sanfoundry.com

10+ "tcpdump" Command Usage Examples in Linux

by linuxcmd2

16-20 minutes

This tutorial explains Linux “tcpdump” command, options and its usage with examples.This tutorial explains Linux “tr” command, options and its usage with examples.

tcpdump – dump traffic on a network

DESCRIPTION

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 received or transferred over a network on a specific interface. It is available under most of the Linux/Unix based operating systems. tcpdump also gives us a option to save captured packets in a file for future analysis.

The tcpdump command prints the headers of packets on a network interface that match the boolean expression. You can run the command with the -w flag to save the packet data in a file for further analysis. You can also run the command with the -r flag to read data from a saved packet file instead reading the packets from a network interface. In all cases, only packets that match expression is processed by the tcpdump command.

SYNOPSIS

tcpdump [-adeflnNOpqRStuvxX] [-c count] [-C file_size] [-F file] [-i interface] [-m module] [-r file] [-s snaplen] [-T type] [-U user] [-w file] [-E algo:secret] [expression]

OPTIONS :

-a
Attempt to convert network and broadcast addresses to names.
-c
Exit after receiving count packets.
-C
Before writing a raw packet to a savefile, check whether the file is currently larger than file_size and, if so, close the current savefile and open a new one. Savefiles after the first savefile will have the name specified with the -w flag, with a number after it, starting at 2 and continuing upward. The units of file_size are millions of bytes (1,000,000 bytes, not 1,048,576 bytes).
-d
Dump the compiled packet-matching code in a human readable form to standard output and stop.
-dd
Dump packet-matching code as a C program fragment.
-ddd
Dump packet-matching code as decimal numbers (preceded with a count).
-e
Print the link-level header on each dump line.
-E
Use algo:secret for decrypting IPsec ESP packets. Algorithms may be des-cbc, 3des-cbc, blowfish-cbc, rc3-cbc, cast128-cbc, or none. The default is des-cbc.
-f
Print `foreign' internet addresses numerically rather than symbolically .
-F
Use file as input for the filter expression. An additional expression given on the command line is ignored.
-i
Listen on interface. If unspecified, tcpdump searches the system interface list for the lowest numbered, configured up interface (excluding loopback). Ties are broken by choosing the earliest match.
-l
Make stdout line buffered. Useful if you want to see the data while capturing it. E.g.,
“tcpdump -l | tee dat” or “tcpdump -l > dat & tail -f dat”.
-n
Don’t convert host addresses to names. This can be used to avoid DNS lookups.
-nn
Don’t convert protocol and port numbers etc. to names either.
-N
Don’t print domain name qualification of host names. E.g., if you give this flag then tcpdump will print “nic” instead of “nic.ddn.mil”.
-O
Do not run the packet-matching code optimizer. This is useful only if you suspect a bug in the optimizer.
-p
Don’t put the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, `-p' cannot be used as an abbreviation for `ether host {local-hw-addr} or ether broadcast'.
-q
Quick (quiet?) output. Print less protocol information so output lines are shorter.
-r
Read packets from file (which was created with the -w option). Standard input is used if file is "-".
-S
Print absolute, rather than relative, TCP sequence numbers.
-s
Snarf snaplen bytes of data from each packet rather than the default of 68 . 68 bytes is adequate for IP, ICMP, TCP and UDP but may truncate protocol information from name server and NFS packets .
-T
Force packets selected by “expression” to be interpreted the specified type. Currently known types are cnfp (Cisco NetFlow protocol), rpc (Remote Procedure Call), rtp (Real-Time Applications protocol), rtcp (Real-Time Applications control protocol), snmp (Simple Network Management Protocol), vat (Visual Audio Tool),

and wb (distributed White Board).

-t

Don't print a timestamp on each dump line.

-tt

Print an unformatted timestamp on each dump line.

-U

Drops root privileges and changes user ID to user and group ID to the primary group of user.

-ttt

Print a delta (in micro-seconds) between current and previous line on each dump line.

-tttt

Print a timestamp in default format proceeded by date on each dump line.

-u

Print undecoded NFS handles.

-v

(Slightly more) verbose output. For example, the time to live, identification, total length and options in an IP packet are printed. Also enables additional packet integrity checks such as verifying the IP and ICMP header checksum.

