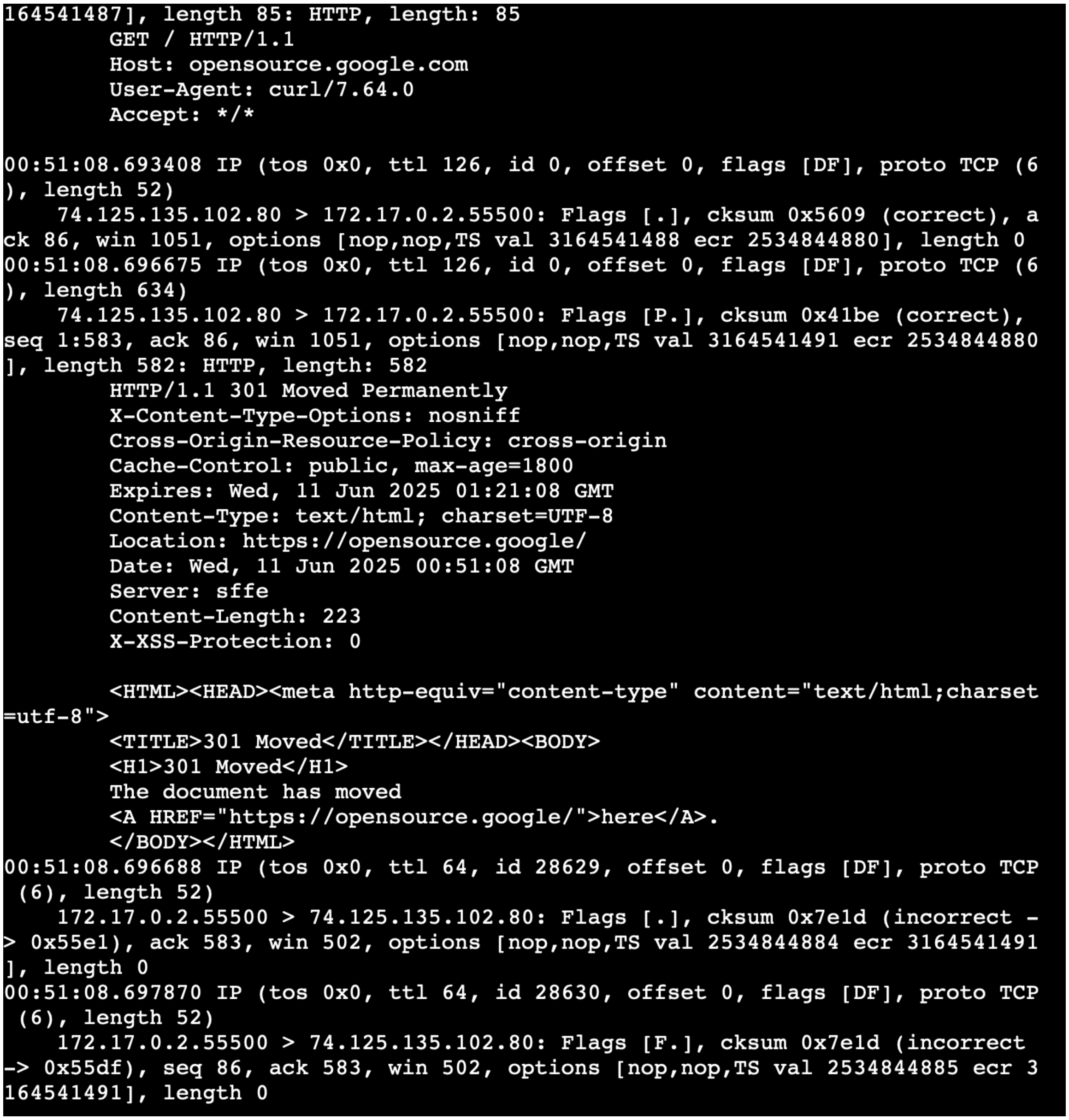
📡 PROJECT 5: PACKET CAPTURE & NETWORK ANALYSIS

Tools Used: tcpdump, Wireshark

Focus: Network traffic inspection, protocol analysis, packet filtering, and evidence review

