Capture the packet

Scenario

You’re a network analyst who needs to use `tcpdump` to capture and analyze live network traffic from a Linux virtual machine.

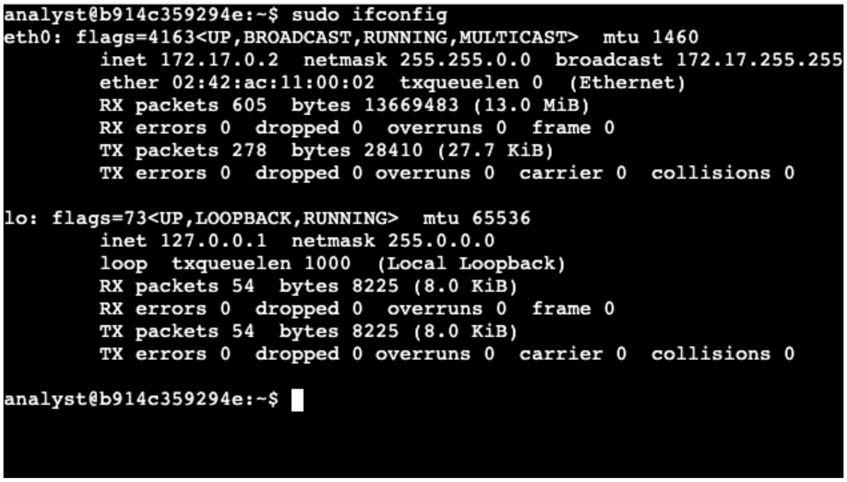
The lab starts with your user account, called `analyst`, already logged in to a Linux terminal.

In this lab activity, you’ll perform tasks associated with using tcpdump to capture network traffic. You’ll capture the data in a packet capture (p-cap) file and then examine the contents of the captured packet data to focus on specific types of traffic.

Task 1. Identify network interfaces

1.1. Identify the interfaces that are available:

Command: sudo ifconfig

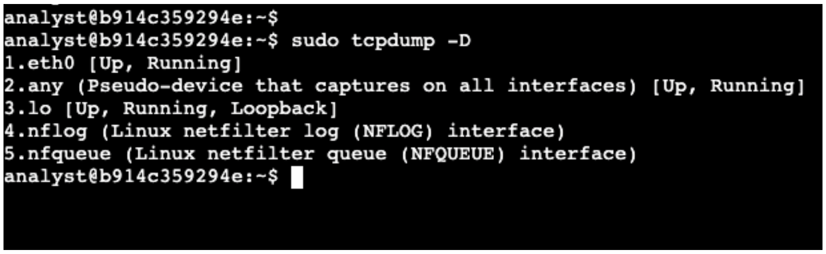


```
analyst@b914c359294e:~$ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1460
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 605  bytes 13669483 (13.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 278  bytes 28410 (27.7 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 54  bytes 8225 (8.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 54  bytes 8225 (8.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

analyst@b914c359294e:~$ █
```

The Ethernet network interface is identified by the entry with the `eth` prefix.

1.2. Identify the interface options available for packet capture:

Command: sudo tcpdump -D



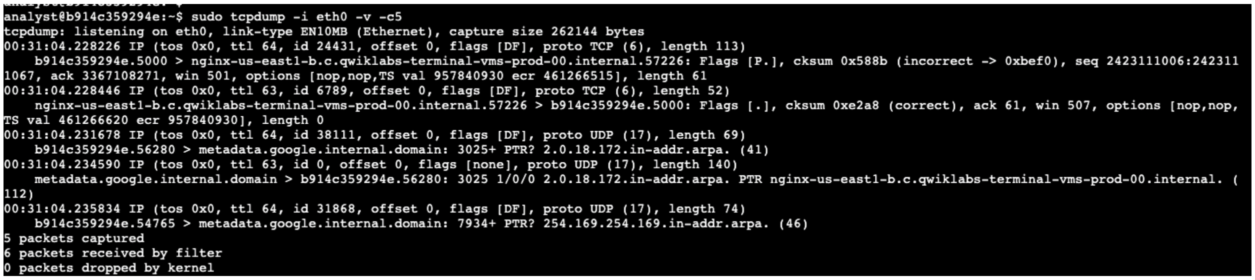
```
analyst@b914c359294e:~$
analyst@b914c359294e:~$ sudo tcpdump -D
1.eth0 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.nflog (Linux netfilter log (NFLOG) interface)
5.nfqueue (Linux netfilter queue (NFQUEUE) interface)
analyst@b914c359294e:~$ 
```

Task 2. Inspect the network traffic of a network interface with tcpdump

2.1. Filter live network packet traffic on an interface.

Command: sudo tcpdump -i eth0 -v -c5

- -i eth0: Capture data specifically from the eth0 interface.
- -v: Display detailed packet data.
- -c5: Capture 5 packets of data.



