

# Capturing and Filtering Network Traffic with Tcpcmdump

Tcpcmdump is a network packet analyzer that utilizes a command line interface. In this project tcpcmdump is run inside of a Linux virtual machine to demonstrate how it can capture and filter web traffic.

## Identifying network interfaces

The first thing to do in this project is to identify the network interfaces that could be used to capture packet data. We do this by using the “sudo ifconfig” command.

```
analyst@50d02b144737:~$ 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 1423  bytes 13798222 (13.1 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 914  bytes 75509 (73.7 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 66  bytes 9409 (9.1 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 66  bytes 9409 (9.1 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

analyst@50d02b144737:~$
```

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

So here we will use *eth0* as the interface that you will capture network packet data from.

On systems where the “ifconfig” command is not available, “sudo tcpcmdump -D” could be used.

```
analyst@50d02b144737:~$ sudo tcpcmdump -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@50d02b144737:~$
```

## Inspecting the network traffic of a network interface

To filter live packet data from the *eth0* interface, we begin by typing in the following command: “sudo tcpcmdump -i eth0 -v -c5”.

```
analyst@50d02b144737:~$ sudo tcpdump -i eth0 -v -c5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:56:58.693267 IP (tos 0x0, ttl 64, id 40728, offset 0, flags [DF], proto TCP (6), length 113)
    50d02b144737.5000 > nginx-us-east1-c.c.qwiklabs-terminal-vms-prod-00.internal.41662: Flags [P.], cksu
    um 0x588e (incorrect -> 0x2d62), seq 1320386951:1320387012, ack 2211840941, win 491, options [nop,nop,TS
    val 1282563350 ecr 720789744], length 61
14:56:58.693620 IP (tos 0x0, ttl 63, id 35003, offset 0, flags [DF], proto TCP (6), length 52)
    nginx-us-east1-c.c.qwiklabs-terminal-vms-prod-00.internal.41662 > 50d02b144737.5000: Flags [.], cksu
    m 0xd865 (correct), ack 61, win 507, options [nop,nop,TS val 720789873 ecr 1282563350], length 0
14:56:58.704247 IP (tos 0x0, ttl 64, id 40729, offset 0, flags [DF], proto TCP (6), length 147)
    50d02b144737.5000 > nginx-us-east1-c.c.qwiklabs-terminal-vms-prod-00.internal.41662: Flags [P.], cksu
    um 0x58b0 (incorrect -> 0x1a16), seq 61:156, ack 1, win 491, options [nop,nop,TS val 1282563361 ecr 7207
    89873], length 95
14:56:58.704615 IP (tos 0x0, ttl 63, id 35004, offset 0, flags [DF], proto TCP (6), length 52)
    nginx-us-east1-c.c.qwiklabs-terminal-vms-prod-00.internal.41662 > 50d02b144737.5000: Flags [.], cksu
    m 0xd7f0 (correct), ack 156, win 507, options [nop,nop,TS val 720789884 ecr 1282563361], length 0
14:56:58.727423 IP (tos 0x0, ttl 64, id 29781, offset 0, flags [DF], proto UDP (17), length 69)
    50d02b144737.55818 > metadata.google.internal.domain: 59430+ PTR? 2.0.21.172.in-addr.arpa. (41)
5 packets captured
8 packets received by filter
0 packets dropped by kernel
```

Let's break down each individual part of the command:

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

-v: Display detailed packet data (verbose).

-c5: Capture 5 packets of data.

In the example data at the start of the packet output, tcpdump reported that it was listening on the *eth0* interface, and it provided information on the link type and the capture size in bytes:

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

On the next line, the first field is the packet's timestamp, followed by the protocol type, IP:

```
14:56:58.693267 IP
```

The verbose option, -v, has provided more details about the IP packet fields, such as TOS, TTL, offset, flags, internal protocol type (in this case, TCP (6)), and the length of the outer IP packet in bytes:

```
(tos 0x0, ttl 64, id 40728, offset 0, flags [DF], proto TCP (6), length 113)
```

In the next section, the data shows the systems that are communicating with each other:

```
50d02b144737.5000 > nginx-us-east1-c.c.qwiklabs-terminal-vms-prod-00.internal.41662:
```

Tcpdump will convert IP addresses into names, as can be seen in the screenshot. The name of the Linux virtual machine used in the project, also included in the command prompt, appears here as the source for one packet and the destination for the second packet. The direction of the arrow (>) indicates the direction of the traffic flow in each packet. Each system name includes a suffix with the port number (.5000 in this case), which is used by the source and the destination systems for this packet.

