Capture Your First Packet

Skills Acquired:

- Identify network interfaces
- Use the tcpdump command to capture network data for inspection
- Interpret the information that tcpdump outputs regarding a packet
- Save and load packet data for later analysis

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:

```
sudo ifconfig
```

This command returns output similar to the following:

```
eth0: flags=4163  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 784  bytes9379957 (8.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 683  bytes 56880 (55.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0 collisions 0
lo: flags=73  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 400  bytes 42122 (041.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 400  bytes 42122 (041.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0 collisions 0
```

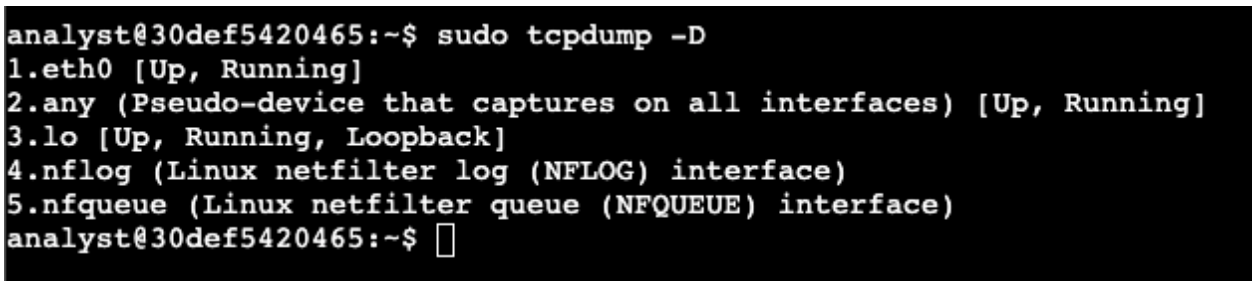
The Ethernet network interface is identified by the entry with the eth prefix.

So, in this lab, you'll use eth0 as the interface that you will capture network packet data from in the following tasks.

2. Use tcpdump to identify the interface options available for packet capture:

sudo tcpdump -D

This command will also allow you to identify which network interfaces are available. This may be useful on systems that do not include the ifconfig command.



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

In this task, you must use tcpdump to filter live network packet traffic on an interface.

- Filter live network packet data from the eth0 interface with tcpdump:

```
sudo tcpdump -i eth0 -v -c5
```

This command will run tcpdump with the following options:

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

```
analyst@30def5420465:~$ sudo tcpdump -i eth0 -v -c5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:52:38.003944 IP (tos 0x0, ttl 64, id 6005, offset 0, flags [DF], proto TCP (6), length
119)
    30def5420465.5000 > nginx-us-east1-b.c.qwiklabs-terminal-vms-prod-00.internal.35176: F
lags [P.], cksum 0x5891 (incorrect -> 0x8278), seq 1656418599:1656418666, ack 303299831, w
in 501, options [nop,nop,TS val 3981757777 ecr 3905179333], length 67
11:52:38.004179 IP (tos 0x0, ttl 63, id 9131, offset 0, flags [DF], proto TCP (6), length
52)
    nginx-us-east1-b.c.qwiklabs-terminal-vms-prod-00.internal.35176 > 30def5420465.5000: F
lags [.], cksum 0x29ce (correct), ack 67, win 507, options [nop,nop,TS val 3905179456 ecr
3981757777], length 0
11:52:38.014538 IP (tos 0x0, ttl 64, id 6006, offset 0, flags [DF], proto TCP (6), length
145)
    30def5420465.5000 > nginx-us-east1-b.c.qwiklabs-terminal-vms-prod-00.internal.35176: F
lags [P.], cksum 0x58ab (incorrect -> 0x12bd), seq 67:160, ack 1, win 501, options [nop,no
p,TS val 3981757787 ecr 3905179456], length 93
11:52:38.014900 IP (tos 0x0, ttl 63, id 9132, offset 0, flags [DF], proto TCP (6), length
52)
    nginx-us-east1-b.c.qwiklabs-terminal-vms-prod-00.internal.35176 > 30def5420465.5000: F
lags [.], cksum 0x295c (correct), ack 160, win 507, options [nop,nop,TS val 3905179467 ecr
 3981757787], length 0
11:52:38.029469 IP (tos 0x0, ttl 64, id 20776, offset 0, flags [DF], proto UDP (17), lengt
h 69)
    30def5420465.42418 > metadata.google.internal.domain: 38959+ PTR? 2.0.18.172.in-addr.a
rpa. (41)
5 packets captured
10 packets received by filter
0 packets dropped by kernel
analyst@30def5420465:~$ sudo tcpdump -i eth0 -v -c5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:58:13.907309 IP (tos 0x0, ttl 64, id 6069, offset 0, flags [DF], proto TCP (6), length
59)
```