📷



```
164541487], length 85: HTTP, length: 85
        GET / HTTP/1.1
        Host: opensource.google.com
        User-Agent: curl/7.64.0
        Accept: */*

00:51:08.693408 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6
), length 52)
    74.125.135.102.80 > 172.17.0.2.55500: Flags [.], cksum 0x5609 (correct), a
ck 86, win 1051, options [nop,nop,TS val 3164541488 ecr 2534844880], length 0
00:51:08.696675 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6
), length 634)
    74.125.135.102.80 > 172.17.0.2.55500: Flags [P.], cksum 0x41be (correct),
seq 1:583, ack 86, win 1051, options [nop,nop,TS val 3164541491 ecr 2534844880
], length 582: HTTP, length: 582
        HTTP/1.1 301 Moved Permanently
        X-Content-Type-Options: nosniff
        Cross-Origin-Resource-Policy: cross-origin
        Cache-Control: public, max-age=1800
        Expires: Wed, 11 Jun 2025 01:21:08 GMT
        Content-Type: text/html; charset=UTF-8
        Location: https://opensource.google/
        Date: Wed, 11 Jun 2025 00:51:08 GMT
        Server: sffe
        Content-Length: 223
        X-XSS-Protection: 0

        <HTML><HEAD><meta http-equiv="content-type" content="text/html;charset
=utf-8">
        <TITLE>301 Moved</TITLE></HEAD><BODY>
        <H1>301 Moved</H1>
        The document has moved
        <A HREF="https://opensource.google/">here</A>.
        </BODY></HTML>
00:51:08.696688 IP (tos 0x0, ttl 64, id 28629, offset 0, flags [DF], proto TCP
 (6), length 52)
    172.17.0.2.55500 > 74.125.135.102.80: Flags [.], cksum 0x7e1d (incorrect -
> 0x55e1), ack 583, win 502, options [nop,nop,TS val 2534844884 ecr 3164541491
], length 0
00:51:08.697870 IP (tos 0x0, ttl 64, id 28630, offset 0, flags [DF], proto TCP
 (6), length 52)
    172.17.0.2.55500 > 74.125.135.102.80: Flags [F.], cksum 0x7e1d (incorrect
-> 0x55df), seq 86, ack 583, win 502, options [nop,nop,TS val 2534844885 ecr 3
164541491], length 0
```

Description: Starting a packet capture using tcpdump in Linux

What I Did:

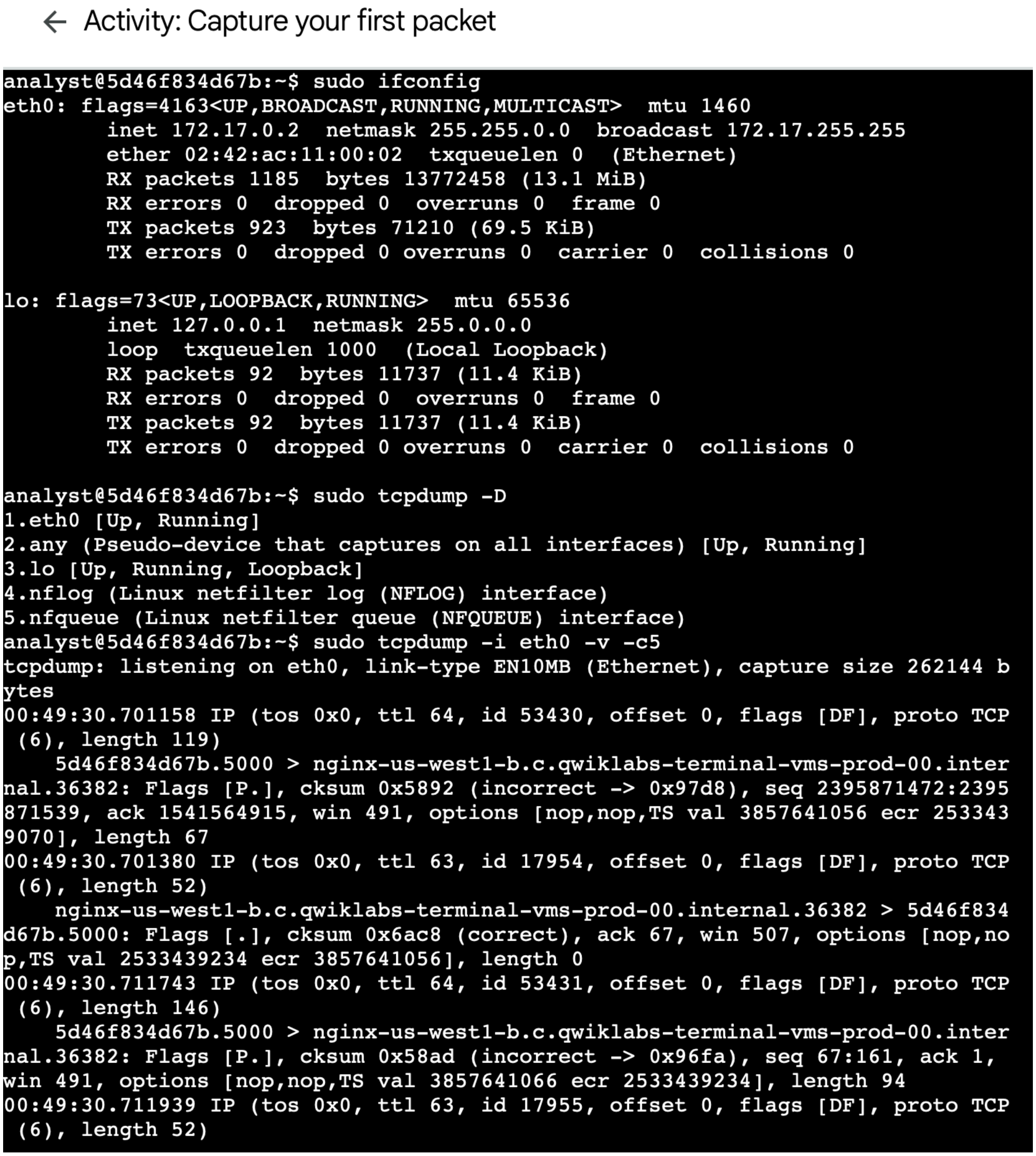
Executed a command to begin packet capture:

```
sudo tcpdump -i eth0 -w capture.pcap
```