The remaining data filters the header data for the inner TCP packet:

```
50d02b144737.5000 > nginx-us-east1-c.c.qwiklabs-terminal-vms-prod-00.internal.41662: Flags [P.], cksu
um 0x588e (incorrect -> 0x2d62), seq 1320386951:1320387012, ack 2211840941, win 491, options [nop,nop,TS
val 1282563350 ecr 720789744], length 61
```

The flags field identifies TCP flags. In this case, the P represents the push flag and the period indicates it's an ACK flag. This means the packet is pushing out data.

The next field is the TCP checksum value, which is used for detecting errors in the data.

This section also includes the sequence and acknowledgment numbers, the window size, and the length of the inner TCP packet in bytes.

## Capture network traffic with tcpdump

Now, let's save the captured network data to a packet capture file. To do this, we will use the command "sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &"

```
analyst@f9bd8cf17bf5:~$ sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &
[1] 12869
analyst@f9bd8cf17bf5:~$ tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

\*Note: This part of the project was run on a separate VM, so the name of the machine is different from the previous section.

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

*-i eth0*: Capture data from the eth0 interface.

*-nn*: Do not attempt to resolve IP addresses or ports to names. This is considered 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 (default HTTP) traffic. We want to save a small sample that contains only web (TCP port 80) network packet data.

*-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.

To generate some HTTP traffic that can be captured, let's run the *curl* command to open the website *opensource.google.com*

```
analyst@f9bd8cf17bf5:~$ 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@f9bd8cf17bf5:~$ 9 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Now, to verify that the data was captured and written onto the *.pcap* file, the command "ls -l capture.pcap" is used.

```
ls -l capture.pcap
-rw-r--r-- 1 root root 1401 Oct 17 15:40 capture.pcap
[1]+  Done                  sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap
```

## Filtering the captured packet data

To filter the data that was collected, there are several commands that could be used.

For example, “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.

The -nn switch is specified again here, as one would want to make sure tcpdump does not perform name lookups of either IP addresses or ports, since this can alert threat actors.

```
analyst@94d8ef17b45:~$ sudo tcpdump -nn -r capture.pcap -v
reading from file capture.pcap, link-type EN10MB (Ethernet)
15:40:47.082285 IP (tos 0x0, ttl 64, id 38152, offset 0, flags [DF], proto TCP (6), length 60)
  172.18.0.2.39242 > 142.251.183.102.80: Flags [S], cksum 0x12a4 (incorrect -> 0x0a84), seq 2021357507, win 32660, options [max 1420,sackOK,TS val 1744324808 ecr 0,nop,wscale 6], length 0
15:40:47.083352 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 60)
  142.251.183.102.80 > 172.18.0.2.39242: Flags [S.], cksum 0x25da (correct), seq 4071931091, ack 2021357508, win 65535, options [max 1420,sackOK,TS val 2623187272 ecr 1744324808,nop,wscale 8], length 0
15:40:47.083369 IP (tos 0x0, ttl 64, id 38153, offset 0, flags [DF], proto TCP (6), length 52)
  172.18.0.2.39242 > 142.251.183.102.80: Flags [L], cksum 0xf29c (incorrect -> 0x327f), ack 1, win 511, options [nop,nop,TS val 1744324809 ecr 2623187272], length 0
15:40:47.083452 IP (tos 0x0, ttl 64, id 38154, offset 0, flags [DF], proto TCP (6), length 137)
  172.18.0.2.39242 > 142.251.183.102.80: Flags [P.], cksum 0xf2f1 (incorrect -> 0xc132), seq 1:86, ack 1, win 511, options [nop,nop,TS val 1744324809 ecr 2623187272], length 85: HTTP, length: 85
    GET / HTTP/1.1
    Host: opensource.google.com
    User-Agent: curl/7.64.0
    Accept: */*

15:40:47.083667 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 52)
  142.251.183.102.80 > 172.18.0.2.39242: Flags [L], cksum 0x500e (correct), ack 86, win 1051, options [nop,nop,TS val 2623187272 ecr 1744324809], length 0
15:40:47.084799 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 590)
  142.251.183.102.80 > 172.18.0.2.39242: Flags [P.], cksum 0x40cd (correct), seq 1:539, ack 86, win 1051, options [nop,nop,TS val 2623187277 ecr 1744324809], 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
    Date: Thu, 17 Oct 2024 15:40:47 GMT
    Expires: Thu, 17 Oct 2024 16:10:47 GMT
    Cache-Control: public, max-age=1800
    Server: gfe
    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>

15:40:47.088812 IP (tos 0x0, ttl 64, id 38155, offset 0, flags [DF], proto TCP (6), length 52)
  172.18.0.2.39242 > 142.251.183.102.80: Flags [L], cksum 0xf29c (incorrect -> 0x500e), ack 539, win 503, options [nop,nop,TS val 1744324814 ecr 2623187277], length 0
15:40:47.090103 IP (tos 0x0, ttl 64, id 38156, offset 0, flags [DF], proto TCP (6), length 52)
  172.18.0.2.39242 > 142.251.183.102.80: Flags [F.], cksum 0xf29c (incorrect -> 0x500c), seq 86, ack 539, win 503, options [nop,nop,TS val 1744324815 ecr 2623187277], length 0
15:40:47.090328 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 52)
  142.251.183.102.80 > 172.18.0.2.39242: Flags [P.], cksum 0x4de5 (correct), seq 539, ack 87, win 1051, options [nop,nop,TS val 2623187279 ecr 1744324815], length 0
```