Task 3. Capture network traffic with tcpdump

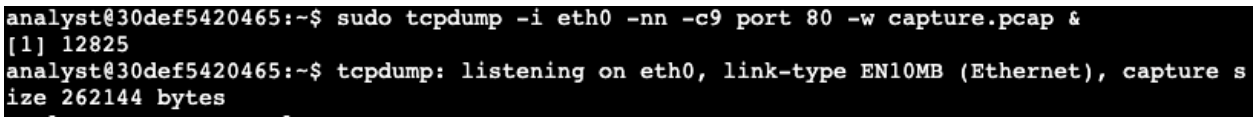
In this task, you will use tcpdump to save the captured network data to a packet capture file.

In the previous command, you used tcpdump to stream all network traffic. Here, you will use a filter and other tcpdump configuration options to save a small sample that contains only web (TCP port 80) network packet data.

1. Capture packet data into a file called capture.pcap:

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

You must press the **ENTER** key to get your command prompt back after running this command.



This command will run tcpdump in the background with the following options:

- -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.

This command runs in the background, but some output text will appear in your terminal. The text will not affect the commands when you follow the steps for the rest of the lab.

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

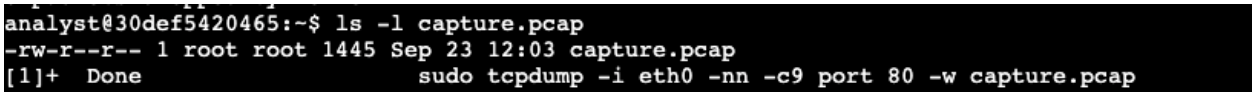
curl opensource.google.com

```
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@30def5420465:~$ 9 packets captured
10 packets received by filter
0 packets dropped by kernel
```

When the curl command is used like this to open a website, it generates some HTTP (TCP port 80) traffic that can be captured.

3. Verify that packet data has been captured:

ls -l capture.pcap



Task 4. Filter the captured packet data

In this task, use tcpdump to filter data from the packet capture file you saved previously.

1. Use the tcpdump command to filter the packet header data from the capture.pcap capture file:

```
sudo tcpdump -nn -r capture.pcap -v
```

This command will run tcpdump with the following options:

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

You must specify the -nn switch again here, as you want to make sure tcpdump does not perform name lookups of either IP addresses or ports, since this can alert threat actors.

```
analyst@30def5420465:~$ sudo tcpdump -nn -r capture.pcap -v
reading from file capture.pcap, link-type EN10MB (Ethernet)
12:03:14.803988 IP (tos 0x0, ttl 64, id 47113, offset 0, flags [DF], proto TCP (6), length
 60)
    172.17.0.2.48690 > 172.217.203.102.80: Flags [S], cksum 0x2482 (incorrect -> 0xa590),
seq 3362487201, win 65320, options [mss 1420,sackOK,TS val 1999635803 ecr 0,nop,wscale 7],
 length 0
12:03:14.805209 IP (tos 0x60, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 6
0)
    172.217.203.102.80 > 172.17.0.2.48690: Flags [S.], cksum 0x6253 (correct), seq 2681372
490, ack 3362487202, win 65535, options [mss 1420,sackOK,TS val 1081467585 ecr 1999635803,
nop,wscale 8], length 0
12:03:14.805261 IP (tos 0x0, ttl 64, id 47114, offset 0, flags [DF], proto TCP (6), length
 52)
    172.17.0.2.48690 > 172.217.203.102.80: Flags [.], cksum 0x247a (incorrect -> 0x8ef8),
ack 1, win 511, options [nop,nop,TS val 1999635804 ecr 1081467585], length 0
```

As in the previous example, you can see the IP packet information along with information about the data that the packet contains.