Skills Used:

- CLI-based packet sniffing
- Saving live traffic for analysis
- Selecting correct network interface

📷



← Activity: Capture your first packet

```
analyst@5d46f834d67b:~$ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1460
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 1185  bytes 13772458 (13.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 923  bytes 71210 (69.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 92  bytes 11737 (11.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 92  bytes 11737 (11.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

analyst@5d46f834d67b:~$ sudo tcpdump -D
1.eth0 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.nflog (Linux netfilter log (NFLOG) interface)
5.nfqueue (Linux netfilter queue (NFQUEUE) interface)
analyst@5d46f834d67b:~$ sudo tcpdump -i eth0 -v -c5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 b
ytes
00:49:30.701158 IP (tos 0x0, ttl 64, id 53430, offset 0, flags [DF], proto TCP
 (6), length 119)
    5d46f834d67b.5000 > nginx-us-west1-b.c.qwiklabs-terminal-vms-prod-00.inter
nal.36382: Flags [P.], cksum 0x5892 (incorrect -> 0x97d8), seq 2395871472:2395
871539, ack 1541564915, win 491, options [nop,nop,TS val 3857641056 ecr 253343
9070], length 67
00:49:30.701380 IP (tos 0x0, ttl 63, id 17954, offset 0, flags [DF], proto TCP
 (6), length 52)
    nginx-us-west1-b.c.qwiklabs-terminal-vms-prod-00.internal.36382 > 5d46f834
d67b.5000: Flags [.], cksum 0x6ac8 (correct), ack 67, win 507, options [nop,no
p,TS val 2533439234 ecr 3857641056], length 0
00:49:30.711743 IP (tos 0x0, ttl 64, id 53431, offset 0, flags [DF], proto TCP
 (6), length 146)
    5d46f834d67b.5000 > nginx-us-west1-b.c.qwiklabs-terminal-vms-prod-00.inter
nal.36382: Flags [P.], cksum 0x58ad (incorrect -> 0x96fa), seq 67:161, ack 1,
win 491, options [nop,nop,TS val 3857641066 ecr 2533439234], length 94
00:49:30.711939 IP (tos 0x0, ttl 63, id 17955, offset 0, flags [DF], proto TCP
 (6), length 52)
```

Description: Live terminal output showing captured TCP/IP traffic

What I Did:

Observed source/destination IPs, protocols, and packet size in real-time.

Skills Used:

- Reading raw traffic in terminal
- Recognizing common ports and protocols
- Interpreting basic TCP flags

📷

Description: Opened .pcap file in Wireshark GUI

What I Did:

Loaded the file into Wireshark for visual inspection of packet content and session details.

Skills Used:

- Using Wireshark to inspect protocol headers
- Navigating packet layers (Ethernet, IP, TCP/UDP)
- Session analysis

📷

Description: Deep packet inspection of HTTP or TCP traffic in Wireshark

What I Did:

Zoomed in on individual packet payloads, reviewed hex/ASCII views.

Skills Used:

- Interpreting payloads
- Analyzing suspicious patterns
- Validating data exposure (e.g., passwords)

📷

Description: Wireshark display filter applied

What I Did:

Filtered traffic to zoom in on key IPs and ports using:

```
tcp.port == 80
ip.addr == 10.0.2.15
```

Skills Used:

- Packet filtering by IP and port
- Creating effective Wireshark rules
- Isolating attack-related traffic

📷

Description: Scrollable overview of all packets captured

What I Did:

Reviewed total traffic and searched for unusual behavior like repeated ARP requests or unrecognized outbound flows.

Skills Used:

- Traffic flow analysis
- Detecting anomalies
- Recon pattern identification

🔍 Project Summary

Category	Skills Used
Packet Sniffing	Used tcpdump to capture and export .pcap files
GUI Analysis	Navigated Wireshark layers and filters
Traffic Filtering	Applied port/IP filters to zoom into relevant activity
Threat Detection	Detected traffic anomalies and explored session payloads

---

💡 Real-World Benefit

1. **SOC Capability**: I can conduct network forensics during incidents, examining attack patterns and payloads.
2. **Network Defense**: Able to spot port scanning, cleartext transmissions, or unauthorized IP connections.
3. **Protocol Mastery**: Comfortable navigating TCP/IP layers and recognizing malicious signatures or traffic spikes.

Let me know when you're ready for **Project 6: Vulnerability & Risk Assessment Reports** — the next step in simulating real-world audits and controls documentation.