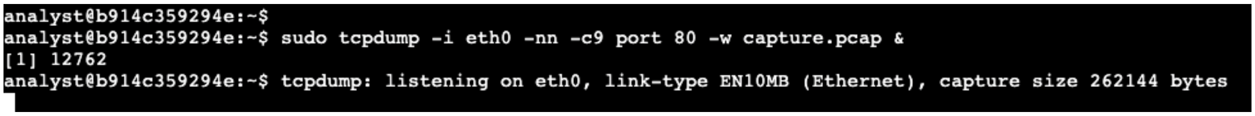
```
analyst@b914c359294e:~$ sudo tcpdump -i eth0 -v -c5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
00:31:04.228226 IP (tos 0x0, ttl 64, id 24431, offset 0, flags [DF], proto TCP (6), length 113)
    b914c359294e.5000 > nginx-us-east1-b.c.qwiklabs-terminal-vms-prod-00.internal.57226: Flags [P.], cksum 0x588b (incorrect -> 0xbef0), seq 2423111006:242311
1067, ack 3367108271, win 501, options [nop,nop,TS val 957840930 ecr 461266515], length 61
00:31:04.228446 IP (tos 0x0, ttl 63, id 6789, offset 0, flags [DF], proto TCP (6), length 52)
    nginx-us-east1-b.c.qwiklabs-terminal-vms-prod-00.internal.57226 > b914c359294e.5000: Flags [.], cksum 0xe2a8 (correct), ack 61, win 507, options [nop,nop,
TS val 461266620 ecr 957840930], length 0
00:31:04.231678 IP (tos 0x0, ttl 64, id 38111, offset 0, flags [DF], proto UDP (17), length 69)
    b914c359294e.56280 > metadata.google.internal.domain: 3025+ PTR? 2.0.18.172.in-addr.arpa. (41)
00:31:04.234590 IP (tos 0x0, ttl 63, id 0, offset 0, flags [none], proto UDP (17), length 140)
    metadata.google.internal.domain > b914c359294e.56280: 3025 1/0/0 2.0.18.172.in-addr.arpa. PTR nginx-us-east1-b.c.qwiklabs-terminal-vms-prod-00.internal. (
112)
00:31:04.235834 IP (tos 0x0, ttl 64, id 31868, offset 0, flags [DF], proto UDP (17), length 74)
    b914c359294e.54765 > metadata.google.internal.domain: 7934+ PTR? 254.169.254.169.in-addr.arpa. (46)
5 packets captured
6 packets received by filter
0 packets dropped by kernel
```

Task 3. Capture network traffic with tcpdump

3.1. Use `tcpdump` to save the captured network data to a packet capture file.

Command: sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &

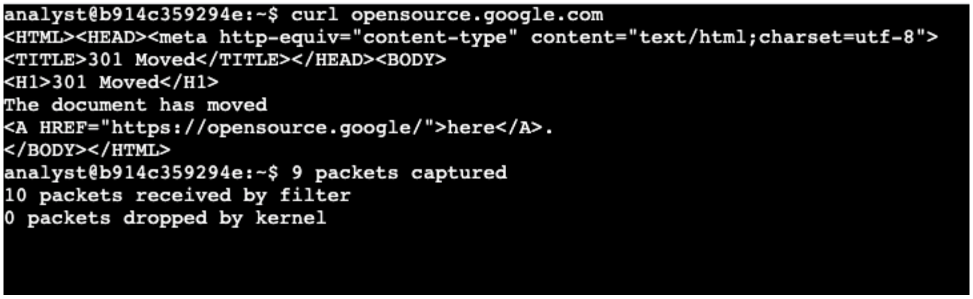
- -i eth0: Capture data from the eth0 interface.
- -nn: Do not attempt to resolve IP addresses or ports to names.This is best practice from a security perspective, as the lookup data may not be valid. It also prevents malicious actors from being alerted to an investigation.
- -c9: Capture 9 packets of data and then exit.
- port 80: Filter only port 80 traffic. This is the default HTTP port.
- -w capture.pcap: Save the captured data to the named file.
- &: This is an instruction to the Bash shell to run the command in the background.



```
analyst@b914c359294e:~$
analyst@b914c359294e:~$ sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &
[1] 12762
analyst@b914c359294e:~$ tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

3.2. Use `curl` to generate some HTTP (port 80) traffic:

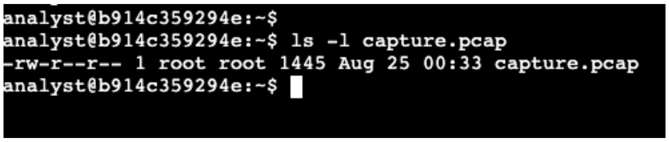
Command: curl opensource.google.com