As before, we can see IP packet information along with information about the data that the packet contains.

To filter extended packet data from the .pcap file, the command “sudo tcpdump -nn -r capture.pcap -X” can be used.

```

analyst@8f9bd8ef17bf51:~$ sudo tcpdump -nn -r capture.pcap -X
reading from file capture.pcap, link-type EN10MB (Ethernet)
15:40:47.082285 IP 172.18.0.2.39242 > 142.251.183.102.80: Flags [S], seq 2021357507, win 32660, options [mss 1420,sackOK,TS val 1744324808 ecr 0,nop,wscale 6], length 0
0x0000: 4500 003c 9508 4000 4006 b33d ac12 0002  E...8...w....
0x0010: 8efb b766 994a 0050 787b 77c4 0800 0800  ...f..P...w....
0x0020: a052 7f94 f2a4 0000 0204 058e 0402 080a  ....P...k(w...
0x0030: 67f8 48c8 0000 0000 0103 0306  ....g.H.....
15:40:47.083352 IP 142.251.183.102.80 > 172.18.0.2.39242: Flags [S.], seq 4071931091, ack 2021357508, win 63535, options [mss 1420,sackOK,TS val 2623187272 ecr 1744324808,nop,wscale 8], length 0
0x0000: 4500 003c 0000 4000 7a06 0a4e 8efb b766  E...8...f...f
0x0010: ac12 0002 0050 994a f2b4 bcd3 787b 77c4  ....P..J...k(w...
0x0020: a012 ffff 25da 0000 0204 058e 0402 080a  ....P...k(w...
0x0030: 9c5a a948 c7f5 40c8 0103 0308  ....J.Hp.H....
15:40:47.083369 IP 172.18.0.2.39242 > 142.251.183.102.80: Flags [J.], ack 1, win 511, options [nop,nop,TS val 1744324809 ecr 2623187272], length 0
0x0000: 4500 0034 9509 4000 4006 b344 ac12 0002  E..4..8...D...
0x0010: 8efb b766 994a 0050 787b 77c4 f2b4 bcd4  ...f..J.P...w....
0x0020: 8010 01ff f23c 0000 0101 080a 67f8 48c9  ....g.H.....
0x0030: 9c5a a948
15:40:47.083452 IP 172.18.0.2.39242 > 142.251.183.102.80: Flags [P.], seq 1:86, ack 1, win 511, options [nop,nop,TS val 1744324809 ecr 2623187272], length 85: HTTP: GET / HTTP/1.1
0x0000: 4500 0089 950a 4000 4006 b2ee ac12 0002  E....8..8.....
0x0010: 8efb b766 994a 0050 787b 77c4 f2b4 bcd4  ...f..J.P...w....
0x0020: 8018 01ff f2f1 0000 0101 080a 67f8 48c9  ....g.H.....
0x0030: 9c5a a948 4743 5420 2f20 48f4 5450 2f21  ..J.HGET / HTTP/1
0x0040: 2a31 0d0a 486f 7374 3a20 6f70 656e 736f  ..l..Host: openso
0x0050: 7572 6365 2a67 6f66 676c 652e 636f 680d  urce.google.com.
0x0060: 0a55 7365 722a 4167 656e 74ba 2063 7572  User-Agent: cur
