

# 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.

Your Linux user's home directory contains a sample packet capture file that you will use at the end of the lab to answer a few questions about the network traffic that it contains.

## Task 1. Identify network interfaces

In this task, you must identify the network interfaces that can be used to capture network packet data.

1. Use `ifconfig` to identify the interfaces that are available:

← Activity: Capture your first packet

```
analyst@52758c0ae67d:~$ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1460
        inet 172.18.0.2  netmask 255.255.0.0  broadcast 172.18.255.255
                ether 02:42:ac:12:00:02  txqueuelen 0  (Ethernet)
                  RX packets 3644  bytes 13934719 (13.2 MiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 3405  bytes 237101 (231.5 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 58  bytes 8433 (8.2 KiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 58  bytes 8433 (8.2 KiB)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

analyst@52758c0ae67d:~$
```

Next, Identify the interface options available for packet capture:

```
analyst@52758c0ae67d:~$ 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@52758c0ae67d:~$
```

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

- Use `sudo tcpdump -i eth0 -v -c5` to filter live network packet data:

```
analyst@52758c0ae67d:~$ sudo tcpdump -i eth0 -v -c5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:41:41.464427 IP (tos 0x0, ttl 64, id 31657, offset 0, flags [DF], proto TCP (6), length 115)
    52758c0ae67d.5000 > nginx-us-central1-b.c.qwiklabs-terminal-vms-prod-00.internal.52964: Flags [P.], cksum 0x588d (incorrect -> 0xb7b5), seq 2596934417:2596934480, ack 2009877808, win 492, options [nop,nop,TS val 3548894881 ecr 2323907120], length 63
18:41:41.464622 IP (tos 0x0, ttl 63, id 57800, offset 0, flags [DF], proto TCP (6), length 52)
    nginx-us-central1-b.c.qwiklabs-terminal-vms-prod-00.internal.52964 > 52758c0ae67d.5000: Flags [.], cksum 0x8e83 (correct), ack 63, win 507, options [nop,nop,TS val 2323907557 ecr 3548894881], length 0
18:41:41.475119 IP (tos 0x0, ttl 64, id 31658, offset 0, flags [DF], proto TCP (6), length 146)
    52758c0ae67d.5000 > nginx-us-central1-b.c.qwiklabs-terminal-vms-prod-00.internal.52964: Flags [P.], cksum 0x58ac (incorrect -> 0x4f99), seq 63:157, ack 1, win 492, options [nop,nop,TS val 3548894891 ecr 2323907557], length 94
18:41:41.475496 IP (tos 0x0, ttl 63, id 57801, offset 0, flags [DF], proto TCP (6), length 52)
    nginx-us-central1-b.c.qwiklabs-terminal-vms-prod-00.internal.52964 > 52758c0ae67d.5000: Flags [.], cksum 0x8e11 (correct), ack 157, win 507, options [nop,nop,TS val 2323907567 ecr 3548894891], length 0
18:41:41.490331 IP (tos 0x0, ttl 64, id 10697, offset 0, flags [DF], proto UDP (17), length 69)
    52758c0ae67d.48552 > metadata.google.internal.domain: 44489+ PTR? 2.0.17.17
2.in-addr.arpa. (41)
5 packets captured
10 packets received by filter
0 packets dropped by kernel
analyst@52758c0ae67d:~$
```

`-i eth0`: Capture data specifically from the eth0 interface.

`-v`: Display detailed packet data.

`-c5`: Capture 5 packets of data.

## 3. Capture Network traffic:

- Capture packet data into a file named called `capture.pcap`: `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.

```
nop,TS val 2323907567 ecr 3548894891], length 0
18:41:41.490331 IP (tos 0x0, ttl 64, id 10697, offset 0, flags [DF], proto UDP
(17), length 69)
    52758c0ae67d.48552 > metadata.google.internal.domain: 44489+ PTR? 2.0.17.17
2.in-addr.arpa. (41)
5 packets captured
10 packets received by filter
0 packets dropped by kernel
analyst@52758c0ae67d:~$ sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &
[1] 12770
analyst@52758c0ae67d:~$ tcpdump: listening on eth0, link-type EN10MB (Ethernet)
, capture size 262144 bytes

analyst@52758c0ae67d:~$
```

- Use curl to generate some HTTP (port 80) traffic: curl opensource.google.com.

```
, capture size 262144 bytes

analyst@52758c0ae67d:~$ 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@52758c0ae67d:~$ 9 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Verify the packet data has been captured: ls -l capture.pcap

```
</BODY></HTML>
analyst@52758c0ae67d:~$ 9 packets captured
10 packets received by filter
0 packets dropped by kernel
ls -l capture.pcap
-rw-r--r-- 1 root root 1401 Aug 21 18:44 capture.pcap
[1]+  Done                      sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.p
cap
analyst@52758c0ae67d:~$
```

#### 4. Filter the captured packet data.

- Filter the packet header data from the capture.pcap capture file: sudo tcpdump -nn -r capture.pcap -v.

