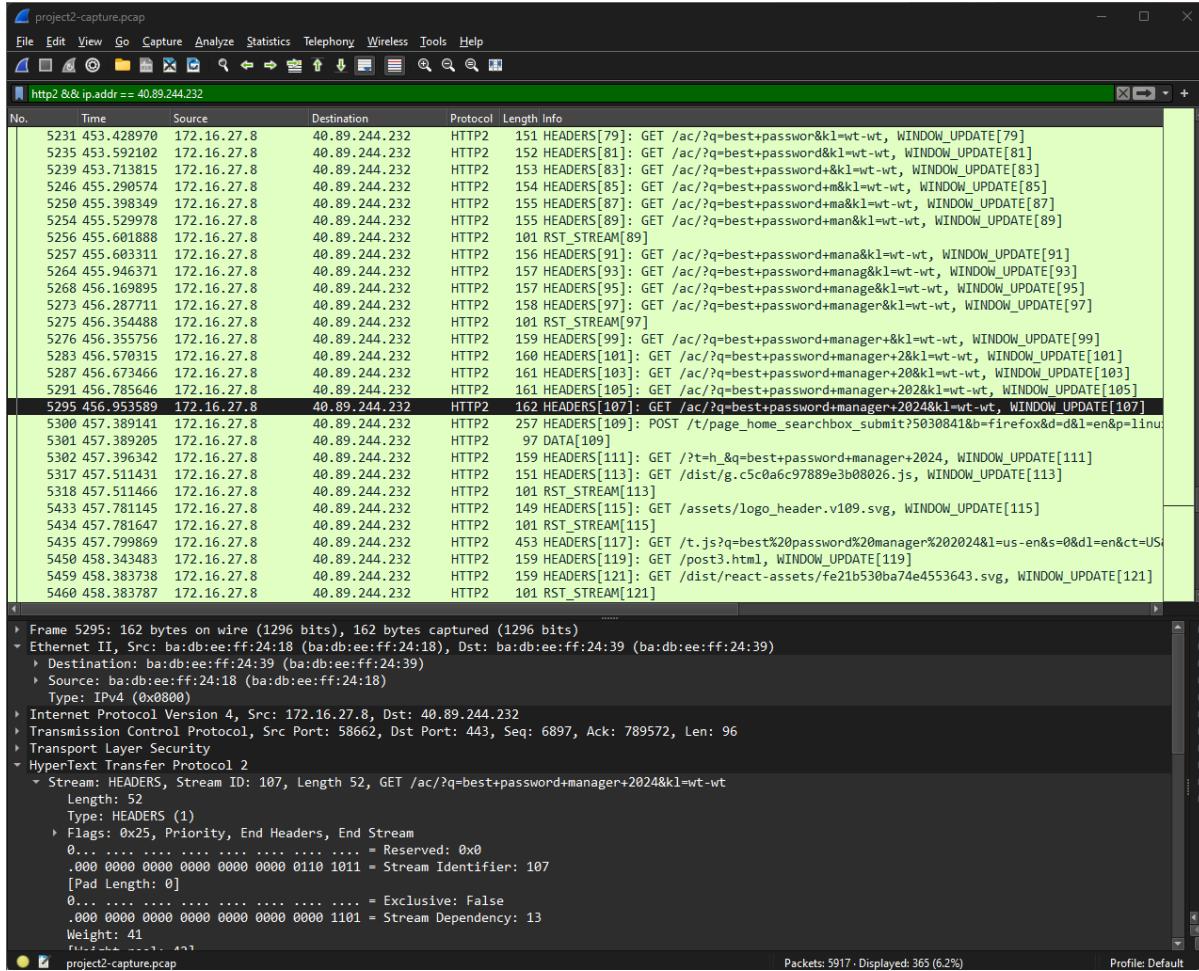
- **Last Complete Search Request:** `q=best+password+manager+2024`
- Found in Frame: **5295 & 5300**
- **Filter Used:** `http2 && ip.addr == 40.89.244.232`

Explanation: To identify the last complete search request made by the user on DuckDuckGo, I applied the filter `http2 && ip.addr == 40.89.244.232` to isolate HTTP2 traffic directed to DuckDuckGo's IP address.

Packet Inspection: In Frame 5295, the HTTP2 GET request displays the following :path header:

`/ac/?q=best+password+manager+2024&k1=wt-wt`

- This path indicates that the search query was for "best password manager 2024".