```
analyst@b914c359294e:~$ curl opensource.google.com
<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>
analyst@b914c359294e:~$ 9 packets captured
10 packets received by filter
0 packets dropped by kernel
```

3.3. Verify that packet data has been captured:

Command: ls -l capture.pcap



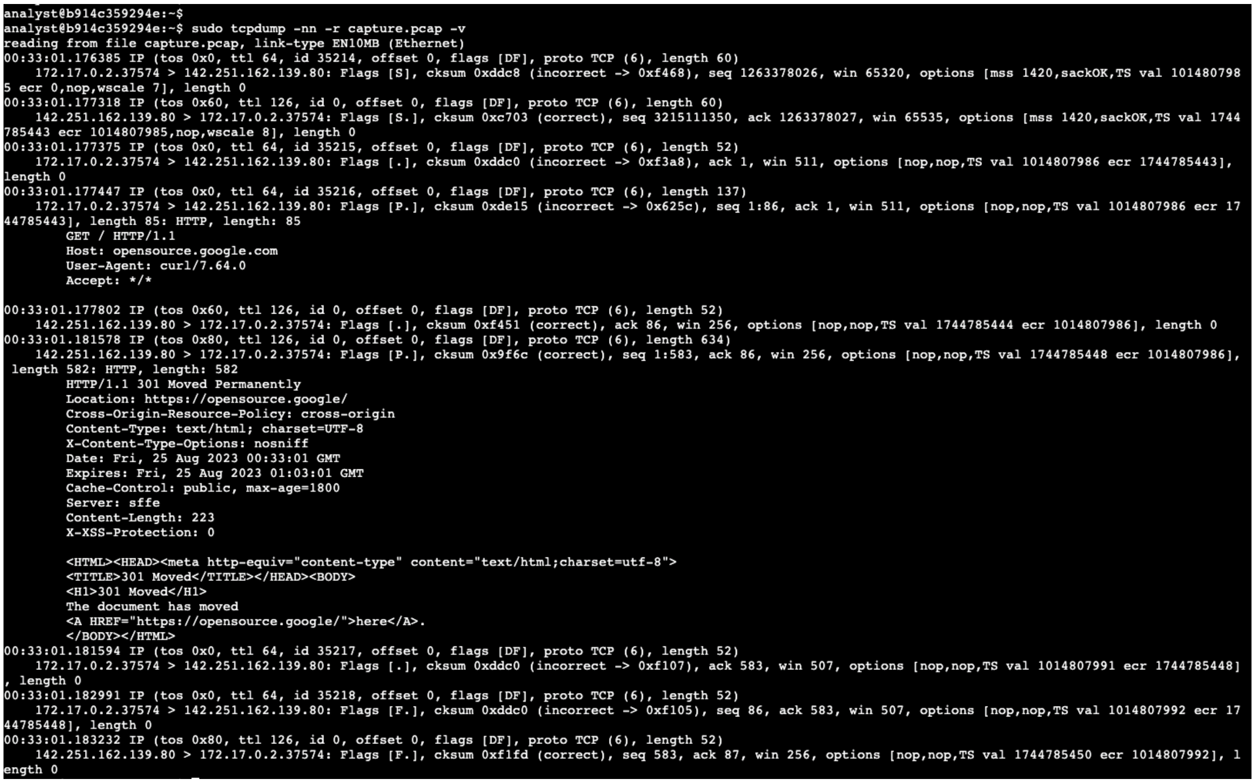
```
analyst@b914c359294e:~$
analyst@b914c359294e:~$ ls -l capture.pcap
-rw-r--r-- 1 root root 1445 Aug 25 00:33 capture.pcap
analyst@b914c359294e:~$ █
```

Task 4. Filter the captured packet data

4.1. Filter the packet header data from the `capture.pcap` capture file:

Command: sudo tcpdump -nn -r capture.pcap -v

- `-nn`: Disable port and protocol name lookup.
- `-r`: Read capture data from the named file.
- `-v`: Display detailed packet data.

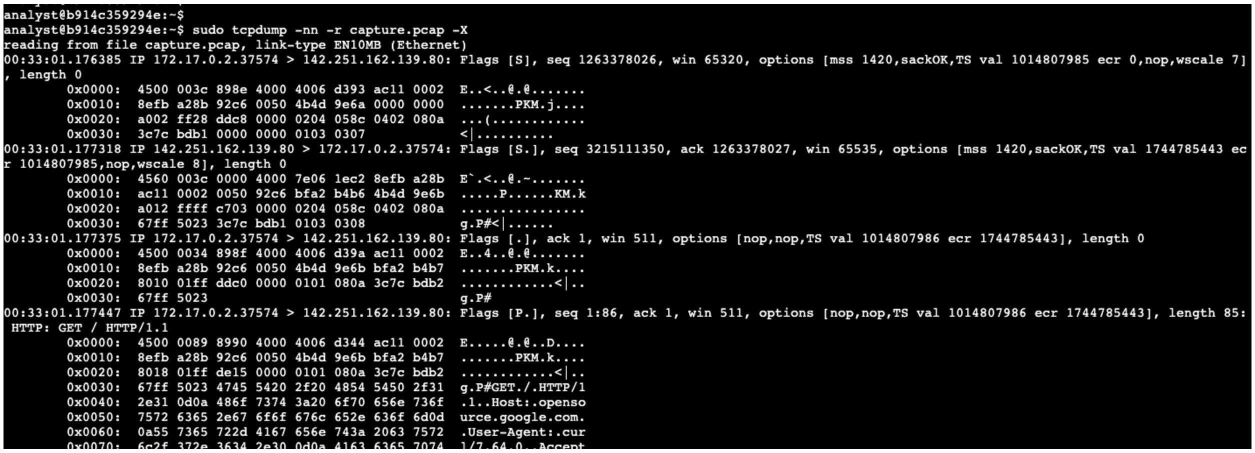