0x0070: 6c2f 372e 3634 2a30 0d0a 4163 6365 7074  /7.64.0..accept
0x0080: 3a20 2a2f 2a0d 0a0d 0a
15:40:47.083667 IP 142.251.183.102.80 > 172.18.0.2.39242: Flags [J.], ack 86, win 1051, options [nop,nop,TS val 2623187272 ecr 1744324809], length 0
0x0000: 4500 0034 0000 4000 7a06 0a4e 8efb b766  E..4..8...N...f
0x0010: ac12 0002 0050 994a f2b4 bcd4 787b 7819  ....P..J...k(x.
0x0020: 8010 041b 300e 0000 0101 080a 9c5a a948  ....P.....J.H
0x0030: 67f8 48c9
15:40:47.088799 IP 142.251.183.102.80 > 172.18.0.2.39242: Flags [P.], seq 1:539, ack 86, win 1051, options [nop,nop,TS val 2623187277 ecr 1744324809], length 538: HTTP: HTTP/1.1 301 Moved Permanently
0x0000: 4500 024e 0000 4000 7a06 0834 8efb b766  E..H..8...4...f
0x0010: ac12 0002 0050 994a f2b4 bcd4 787b 7819  ....P..J...k(x.
0x0020: 8018 041b 40c4 0000 0101 080a 9c5a a94d  ....8.....J.H
0x0030: 67f8 48c9 4854 5450 2f21 2a31 2033 3031  g.H.HTTP/1.1.301
0x0040: 2045 6f76 6564 2050 6572 6863 6a65 6a74  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: 2a67 6f66 676c 652e 0d0a 436f 6a74 656e  -google...Conten
0x0080: 742d 5479 7065 3a20 7465 7874 2668 746d  t-Type: text/htm
0x0090: 6c3b 2063 6861 7273 6574 3d55 5446 2d38  l; charset=UTF-8
0x00a0: 0d0e 582d 636f 6a74 656e 742d 5479 7065  ..-Content-Type
0x00b0: 2d4f 7074 696f 6a73 3a20 6a6f 736e 696e  -Options: no-cif
0x00c0: 660d 0a44 6174 653a 2054 6875 2c20 3137  f..Date: Thu, 17
0x00d0: 204f 6374 2032 3032 3a20 3135 3a24 303a  Oct. 2024. 15:40:
0x00e0: 3437 2047 4d54 0d0a 4578 7069 7265 733a  47 GMT, Expires:
0x00f0: 2054 6875 2c20 3137 204f 6374 2032 3032  Thu, 17 Oct. 202
0x100: 3420 3136 3a31 303a 3437 2047 4d54 0d0a  4.16:10:47 GMT.
0x110: 4361 6368 652d 436f 6a74 726f 6a3a 2070  Cache-Control: p
0x120: 7562 6c69 632c 206d 6178 2d61 6765 3d31  ublic, max-age=1
0x130: 3830 300d 0a53 6372 7665 723a 2073 6466  800..Server: aff
0x140: 450d 0a43 6f6e 7465 6a74 2d4e 436e 6774  e..Content-Lengt
0x150: 683a 2032 3233 0d0a 5d2d 5853 532d 5072  hi.221.X-XSS-Pro

```

The difference here is the -X option which displays the hexadecimal and ASCII output format packet data. Such output could be analyzed to detect patterns or anomalies during malware analysis or forensic analysis.