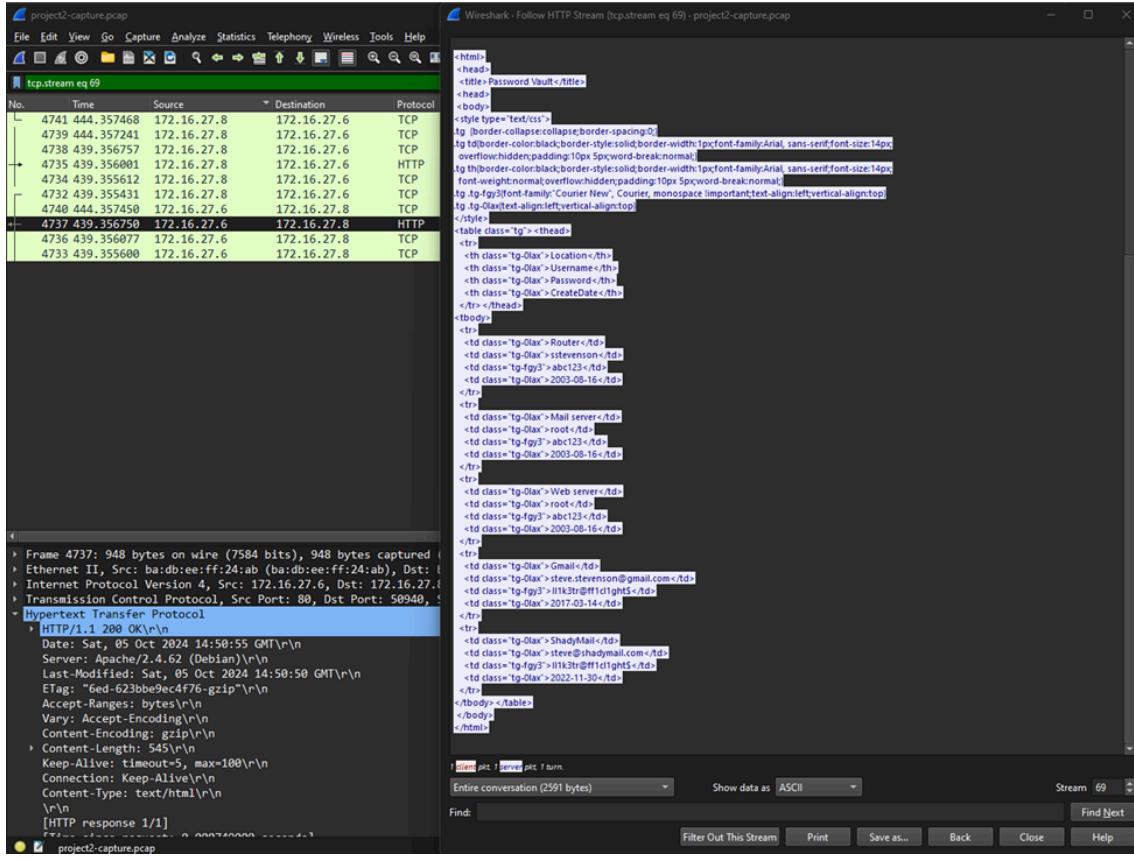
13. What information was provided on the webpage loaded from the local web server?

The data was found in **Frame 4737**, as shown in the HTTP response from the web server at Port 50940.

The response was captured in **http && ip.addr == 172.16.27.6**

The webpage titled "Password Vault" on the local web server displayed sensitive information in a table format. It included:

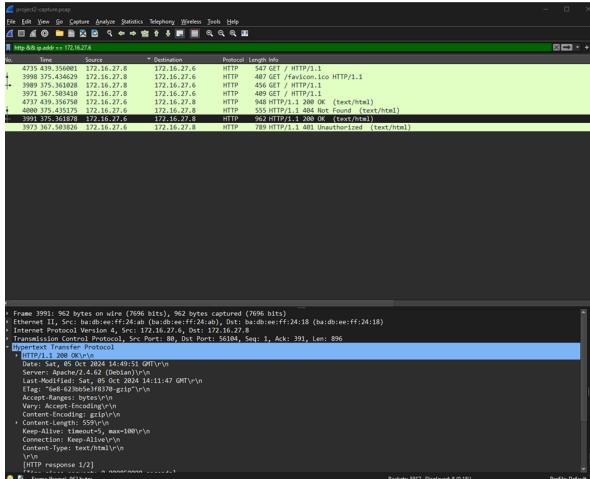
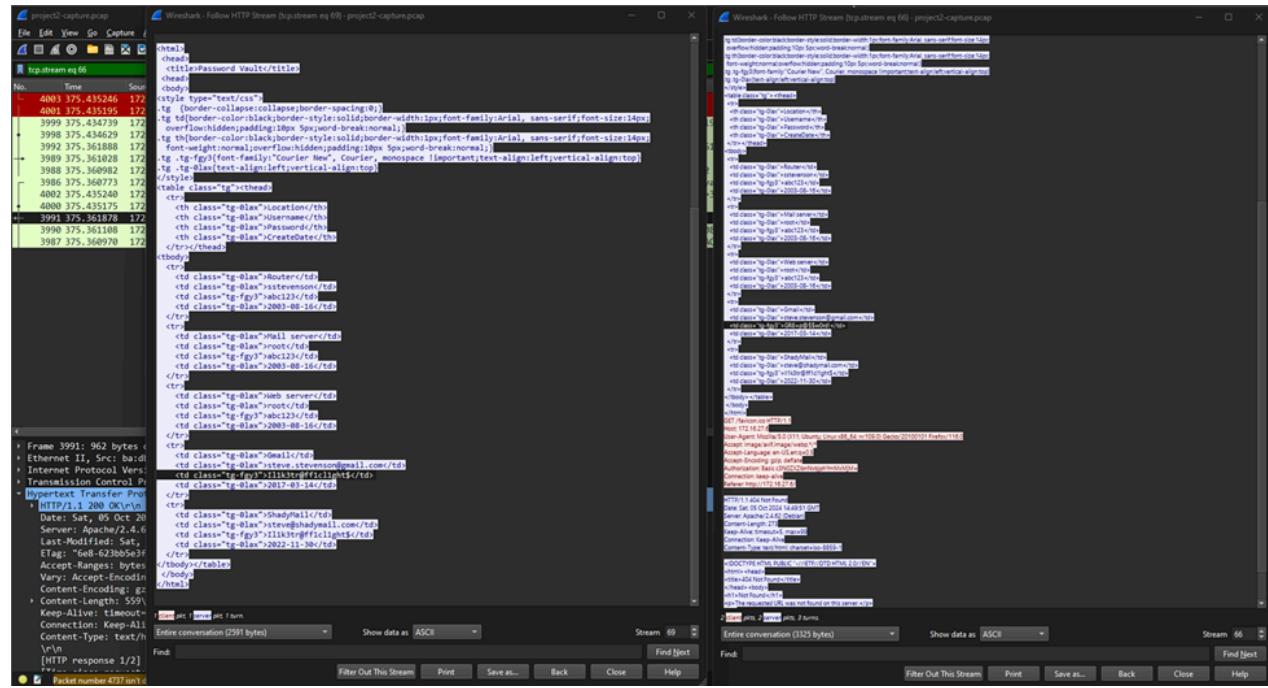
Location , Username , Password, CreateDate