```
analyst@b914c359294e:~$
analyst@b914c359294e:~$ sudo tcpdump -nn -r capture.pcap -v
reading from file capture.pcap, link-type EN10MB (Ethernet)
00:33:01.176385 IP (tos 0x0, ttl 64, id 35214, offset 0, flags [DF], proto TCP (6), length 60)
    172.17.0.2.37574 > 142.251.162.139.80: Flags [S], cksum 0xddc8 (incorrect -> 0xf468), seq 1263378026, win 65320, options [mss 1420,sackOK,TS val 101480798
5 ecr 0,nop,wscale 7], length 0
00:33:01.177318 IP (tos 0x60, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    142.251.162.139.80 > 172.17.0.2.37574: Flags [S.], cksum 0xc703 (correct), seq 3215111350, ack 1263378027, win 65535, options [mss 1420,sackOK,TS val 1744
785443 ecr 1014807985,nop,wscale 8], length 0
00:33:01.177375 IP (tos 0x0, ttl 64, id 35215, offset 0, flags [DF], proto TCP (6), length 52)
    172.17.0.2.37574 > 142.251.162.139.80: Flags [.], cksum 0xddc0 (incorrect -> 0xf3a8), ack 1, win 511, options [nop,nop,TS val 1014807986 ecr 1744785443],
length 0
00:33:01.177447 IP (tos 0x0, ttl 64, id 35216, offset 0, flags [DF], proto TCP (6), length 137)
    172.17.0.2.37574 > 142.251.162.139.80: Flags [P.], cksum 0xde15 (incorrect -> 0x625c), seq 1:86, ack 1, win 511, options [nop,nop,TS val 1014807986 ecr 17
44785443], length 85: HTTP, length: 85
        GET / HTTP/1.1
        Host: opensource.google.com
        User-Agent: curl/7.64.0
        Accept: */*

