TCPDump

Project Description

I am a network analyst who uses tcpdump to capture and analyze network traffic from a Linux virtual machine.

Identify Network Interfaces

```
analyst@4b8a8483d584:~$ ifconfig
-bash: ifconfig: command not found
analyst@4b8a8483d584:~$ 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 841  bytes 13718648 (13.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 484  bytes 45857 (44.7 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 135  bytes 15957 (15.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 135  bytes 15957 (15.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

I use sudo to obtain superuser permissions to access the ifconfig program to display network adapter configuration information.

The network interface eth0 has the IP address 172.17.0.2.

```
analyst@4b8a8483d584:~$ 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)
```

I run tcpdump with the -D flag to list interfaces on which tcpdump can capture packets. This command is helpful on computers where ifconfig is unavailable.

Inspect network traffic of a network interface with tcpdump

```
analyst@4b8a8483d584:~$ sudo tcpdump -i eth0 -v -c5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:40:12.630078 IP (tos 0x0, ttl 64, id 20035, offset 0, flags [DF], proto TCP (6), length 104)
    4b8a8483d584.5000 > nginx-us-west1-c.c.qwiklabs-terminal-vms-prod-00.internal.56440: Flags [P.], cksum
 0x5886 (incorrect -> 0xcadf), seq 857782288:857782340, ack 1450882533, win 501, options [nop,nop,TS val 7
90202555 ecr 3150368259], length 52
13:40:12.630297 IP (tos 0x0, ttl 63, id 6237, offset 0, flags [DF], proto TCP (6), length 52)
    nginx-us-west1-c.c.qwiklabs-terminal-vms-prod-00.internal.56440 > 4b8a8483d584.5000: Flags [.], cksum
0xf28b (correct), ack 52, win 501, options [nop,nop,TS val 3150368431 ecr 790202555], length 0
13:40:12.630314 IP (tos 0x0, ttl 64, id 16715, offset 0, flags [DF], proto TCP (6), length 104)
    4b8a8483d584.5000 > nginx-us-west1-c.c.qwiklabs-terminal-vms-prod-00.internal.56476: Flags [P.], cksum
 0x5886 (incorrect -> 0xfd88), seq 3582794748:3582794800, ack 40358098, win 501, options [nop,nop,TS val 7
90202555 ecr 3150368259], length 52
13:40:12.630446 IP (tos 0x0, ttl 63, id 52603, offset 0, flags [DF], proto TCP (6), length 52)
    nginx-us-west1-c.c.qwiklabs-terminal-vms-prod-00.internal.56476 > 4b8a8483d584.5000: Flags [.], cksum
0x2535 (correct), ack 52, win 501, options [nop,nop,TS val 3150368431 ecr 790202555], length 0
13:40:12.640761 IP (tos 0x0, ttl 64, id 20036, offset 0, flags [DF], proto TCP (6), length 142)
    4b8a8483d584.5000 > nginx-us-west1-c.c.qwiklabs-terminal-vms-prod-00.internal.56440: Flags [P.], cksum
 0x58ac (incorrect -> 0x2ee7), seq 52:142, ack 1, win 501, options [nop,nop,TS val 790202566 ecr 315036843
1], length 90
5 packets captured
14 packets received by filter
3 packets dropped by kernel
```

I run tcpdump with the following options:
-i eth0: capture data from eth0
-v: Be verbose and show detailed packet data
-c5: Capture 5 data packets

tcpdump says it is listening on eth0 and shows the link type and capture size in bytes.

The next line shows one of the packets. It starts with the timestamp and the protocol type (IP). The verbose flag has shown IP packet field information such as Type Of Service, Time To Live, ID, offset, flags, internal protocol type and length of the outer packet in bytes. These properties relate to the network IP packet but their details are beyond the scope of this lab.

The next line shows the two systems which are communicating with each other. The direction arrow shows the direction of traffic flow. The port is also shown (56440 in the screenshot). It is the default setting that tcpdump converts IP addresses into names. This may sometimes alert an adversary that they are being investigated and can be turned off with the -n flag [1].

The next line shows the inner TCP packet's header data. P stands for the push flag, the period shows it is an ACK flag. The packet is pushing data out. The next field is TCP checksum value, this is used to detect errors in data. It also includes the sequence and acknowledgment numbers, the window size, and length of inner TCP packet. These details are not the focus of this lab.

Capture Network Traffic with tcpdump

```
analyst@4b8a8483d584:~$ sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &
[1] 12914
analyst@4b8a8483d584:~$ tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
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@4b8a8483d584:~$ 9 packets captured
10 packets received by filter
0 packets dropped by kernel

[1]+  Done                    sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap
```

I run tcpdump in the background with these flags:

-i eth0: capture from eth0 interface
-nn: do not resolve IP addresses or port names. Lookup data may be invalid. Malicious actors are also not alerted to the investigation
-c9: capture 9 packets of data
port 80: capture only port 80 traffic (this is the port for http)
-w capture.pcap: save captured data as capture.pcap
&: run the command in the background

curl generates http traffic by fetching the page opensource.google.com

Filter captured packet data

