Menu

12 Tcpdump Commands – A Network Sniffer Tool

Narad Shrestha | Last Updated: September 1, 2021 | Linux Commands, Networking Commands | 30 Comments

In our previous article, we have seen 20 Netstat Commands (netstat now replaced by ss command) to monitor or manage a Linux network. This is our another ongoing series of packet sniffer tool called **tcpdump**. Here, we are going to show you how to install **tcpdump** and then we discuss and cover some useful commands with their practical examples.

Linux tcpdump command examples

tcpdump is a most powerful and widely used command-line packets sniffer or package analyzer tool which is used to capture or filter **TCP/IP** packets that are received or transferred over a network on a specific interface.

[You might also like: 16 Useful Bandwidth Monitoring Tools to Analyze Network Usage in Linux **]**

It is available under most of the **Linux/Unix-based** operating systems. tcpdump also gives us an option to save captured packets in a file for future analysis. It saves the file in a **pcap** format, that can be viewed by tcpdump command or an open-source GUI-based tool called Wireshark (Network Protocol Analyzer) that reads tcpdump **pcap** format files.

How to Install tcpdump in Linux

Many Linux distributions already shipped with the **tcpdump** tool, if in case you don't have it on a system, you can install it using either of the following commands.

```
$ sudo apt-get install tcpdump  [On Debian, Ubuntu and Mint]
$ sudo yum install tcpdump           [On RHEL/CentOS/Fedora and Ro
$ sudo emerge -a sys-apps/tcpdump    [On Gentoo Linux]
$ sudo pacman -S tcpdump             [On Arch Linux]
$ sudo zypper install tcpdump        [On OpenSUSE]
```

Getting Started with tcpdump Command Examples

Once the **tcpdump** tool is installed on your system, you can continue to browse following commands with their examples.

1. Capture Packets from Specific Interface

The command screen will scroll up until you interrupt and when we execute the **tcpdump** command it will captures from all the interfaces, however with `-i` switch only capture from the desired interface.

```
# tcpdump -i eth0

tcpdump: verbose output suppressed, use -v or -vv for full protoco
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
11:33:31.976358 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspoole
11:33:31.976603 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ss
11:33:31.977243 ARP, Request who-has tecmint.com tell 172.16.25.12
11:33:31.977359 ARP, Reply tecmint.com is-at 00:14:5e:67:26:1d (ou
11:33:31.977367 IP 172.16.25.126.54807 > tecmint.com: 4240+ PTR? 1
11:33:31.977599 IP tecmint.com > 172.16.25.126.54807: 4240 NXDomai

11:33:31.977742 IP 172.16.25.126.44519 > tecmint.com: 40988+ PTR?
11:33:32.028747 IP 172.16.20.33.netbios-ns > 172.16.31.255.netbios
11:33:32.112045 IP 172.16.21.153.netbios-ns > 172.16.31.255.netbio
11:33:32.115606 IP 172.16.21.144.netbios-ns > 172.16.31.255.netbio
11:33:32.156576 ARP, Request who-has 172.16.16.37 tell old-oracleh
11:33:32.348738 IP tecmint.com > 172.16.25.126.44519: 40988 NXDoma
```

2. Capture Only N Number of Packets

When you run the **tcpdump** command it will capture all the packets for the specified interface, until you **hit** the cancel button. But using `-c` option, you can capture a specified number of packets. The below example will only capture **6** packets.

```
# tcpdump -c 5 -i eth0

tcpdump: verbose output suppressed, use -v or -vv for full prot
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
```

```
11:40:20.281355 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspooler
11:40:20.281586 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ssh
11:40:20.282244 ARP, Request who-has tecmint.com tell 172.16.25.126
11:40:20.282360 ARP, Reply tecmint.com is-at 00:14:5e:67:26:1d (oui
11:40:20.282369 IP 172.16.25.126.53216 > tecmint.com.domain: 49504+
11:40:20.332494 IP tecmint.com.netbios-ssn > 172.16.26.17.nimaux: P
6 packets captured
23 packets received by filter
0 packets dropped by kernel
```