```
cap
analyst@52758c0ae67d:~$ sudo tcpdump -nn -r capture.pcap -v
reading from file capture.pcap, link-type EN10MB (Ethernet)
18:44:36.722273 IP (tos 0x0, ttl 64, id 39252, offset 0, flags [DF], proto TCP
(6), length 60)
    172.18.0.2.35762 > 209.85.145.102.80: Flags [S], cksum 0x0eff (incorrect ->
0x8fac), seq 1584878191, win 32660, options [mss 1420,sackOK,TS val 2231265328
ecr 0,nop,wscale 6], length 0
18:44:36.723461 IP (tos 0x60, ttl 126, id 0, offset 0, flags [DF], proto TCP (6
), length 60)
    209.85.145.102.80 > 172.18.0.2.35762: Flags [S.], cksum 0x65e9 (correct), s
eq 1496339935, ack 1584878192, win 65535, options [mss 1420,sackOK,TS val 29376
04892 ecr 2231265328,nop,wscale 8], length 0
18:44:36.723482 IP (tos 0x0, ttl 64, id 39253, offset 0, flags [DF], proto TCP
(6), length 52)
    172.18.0.2.35762 > 209.85.145.102.80: Flags [.], cksum 0x0ef7 (incorrect ->
0x928e), ack 1, win 511, options [nop,nop,TS val 2231265329 ecr 2937604892], l
ength 0
18:44:36.723568 IP (tos 0x0, ttl 64, id 39254, offset 0, flags [DF], proto TCP
(6), length 137)
    172.18.0.2.35762 > 209.85.145.102.80: Flags [P.], cksum 0x0f4c (incorrect ->
0x0142), seq 1:86, ack 1, win 511, options [nop,nop,TS val 2231265329 ecr 293
7604892], length 85: HTTP, length: 85
        GET / HTTP/1.1
        Host: opensource.google.com
        User-Agent: curl/7.64.0
        Accept: /*

18:44:36.723994 IP (tos 0x60, ttl 126, id 0, offset 0, flags [DF], proto TCP (6
), length 52)
    209.85.145.102.80 > 172.18.0.2.35762: Flags [.], cksum 0x9337 (correct), ac
k 86, win 256, options [nop,nop,TS val 2937604893 ecr 2231265329], length 0
18:44:36.726160 IP (tos 0x80, ttl 126, id 0, offset 0, flags [DF], proto TCP (6
), length 590)
    209.85.145.102.80 > 172.18.0.2.35762: Flags [P.], cksum 0x5e53 (correct), s
eq 1:539, ack 86, win 256, options [nop,nop,TS val 2937604895 ecr 2231265329],
length 538: HTTP, length: 538
        HTTP/1.1 301 Moved Permanently
        Location: https://opensource.google/
        Content-Type: text/html; charset=UTF-8
        X-Content-Type-Options: nosniff
```

**-nn:** Disable port and protocol name lookup.

**-r:** Read capture data from the named file.

**-v:** Display detailed packet data.

- Filter the extended packet data from the capture.pcap capture file: sudo tcpdump -nn -r capture.pcap -X.

```

nghost 0
analyst@52758c0ae67d:~$ sudo tcpdump -nn -r capture.pcap -X
reading from file capture.pcap, link-type EN10MB (Ethernet)
18:44:36.722273 IP 172.18.0.2.35762 > 209.85.145.102.80: Flags [S], seq 1584878
191, win 32660, options [mss 1420,sackOK,TS val 2231265328 ecr 0,nop,wscale 6],
length 0
    0x0000: 4500 003c 9954 4000 4006 9297 ac12 0002 E..<.T@.e.....
    0x0010: d155 9166 8bb2 0050 5e77 526f 0000 0000 .U.f...P^wRo...
    0x0020: a002 7f94 0eff 0000 0204 058c 0402 080a .....
    0x0030: 84fe 6830 0000 0000 0103 0306 ..h0.....
18:44:36.723461 IP 209.85.145.102.80 > 172.18.0.2.35762: Flags [S.], seq 149633
9935, ack 1584878192, win 65535, options [mss 1420,sackOK,TS val 2937604892 ecr
2231265328,nop,wscale 8], length 0
    0x0000: 4560 003c 0000 4000 7e06 ed8b d155 9166 E^.<..@.-....U.f
    0x0010: ac12 0002 0050 8bb2 5930 55df 5e77 5270 .....P..YOU.^wRp
    0x0020: a012 ffff 65e9 0000 0204 058c 0402 080a ....e.....
    0x0030: af18 4b1c 84fe 6830 0103 0308 ..K...h0....
18:44:36.723482 IP 172.18.0.2.35762 > 209.85.145.102.80: Flags [., ack 1, win
511, options [nop,nop,TS val 2231265329 ecr 2937604892], length 0
    0x0000: 4500 0034 9955 4000 4006 929e ac12 0002 E..4.U@.e.....
    0x0010: d155 9166 8bb2 0050 5e77 5270 5930 55e0 .U.f...P^wRpYOU.
    0x0020: 8010 01ff 0eff 0000 0101 080a 84fe 6831 .....h1
    0x0030: af18 4b1c ..K.
18:44:36.723568 IP 172.18.0.2.35762 > 209.85.145.102.80: Flags [P.], seq 1:86,
ack 1, win 511, options [nop,nop,TS val 2231265329 ecr 2937604892], length 85:
HTTP: GET / HTTP/1.1
    0x0000: 4500 0089 9956 4000 4006 9248 ac12 0002 E....V@.e..H....
    0x0010: d155 9166 8bb2 0050 5e77 5270 5930 55e0 .U.f...P^wRpYOU.
    0x0020: 8018 01ff 0f4c 0000 0101 080a 84fe 6831 .....L.....h1
    0x0030: af18 4b1c 4745 5420 2f20 4854 5450 2f31 ..K.GET./HTTP/1
    0x0040: 2e31 0d0a 486f 7374 3a20 6f70 656e 736f .1..Host:.openso

```

**-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.