-vv

Even more verbose output. For example, additional fields are printed from NFS reply packets, and SMB packets are fully decoded.

-vvv

Even more verbose output. For example, telnet SB ... SE options are printed in full. With -X telnet options are printed in hex as well.

-w

Write the raw packets to file rather than parsing and printing them out. They can later be printed with the -r option. Standard output is used if file is "-".

-x

Print each packet (minus its link level header) in hex. The smaller of the entire packet or snaplen bytes will be printed. Note that this is the entire link-layer packet, so for link layers that pad (e.g. Ethernet), the padding bytes will also be printed when the higher layer packet is shorter than the required padding.

-X

When printing hex, print ascii too. Thus if -x is also set, the packet is printed in hex/ascii. This is very handy for analysing new protocols. Even if -x is not also set, some parts of some packets may be printed in hex/ascii.

EXAMPLES

1. Capture Packets from Specific Interface

```
# tcpdump -i eth0

tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
11:33:31.976358 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
3500440357:3500440553, ack 3652628334, win 18760,
length 196
11:33:31.976603 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 196, win 64487,
length 0
11:33:31.977243 ARP, Request who-has tecmint.com
tell 172.16.25.126, length 28
11:33:31.977359 ARP, Reply tecmint.com is-at
00:14:5e:67:26:1d (oui Unknown), length 46
11:33:31.977367 IP 172.16.25.126.54807 >
tecmint.com: 4240+ PTR? 125.25.16.172.in-
addr.arpa. (44)
11:33:31.977599 IP tecmint.com >
172.16.25.126.54807: 4240 NXDomain 0/1/0 (121)
11:33:31.977742 IP 172.16.25.126.44519 >
tecmint.com: 40988+ PTR? 126.25.16.172.in-
addr.arpa. (44)
11:33:32.028747 IP 172.16.20.33.netbios-ns >
172.16.31.255.netbios-ns: NBT UDP PACKET(137):
QUERY; REQUEST; BROADCAST
11:33:32.112045 IP 172.16.21.153.netbios-ns >
172.16.31.255.netbios-ns: NBT UDP PACKET(137):
QUERY; REQUEST; BROADCAST
11:33:32.115606 IP 172.16.21.144.netbios-ns >
172.16.31.255.netbios-ns: NBT UDP PACKET(137):
QUERY; REQUEST; BROADCAST
11:33:32.156576 ARP, Request who-has 172.16.16.37
tell old-oraclehp1.midcorp.mid-day.com, length 46
11:33:32.348738 IP tecmint.com >
172.16.25.126.44519: 40988 NXDomain 0/1/0 (121)
```

2. Capture Only N Number of Packets

```
# tcpdump -c 5 -i eth0

tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
11:40:20.281355 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
3500447285:3500447481, ack 3652629474, win 18760,
length 196
11:40:20.281586 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 196, win 65235,
length 0
11:40:20.282244 ARP, Request who-has tecmint.com
tell 172.16.25.126, length 28
11:40:20.282360 ARP, Reply tecmint.com is-at
00:14:5e:67:26:1d (oui Unknown), length 46
11:40:20.282369 IP 172.16.25.126.53216 >
tecmint.com.domain: 49504+ PTR? 125.25.16.172.in-
addr.arpa. (44)
11:40:20.332494 IP tecmint.com.netbios-ssn >
172.16.26.17.nimaux: Flags [P.], seq
3058424861:3058424914, ack 693912021, win 64190,
length 53 NBT Session Packet: Session Message
6 packets captured
```

```
23 packets received by filter
0 packets dropped by kernel
```

3. Print Captured Packets in ASCII

```
# tcpdump -A -i eth0

tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
09:31:31.347508 IP 192.168.0.2.ssh >
192.168.0.1.nokia-ann-ch1: Flags [P.], seq
3329372346:3329372542, ack 4193416789, win 17688,
length 196
M.r0...vUP.E.X.......~.%..>N..oFk.........KQ..)Eq.d.,....r^l......m\.oyE....-....g~m..Xy.6..l.....c.O.@...o_..J....i.*.....2f.mQH...Q.c.
09:31:31.347760 IP 192.168.0.1.nokia-ann-ch1 >
192.168.0.2.ssh: Flags [.], ack 196, win 64351,
length 0
M....vU.rl~P.._..........
^C09:31:31.349560 IP 192.168.0.2.46393 >
b.resolvers.Level3.net.domain: 11148+ PTR?
1.0.168.192.in-addr.arpa. (42)
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 Captured Packets in HEX and ASCII