3. Print Captured Packets in ASCII

The below **tcpdump** command with the option `-A` displays the package in **ASCII** format. It is a character-encoding scheme format.

```
# tcpdump -A -i eth0

tcpdump: verbose output suppressed, use -v or -vv for full protocol
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
09:31:31.347508 IP 192.168.0.2.ssh > 192.168.0.1.nokia-ann-ch1: Fla
M.r0...vUP.E.X.......~.%..>N..oFk.........KQ..)Eq.d.,....r^l.....r
09:31:31.347760 IP 192.168.0.1.nokia-ann-ch1 > 192.168.0.2.ssh: Fla
M....vU.r1~P..._..........
^C09:31:31.349560 IP 192.168.0.2.46393 > b.resolvers.Level3.net.dom
E..F..@.@.............9.5.2.f+............1.0.168.192.in-addr.arpa.

3 packets captured
11 packets received by filter
0 packets dropped by kernel
```

4. Display Available Interfaces

To list the number of available interfaces on the system, run the following command with `-D` option.

```
# tcpdump -D

 1.eth0
2.eth1
3.usbmon1 (USB bus number 1)
4.usbmon2 (USB bus number 2)
5.usbmon3 (USB bus number 3)
6.usbmon4 (USB bus number 4)
7.usbmon5 (USB bus number 5)
8.any (Pseudo-device that captures on all interfaces)
9.lo
```

5. Display Captured Packets in HEX and ASCII

The following command with option `-xx` capture the data of each packet, including its link level header in **HEX** and **ASCII** format.

```
# tcpdump -XX -i eth0

11:51:18.974360 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspoole
        0x0000:  b8ac 6f2e 57b3 0001 6c99 1468 0800 4510  ..o.W...
        0x0010:  00ec 8783 4000 4006 275d ac10 197e ac10  ....@.@.
        0x0020:  197d 0016 1129 d12a af51 d9b6 d5ee 5018  .}...).*
        0x0030:  4948 8bfa 0000 0e12 ea4d 22d1 67c0 f123  IH.....
        0x0040:  9013 8f68 aa70 29f3 2efc c512 5660 4fe8  ...h.p).
        0x0050:  590a d631 f939 dd06 e36a 69ed cac2 95b6  Y..1.9..
        0x0060:  f8ba b42a 344b 8e56 a5c4 b3a2 ed82 c3a1  ...*4K.V
        0x0070:  80c8 7980 11ac 9bd7 5b01 18d5 8180 4536  ..y.....
        0x0080:  30fd 4f6d 4190 f66f 2e24 e877 ed23 8eb0  0.OmA..o
        0x0090:  5a1d f3ec 4be4 e0fb 8553 7c85 17d9 866f  Z...K...
        0x00a0:  c279 0d9c 8f9d 445b 7b01 81eb 1b63 7f12  .y....
        0x00b0:  71b3 1357 52c7 cf00 95c6 c9f6 63b1 ca51  q..WR...
        0x00c0:  0ac6 456e 0620 38e6 10cb 6139 fb2a a756  ..En..8.
```

```
        0x00d0:  37d6 c5f3 f5f3 d8e8 3316 d14f d7ab fd93  7.......
        0x00e0:  1137 61c1 6a5c b4d1 ddda 380a f782 d983  .7a.j\..
        0x00f0:  62ff a5a9 bb39 4f80 668a                 b....9O.
11:51:18.974759 IP 172.16.25.126.60952 > mddc-01.midcorp.mid-day.c
        0x0000:  0014 5e67 261d 0001 6c99 1468 0800 4500  ..^g&...
        0x0010:  0048 5a83 4000 4011 5e25 ac10 197e ac10  .HZ.@.@.
        0x0020:  105e ee18 0035 0034 8242 391c 0100 0001  .^...5.4
        0x0030:  0000 0000 0000 0331 3235 0232 3502 3136  .......1
        0x0040:  0331 3732 0769 6e2d 6164 6472 0461 7270  .172.in-
        0x0050:  6100 000c 0001                           a.....
```

6. Capture and Save Packets in a File

As we said, that **tcpdump** has a feature to capture and save the file in a **.pcap** format, to do this just execute the command with `-w` option.

```
# tcpdump -w 0001.pcap -i eth0

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture s
4 packets captured
4 packets received by filter
0 packets dropped by kernel
```