Details from the Table:

- Router
 - Username: sstevenson
 - Password: abc123
 - CreateDate: 2003-08-16
- Mail server
 - Username: root
 - Password: abc123
 - CreateDate: 2003-08-16
- Web server
 - Username: root
 - Password: abc123
 - CreateDate: 2003-08-16
- Gmail
 - Username: steve.stevenson@gmail.com
 - Password: ll1k3tr@ff1cl1ght\$
 - CreateDate: 2017-03-14
- ShadyMail
 - Username: steve@shadymail.com
 - Password: ll1k3tr@ff1cl1ght\$
 - CreateDate: 2022-11-30

14. What record on the local web server was changed (two requests for the same page ... you will need to compare)?



The "Password Vault" webpage on the local web server showed a modification between two requests, specifically in the Gmail entry's password.

1. Gmail Password Update:

- First Request: The password for Gmail was "GR8+p@\$\$w0rd!".
- Second Request: The password was updated to "ll1k3tr@ff1cl1ght\$".
- Interpretation: This change indicates a recent update to the Gmail record stored on the server.

2. ETag Modification:

- First Request ETag: "6e8-623bb5e3f8370-gzip"
- Second Request ETag: "6ed-623bbe9ec4f76-gzip"

- Interpretation: The ETag change confirms that the content of the resource changed between requests, reflecting the Gmail password update.

Relevant Packet Numbers:

- First Request Packet: **Frame 3991**
- Second Request Packet: **Frame 4737**
- Filter:**http && ip.addr == 172.16.27.6**

15. What vendor made the local router and what model is it, which can be found via HTTP interactions?

The screenshot shows a Wireshark capture window titled "project2-capture.pcap". The display filter is set to "http && ip.addr == 172.16.27.30". The main pane displays 2322 captured packets. The details pane shows the expanded content of packet 2322, which is an HTTP response. The response is from an Apache/2.4.62 (Debian) server. The "Date" header indicates the response was sent on Saturday, October 5, 2024, at 14:45:42 GMT. The "Server" header identifies the server as Apache. The "Last-Modified" and "ETag" headers provide information about the last modification of the resource. The "Accept-Ranges" header specifies that the server supports byte ranges. The "Content-Length" header indicates the size of the response body. The "Keep-Alive" and "Connection" headers indicate the server's support for persistent connections. The body of the response contains the requested file data, which is identified as "tomatousb.png".

In the HTTP interaction data, specifically in packet number 2322, the local router's vendor and model information can be identified. Here's how this was determined:

- **Packet Number: 2322 , Display Filter: http && ip.addr == 172.16.27.30**
- **Vendor: FreshTomato**
 - **Explanation:** The router_name and wl_ssid fields in the packet data are set to "FreshTomato." FreshTomato is a custom firmware commonly used on specific router brands, indicating the device is running this firmware.
- **Model: Linksys E2500 v2**

- Explanation: The model information is directly listed under the `t_model_name` field in the packet, showing "Linksys E2500 v2" as the model.