```
pcap -v@4b8a8483d584:~$ sudo tcpdump -nn -r capture.
reading from file capture.pcap, link-type EN10MB (Ethernet)
13:58:54.170565 IP (tos 0x0, ttl 64, id 30427, offset 0, flags [DF], proto TCP (6), length 60)
    172.17.0.2.49896 > 74.125.135.139.80: Flags [S], cksum 0x7e4a (incorrect -> 0x4750), seq 3339124237, win 65320, options [
mss 1420,sackOK,TS val 3478531055 ecr 0,nop,wscale 7], length 0
13:58:54.171577 IP (tos 0x60, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    74.125.135.139.80 > 172.17.0.2.49896: Flags [S.], cksum 0x8cc3 (correct), seq 402119830, ack 3339124238, win 65535, optio
ns [mss 1420,sackOK,TS val 2465083943 ecr 3478531055,nop,wscale 8], length 0
13:58:54.171657 IP (tos 0x0, ttl 64, id 30428, offset 0, flags [DF], proto TCP (6), length 52)
    172.17.0.2.49896 > 74.125.135.139.80: Flags [.], cksum 0x7e42 (incorrect -> 0xb968), ack 1, win 511, options [nop,nop,TS
val 3478531056 ecr 2465083943], length 0
13:58:54.171753 IP (tos 0x0, ttl 64, id 30429, offset 0, flags [DF], proto TCP (6), length 137)
    172.17.0.2.49896 > 74.125.135.139.80: Flags [P.], cksum 0x7e97 (incorrect -> 0x281c), seq 1:86, ack 1, win 511, options [
nop,nop,TS val 3478531056 ecr 2465083943], length 85: HTTP, length: 85
        GET / HTTP/1.1
        Host: opensource.google.com
        User-Agent: curl/7.64.0
        Accept: */*

13:58:54.171934 IP (tos 0x60, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    74.125.135.139.80 > 172.17.0.2.49896: Flags [.], cksum 0xba12 (correct), ack 86, win 256, options [nop,nop,TS val 2465083
943 ecr 3478531056], length 0
13:58:54.176715 IP (tos 0x80, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 634)
    74.125.135.139.80 > 172.17.0.2.49896: Flags [P.], cksum 0x7cb4 (correct), seq 1:583, ack 86, win 256, options [nop,nop,TS
 val 2465083948 ecr 3478531056], 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: Sat, 25 Nov 2023 13:58:54 GMT
        Expires: Sat, 25 Nov 2023 14:28:54 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>
13:58:54.176732 IP (tos 0x0, ttl 64, id 30430, offset 0, flags [DF], proto TCP (6), length 52)
    172.17.0.2.49896 > 74.125.135.139.80: Flags [.], cksum 0x7e42 (incorrect -> 0xb6c7), ack 583, win 507, options [nop,nop,T
S val 3478531061 ecr 2465083948], length 0
13:58:54.178132 IP (tos 0x0, ttl 64, id 30431, offset 0, flags [DF], proto TCP (6), length 52)
    172.17.0.2.49896 > 74.125.135.139.80: Flags [F.], cksum 0x7e42 (incorrect -> 0xb6c5), seq 86, ack 583, win 507, options [
nop,nop,TS val 3478531062 ecr 2465083948], length 0
13:58:54.178599 IP (tos 0x80, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    74.125.135.139.80 > 172.17.0.2.49896: Flags [F.], cksum 0xb7bd (correct), seq 583, ack 87, win 256, options [nop,nop,TS v
al 2465083950 ecr 3478531062], length 0
```

I run tcpdump with these flags:

-nn: disables port/protocol name resolution
-r capture.pcap: read capture data from capture.pcap
-v: display verbose packet details