2. Use the tcpdump command to filter the extended packet data from the capture.pcap capture file:

```
sudo tcpdump -nn -r capture.pcap -X
```

This command will run tcpdump with the following options:

- -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@30def5420465:~$ sudo tcpdump -nn -r capture.pcap -X
reading from file capture.pcap, link-type EN10MB (Ethernet)
12:03:14.803988 IP 172.17.0.2.48690 > 172.217.203.102.80: Flags [S], seq 3362487201, win 6
5320, options [mss 1420,sackOK,TS val 1999635803 ecr 0,nop,wscale 7], length 0
        0x0000:  4500 003c b809 4000 4006 5e5f ac11 0002  E..<..@.@.^_....
        0x0010:  acd9 cb66 be32 0050 c86b 7ba1 0000 0000  ...f.2.P.k{.....
        0x0020:  a002 ff28 2482 0000 0204 058c 0402 080a  ...($...........
        0x0030:  7730 055b 0000 0000 0103 0307            w0.[........
12:03:14.805209 IP 172.217.203.102.80 > 172.17.0.2.48690: Flags [S.], seq 2681372490, ack
3362487202, win 65535, options [mss 1420,sackOK,TS val 1081467585 ecr 1999635803,nop,wscal
e 8], length 0
        0x0000:  4560 003c 0000 4000 7e06 d808 acd9 cb66  E`.<..@.~......f
        0x0010:  ac11 0002 0050 be32 9fd2 7f4a c86b 7ba2  .....P.2...J.k{.
        0x0020:  a012 ffff 6253 0000 0204 058c 0402 080a  ....bS..........
        0x0030:  4075 e2c1 7730 055b 0103 0308            @u..w0.[....
12:03:14.805261 IP 172.17.0.2.48690 > 172.217.203.102.80: Flags [.], ack 1, win 511, optio
ns [nop,nop,TS val 1999635804 ecr 1081467585], length 0
        0x0000:  4500 0034 b80a 4000 4006 5e66 ac11 0002  E..4..@.@.^f....
        0x0010:  acd9 cb66 be32 0050 c86b 7ba2 9fd2 7f4b  ...f.2.P.k{....K
        0x0020:  8010 01ff 247a 0000 0101 080a 7730 055c  ....$z......w0.\
        0x0030:  4075 e2c1                                @u..
12:03:14.805317 IP 172.17.0.2.48690 > 172.217.203.102.80: Flags [P.], seq 1:86, ack 1, win
 511, options [nop,nop,TS val 1999635804 ecr 1081467585], length 85: HTTP: GET / HTTP/1.1
        0x0000:  4500 0089 b80b 4000 4006 5e10 ac11 0002  E.....@.@.^.....
        0x0010:  acd9 cb66 be32 0050 c86b 7ba2 9fd2 7f4b  ...f.2.P.k{....K
        0x0020:  8018 01ff 24cf 0000 0101 080a 7730 055c  ....$.......w0.\
        0x0030:  4075 e2c1 4745 5420 2f20 4854 5450 2f31  @u..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
        0x0080:  3a20 2a2f 2a0d 0a0d 0a                   :.*/*....
12:03:14.805516 IP 172.217.203.102.80 > 172.17.0.2.48690: Flags [.], ack 86, win 256, opti
ons [nop,nop,TS val 1081467586 ecr 1999635804], length 0
        0x0000:  4560 0034 0000 4000 7e06 d810 acd9 cb66  E`.4..@.~......f
```

Test your understanding

To test your ability to capture and view network data, answer the questions.

- **What command would you use to capture 3 packets on any interface with the verbose option?**

  sudo tcpdump -c3 -i any -v

- **What does the -i option indicate?**

  The network interface to monitor

- **What type of information does the -v option include?**

  Verbose information

- **What tcpdump command can you use to identify the interfaces that are available to perform a packet capture on?**

  sudo tcpdump -D