```
# tcpdump -XX -i eth0

11:51:18.974360 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
3509235537:3509235733, ack 3652638190, win 18760,
length 196
        0x0000:  b8ac 6f2e 57b3 0001 6c99 1468
0800 4510  ..o.W...l..h..E.
        0x0010:  00ec 8783 4000 4006 275d ac10
197e ac10  ....@.@.']...~..
        0x0020:  197d 0016 1129 d12a af51 d9b6
d5ee 5018  .}...).*.Q....P.
        0x0030:  4948 8bfa 0000 0e12 ea4d 22d1
67c0 f123  IH.......M".g..#
        0x0040:  9013 8f68 aa70 29f3 2efc c512
5660 4fe8  ...h.p).....V`O.
        0x0050:  590a d631 f939 dd06 e36a 69ed
cac2 95b6  Y..1.9...ji.....
        0x0060:  f8ba b42a 344b 8e56 a5c4 b3a2
ed82 c3a1  ...*4K.V........
        0x0070:  80c8 7980 11ac 9bd7 5b01 18d5
8180 4536  ..y.....[.....E6
        0x0080:  30fd 4f6d 4190 f66f 2e24 e877
ed23 8eb0  0.OmA..o.$.w.#..
        0x0090:  5a1d f3ec 4be4 e0fb 8553 7c85
17d9 866f  Z...K....S|....o
        0x00a0:  c279 0d9c 8f9d 445b 7b01 81eb
1b63 7f12  .y....D[{....c..
        0x00b0:  71b3 1357 52c7 cf00 95c6 c9f6
63b1 ca51  q..WR.......c..Q
        0x00c0:  0ac6 456e 0620 38e6 10cb 6139
fb2a a756  ..En..8...a9.*.V
        0x00d0:  37d6 c5f3 f5f3 d8e8 3316 d14f
d7ab fd93  7.......3..O....
        0x00e0:  1137 61c1 6a5c b4d1 ddda 380a
f782 d983  .7a.j\....8.....
        0x00f0:  62ff a5a9 bb39 4f80 668a
b....9O.f.
11:51:18.974759 IP 172.16.25.126.60952 >
mddc-01.midcorp.mid-day.com.domain: 14620+ PTR?
125.25.16.172.in-addr.arpa. (44)
        0x0000:  0014 5e67 261d 0001 6c99 1468
0800 4500  ..^g&...l..h..E.
        0x0010:  0048 5a83 4000 4011 5e25 ac10
197e ac10  .HZ.@.@.^%...~..
        0x0020:  105e ee18 0035 0034 8242 391c
0100 0001  .^...5.4.B9.....
        0x0030:  0000 0000 0000 0331 3235 0232
3502 3136  .......125.25.16
        0x0040:  0331 3732 0769 6e2d 6164 6472
0461 7270  .172.in-addr.arp
        0x0050:  6100 000c 0001
a.....
```

5. Display Available Interfaces

```
# tcpdump -D

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

6. Capture and Save Packets in a File

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

tcpdump: listening on eth0, link-type EN10MB
(Ethernet), capture size 65535 bytes
4 packets captured
4 packets received by filter
0 packets dropped by kernel
```

7. Read Captured Packets File

```
# 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: Flags [P.], seq
3353041614:3353041746, ack 4193563273, win 18760,
```

```
length 132
09:59:34.963022 IP 192.168.0.1.nokia-ann-ch1 >
192.168.0.2.ssh: Flags [.], ack 132, win 65351,
length 0
09:59:36.935309 IP 192.168.0.1.netbios-dgm >
192.168.0.255.netbios-dgm: NBT UDP PACKET(138)
09:59:37.528731 IP 192.168.0.1.nokia-ann-ch1 >
192.168.0.2.ssh: Flags [P.], seq 1:53, ack 132,
win 65351, length 5
```

8. Capture packets for a specific interface