00:33:01.177802 IP (tos 0x60, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    142.251.162.139.80 > 172.17.0.2.37574: Flags [.], cksum 0xf451 (correct), ack 86, win 256, options [nop,nop,TS val 1744785444 ecr 1014807986], length 0
00:33:01.181578 IP (tos 0x80, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 634)
    142.251.162.139.80 > 172.17.0.2.37574: Flags [P.], cksum 0x9f6c (correct), seq 1:583, ack 86, win 256, options [nop,nop,TS val 1744785448 ecr 1014807986],
 length 582: HTTP, length: 582
        HTTP/1.1 301 Moved Permanently
        Location: https://opensource.google/
        Cross-Origin-Resource-Policy: cross-origin
        Content-Type: text/html; charset=UTF-8
        X-Content-Type-Options: nosniff
        Date: Fri, 25 Aug 2023 00:33:01 GMT
        Expires: Fri, 25 Aug 2023 01:03:01 GMT
        Cache-Control: public, max-age=1800
        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>
00:33:01.181594 IP (tos 0x0, ttl 64, id 35217, offset 0, flags [DF], proto TCP (6), length 52)
    172.17.0.2.37574 > 142.251.162.139.80: Flags [.], cksum 0xddc0 (incorrect -> 0xf107), ack 583, win 507, options [nop,nop,TS val 1014807991 ecr 1744785448]
, length 0
00:33:01.182991 IP (tos 0x0, ttl 64, id 35218, offset 0, flags [DF], proto TCP (6), length 52)
    172.17.0.2.37574 > 142.251.162.139.80: Flags [F.], cksum 0xddc0 (incorrect -> 0xf105), seq 86, ack 583, win 507, options [nop,nop,TS val 1014807992 ecr 17
44785448], length 0
00:33:01.183232 IP (tos 0x80, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    142.251.162.139.80 > 172.17.0.2.37574: Flags [F.], cksum 0xf1fd (correct), seq 583, ack 87, win 256, options [nop,nop,TS val 1744785450 ecr 1014807992], l
ength 0
```

4.2. Filter the extended packet data from the `capture.pcap` capture file:

Command: sudo tcpdump -nn -r capture.pcap -X

- -nn: Disable port and protocol name lookup.
- -r: Read capture data from the named file.
- -X: Display the hexadecimal and ASCII output format packet data. Security analysts can analyze hexadecimal and ASCII output to detect patterns or anomalies during malware analysis or forensic analysis.



```
analyst@b914c359294e:~$
analyst@b914c359294e:~$ sudo tcpdump -nn -r capture.pcap -X
reading from file capture.pcap, link-type EN10MB (Ethernet)
00:33:01.176385 IP 172.17.0.2.37574 > 142.251.162.139.80: Flags [S], seq 1263378026, win 65320, options [mss 1420,sackOK,TS val 1014807985 ecr 0,nop,wscale 7]
, length 0
        0x0000:  4500 003c 898e 4000 4006 d393 ac11 0002  E..<..@.@.......
        0x0010:  8efb a28b 92c6 0050 4b4d 9e6a 0000 0000  .......PKM.j....
        0x0020:  a002 ff28 ddc8 0000 0204 058c 0402 080a  ...(............
        0x0030:  3c7c bdb1 0000 0000 0103 0307            <|..........
00:33:01.177318 IP 142.251.162.139.80 > 172.17.0.2.37574: Flags [S.], seq 3215111350, ack 1263378027, win 65535, options [mss 1420,sackOK,TS val 1744785443 ec
r 1014807985,nop,wscale 8], length 0
        0x0000:  4560 003c 0000 4000 7e06 1ec2 8efb a28b  E`.<..@.~.......
        0x0010:  ac11 0002 0050 92c6 bfa2 b4b6 4b4d 9e6b  .....P......KM.k
        0x0020:  a012 ffff c703 0000 0204 058c 0402 080a  ................
        0x0030:  67ff 5023 3c7c bdb1 0103 0308            g.P#<|......
00:33:01.177375 IP 172.17.0.2.37574 > 142.251.162.139.80: Flags [.], ack 1, win 511, options [nop,nop,TS val 1014807986 ecr 1744785443], length 0
        0x0000:  4500 0034 898f 4000 4006 d39a ac11 0002  E..4..@.@.......
        0x0010:  8efb a28b 92c6 0050 4b4d 9e6b bfa2 b4b7  .......PKM.k....
        0x0020:  8010 01ff ddc0 0000 0101 080a 3c7c bdb2  ............<|..
        0x0030:  67ff 5023                                g.P#
00:33:01.177447 IP 172.17.0.2.37574 > 142.251.162.139.80: Flags [P.], seq 1:86, ack 1, win 511, options [nop,nop,TS val 1014807986 ecr 1744785443], length 85:
 HTTP: GET / HTTP/1.1
        0x0000:  4500 0089 8990 4000 4006 d344 ac11 0002  E.....@.@..D....
        0x0010:  8efb a28b 92c6 0050 4b4d 9e6b bfa2 b4b7  .......PKM.k....
        0x0020:  8018 01ff de15 0000 0101 080a 3c7c bdb2  ............<|..
        0x0030:  67ff 5023 4745 5420 2f20 4854 5450 2f31  g.P#GET./.HTTP/1
        0x0040:  2e31 0d0a 486f 7374 3a20 6f70 656e 736f  .1..Host:.openso
        0x0050:  7572 6365 2e67 6f6f 676c 652e 636f 6d0d  urce.google.com.
        0x0060:  0a55 7365 722d 4167 656e 743a 2063 7572  .User-Agent:.cur
        0x0070:  6c2f 372e 3634 2e30 0d0a 4163 6365 7074  l/7.64.0..Accept
```