7. Read Captured Packets File

To read and analyze captured packet **0001.pcap** file use the command with `-r` option, as shown below.

```
# tcpdump -r 0001.pcap

reading from file 0001.pcap, link-type EN10MB (Ethernet)
09:59:34.839117 IP 192.168.0.2.ssh > 192.168.0.1.nokia-ann-ch1: Fl
```

```
09:59:34.963022 IP 192.168.0.1.nokia-ann-ch1 > 192.168.0.2.ssh: Fl
09:59:36.935309 IP 192.168.0.1.netbios-dgm > 192.168.0.255.netbios
09:59:37.528731 IP 192.168.0.1.nokia-ann-ch1 > 192.168.0.2.ssh: Fl
```

8. Capture IP Address Packets

To capture packets for a specific interface, run the following command with option `-n` .

```
# tcpdump -n -i eth0

tcpdump: verbose output suppressed, use -v or -vv for full protoco
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
12:07:03.952358 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspoole
12:07:03.952602 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ss
12:07:03.953311 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspoole
12:07:03.954288 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspoole
12:07:03.954502 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ss
12:07:03.955298 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspoole
12:07:03.955425 IP 172.16.23.16.netbios-ns > 172.16.31.255.netbios
12:07:03.956299 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspoole
12:07:03.956535 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ss
```

9. Capture only TCP Packets.

To capture packets based on **TCP** port, run the following command with option **tcp**.

```
# tcpdump -i eth0 tcp

tcpdump: verbose output suppressed, use -v or -vv for full prot
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
```

```
12:10:36.216358 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspooler
12:10:36.216592 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ssh
12:10:36.219069 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspooler
12:10:36.220039 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspooler
12:10:36.220260 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ssh
12:10:36.222045 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspooler
12:10:36.223036 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspooler
12:10:36.223252 IP 172.16.25.125.apwi-rxspooler > 172.16.25.126.ssh
^C12:10:36.223461 IP mid-pay.midcorp.mid-day.com.netbios-ssn > 172
```

10. Capture Packet from Specific Port

Let's say you want to capture packets for specific port 22, execute the below command by specifying port number **22** as shown below.

```
# tcpdump -i eth0 port 22

tcpdump: verbose output suppressed, use -v or -vv for full protocol
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
10:37:49.056927 IP 192.168.0.2.ssh > 192.168.0.1.nokia-ann-ch1: Fla
10:37:49.196436 IP 192.168.0.2.ssh > 192.168.0.1.nokia-ann-ch1: Fla
10:37:49.196615 IP 192.168.0.1.nokia-ann-ch1 > 192.168.0.2.ssh: Fla
```

Learn Linux in One Week and Go From Zero to Hero - Get This Book

```
10:37:49.381322 IP 192.168.0.1.nokia-ann-ch1 > 192.168.0.2.ssh: Fla
```

11. Capture Packets from source IP

To capture packets from source **IP**, say you want to capture packets for **192.168.0.2**, use the command as follows.

```
# tcpdump -i eth0 src 192.168.0.2
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
10:49:15.746474 IP 192.168.0.2.ssh > 192.168.0.1.nokia-ann-ch1: Fla
10:49:15.748554 IP 192.168.0.2.56200 > b.resolvers.Level3.net.domai
10:49:15.912165 IP 192.168.0.2.56234 > b.resolvers.Level3.net.domai
10:49:16.074720 IP 192.168.0.2.33961 > b.resolvers.Level3.net.domai
```

12. Capture Packets from destination IP

To capture packets from destination **IP**, say you want to capture packets for **50.116.66.139**, use the command as follows.

```
# tcpdump -i eth0 dst 50.116.66.139

tcpdump: verbose output suppressed, use -v or -vv for full protocol
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
10:55:01.798591 IP 192.168.0.2.59896 > 50.116.66.139.http: Flags [
10:55:05.527476 IP 192.168.0.2.59894 > 50.116.66.139.http: Flags [
10:55:05.626027 IP 192.168.0.2.59894 > 50.116.66.139.http: Flags [
```

This article may help you to explore the **tcpdump** command in-depth and also to capture and analyze packets in the future. There are a number of options available, you can use the options as per your requirement. Please share if you find this article useful through our comment box.

‹ How to Create eLearning Platform with Moodle and ONLYOFFICE

How to Install WordPress on Rocky Linux 8 ›

> If you liked this article, then do subscribe to email alerts for Linux tutorials. If you have any questions or doubts? do ask for help in the comments section.