```
analyst@4b8a8483d584:~$ sudo tcpdump -nn -r capture.pcap -X
reading from file capture.pcap, link-type EN10MB (Ethernet)
13:58:54.170565 IP 172.17.0.2.49896 > 74.125.135.139.80: Flags [S], seq 3339124237, win 65320, options [mss 1420,sackOK,TS va
l 3478531055 ecr 0,nop,wscale 7], length 0
        0x0000:  4500 003c 76db 4000 4006 45c5 ac11 0002  E..<v.@.@.E.....
        0x0010:  4a7d 878b c2e8 0050 c706 fe0d 0000 0000  J}.....P........
        0x0020:  a002 ff28 7e4a 0000 0204 058c 0402 080a  ...(~J..........
        0x0030:  cf56 2bef 0000 0000 0103 0307            .V+.........
13:58:54.171577 IP 74.125.135.139.80 > 172.17.0.2.49896: Flags [S.], seq 402119830, ack 3339124238, win 65535, options [mss 1
420,sackOK,TS val 2465083943 ecr 3478531055,nop,wscale 8], length 0
        0x0000:  4560 003c 0000 4000 7e06 7e40 4a7d 878b  E`.<..@.~.~@J}..
        0x0010:  ac11 0002 0050 c2e8 17f7 dc96 c706 fe0e  .....P..........
        0x0020:  a012 ffff 8cc3 0000 0204 058c 0402 080a  ................
        0x0030:  92ee 3227 cf56 2bef 0103 0308            ..2'.V+.....
13:58:54.171657 IP 172.17.0.2.49896 > 74.125.135.139.80: Flags [.], ack 1, win 511, options [nop,nop,TS val 3478531056 ecr 24
65083943], length 0
        0x0000:  4500 0034 76dc 4000 4006 45cc ac11 0002  E..4v.@.@.E.....
        0x0010:  4a7d 878b c2e8 0050 c706 fe0e 17f7 dc97  J}.....P........
        0x0020:  8010 01ff 7e42 0000 0101 080a cf56 2bf0  ....~B.......V+.
        0x0030:  92ee 3227                                ..2'
13:58:54.171753 IP 172.17.0.2.49896 > 74.125.135.139.80: Flags [P.], seq 1:86, ack 1, win 511, options [nop,nop,TS val 347853
1056 ecr 2465083943], length 85: HTTP: GET / HTTP/1.1
        0x0000:  4500 0089 76dd 4000 4006 4576 ac11 0002  E...v.@.@.Ev....
        0x0010:  4a7d 878b c2e8 0050 c706 fe0e 17f7 dc97  J}.....P........
        0x0020:  8018 01ff 7e97 0000 0101 080a cf56 2bf0  ....~........V+.
        0x0030:  92ee 3227 4745 5420 2f20 4854 5450 2f31  ..2'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                   :.*/*....
13:58:54.171934 IP 74.125.135.139.80 > 172.17.0.2.49896: Flags [.], ack 86, win 256, options [nop,nop,TS val 2465083943 ecr 3
478531056], length 0
        0x0000:  4560 0034 0000 4000 7e06 7e48 4a7d 878b  E`.4..@.~.~HJ}..
        0x0010:  ac11 0002 0050 c2e8 17f7 dc97 c706 fe63  .....P.........c
        0x0020:  8010 0100 ba12 0000 0101 080a 92ee 3227  ..............2'
        0x0030:  cf56 2bf0                                .V+.
13:58:54.176715 IP 74.125.135.139.80 > 172.17.0.2.49896: Flags [P.], seq 1:583, ack 86, win 256, options [nop,nop,TS val 2465
083948 ecr 3478531056], length 582: HTTP: HTTP/1.1 301 Moved Permanently
        0x0000:  4580 027a 0000 4000 7e06 7be2 4a7d 878b  E..z..@.~.{.J}..
        0x0010:  ac11 0002 0050 c2e8 17f7 dc97 c706 fe63  .....P.........c
        0x0020:  8018 0100 7cb4 0000 0101 080a 92ee 322c  ....|.........2,
        0x0030:  cf56 2bf0 4854 5450 2f31 2e31 2033 3031  .V+.HTTP/1.1.301
        0x0040:  204d 6f76 6564 2050 6572 6d61 6e65 6e74  .Moved.Permanent
        0x0050:  6c79 0d0a 4c6f 6361 7469 6f6e 3a20 6874  ly..Location:.ht
        0x0060:  7470 733a 2f2f 6f70 656e 736f 7572 6365  tps://opensource
        0x0070:  2e67 6f6f 676c 652f 0d0a 4372 6f73 732d  .google/..Cross-
        0x0080:  4f72 6967 696e 2d52 6573 6f75 7263 652d  Origin-Resource-
        0x0090:  506f 6c69 6379 3a20 6372 6f73 732d 6f72  Policy:.cross-or
        0x00a0:  6967 696e 0d0a 436f 6e74 656e 742d 5479  igin..Content-Ty
        0x00b0:  7065 3a20 7465 7874 2f68 746d 6c3b 2063  pe:.text/html;.c
        0x00c0:  6861 7273 6574 3d55 5446 2d38 0d0a 582d  harset=UTF-8..X-
        0x00d0:  436f 6e74 656e 742d 5479 7065 2d4f 7074  Content-Type-Opt
```

I run tcpdump again with these new flags:
-X: Display ASCII and hexadecimal output. This output can be analyzed to detect malware signatures and aid in forensic analysis.

Summary

In this lab experience I identified network interfaces, captured network data with tcpdump, interpreted tcpdump output and saved and analyzed packet data.

References

1. Heavyd (2015) H*ow to make tcpdump to display ip and port number but not hostname and protocol* https://superuser.com/questions/587302/how-to-make-tcpdump-to-display-ip-and-port-number-but-not-hostname-and-protocol.
2. Google Cybersecurity Certificate (2023) Capture your first packet