```
# tcpdump -n -i eth0

tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
12:07:03.952358 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
3509512873:3509513069, ack 3652639034, win 18760,
length 196
12:07:03.952602 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 196, win 64171,
length 0
12:07:03.953311 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
196:504, ack 1, win 18760, length 308
12:07:03.954288 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
504:668, ack 1, win 18760, length 164
12:07:03.954502 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 668, win 65535,
length 0
12:07:03.955298 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
668:944, ack 1, win 18760, length 276
12:07:03.955425 IP 172.16.23.16.netbios-ns >
172.16.31.255.netbios-ns: NBT UDP PACKET(137):
REGISTRATION; REQUEST; BROADCAST
12:07:03.956299 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
944:1236, ack 1, win 18760, length 292
12:07:03.956535 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 1236, win 64967,
length 0
```

9. Capture only TCP Packets.

```
# tcpdump -i eth0 tcp

tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
12:10:36.216358 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
3509646029:3509646225, ack 3652640142, win 18760,
length 196
12:10:36.216592 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 196, win 64687,
length 0
12:10:36.219069 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
196:504, ack 1, win 18760, length 308
12:10:36.220039 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
504:668, ack 1, win 18760, length 164
12:10:36.220260 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 668, win 64215,
length 0
12:10:36.222045 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
668:944, ack 1, win 18760, length 276
12:10:36.223036 IP 172.16.25.126.ssh >
172.16.25.125.apwi-rxspooler: Flags [P.], seq
944:1108, ack 1, win 18760, length 164
12:10:36.223252 IP 172.16.25.125.apwi-rxspooler >
172.16.25.126.ssh: Flags [.], ack 1108, win 65535,
length 0
^C12:10:36.223461 IP mid-pay.midcorp.mid-
day.com.netbios-ssn > 172.16.22.183.recipe: Flags
[.], seq 283256512:283256513, ack 550465221, win
65531, length 1[|SMB]
```

10. Capture Packet from Specific Port

```
# tcpdump -i eth0 port 22

tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
10:37:49.056927 IP 192.168.0.2.ssh >
192.168.0.1.nokia-ann-ch1: Flags [P.], seq
3364204694:3364204890, ack 4193655445, win 20904,
length 196
10:37:49.196436 IP 192.168.0.2.ssh >
192.168.0.1.nokia-ann-ch1: Flags [P.], seq
4294967244:196, ack 1, win 20904, length 248
10:37:49.196615 IP 192.168.0.1.nokia-ann-ch1 >
192.168.0.2.ssh: Flags [.], ack 196, win 64491,
length 0
10:37:49.379298 IP 192.168.0.2.ssh >
192.168.0.1.nokia-ann-ch1: Flags [P.], seq
196:616, ack 1, win 20904, length 420
10:37:49.381080 IP 192.168.0.2.ssh >
192.168.0.1.nokia-ann-ch1: Flags [P.], seq
616:780, ack 1, win 20904, length 164
10:37:49.381322 IP 192.168.0.1.nokia-ann-ch1 >
192.168.0.2.ssh: Flags [.], ack 780, win 65535,
length 0
```

11. Capture Packets from source IP

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

```
tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
10:49:15.746474 IP 192.168.0.2.ssh >
192.168.0.1.nokia-ann-ch1: Flags [P.], seq
3364578842:3364579038, ack 4193668445, win 20904,
length 196
10:49:15.748554 IP 192.168.0.2.56200 >
b.resolvers.Level3.net.domain: 11289+ PTR?
1.0.168.192.in-addr.arpa. (42)
10:49:15.912165 IP 192.168.0.2.56234 >
b.resolvers.Level3.net.domain: 53106+ PTR?
2.0.168.192.in-addr.arpa. (42)
10:49:16.074720 IP 192.168.0.2.33961 >
b.resolvers.Level3.net.domain: 38447+ PTR?
2.2.2.4.in-addr.arpa. (38)
```

12. Capture Packets from destination IP

```
# tcpdump -i eth0 dst 50.116.66.139

tcpdump: verbose output suppressed, use -v or -vv
for full protocol decode
listening on eth0, link-type EN10MB (Ethernet),
capture size 65535 bytes
10:55:01.798591 IP 192.168.0.2.59896 >
50.116.66.139.http: Flags [.], ack 2480401451, win
318, options [nop,nop,TS val 7955710 ecr
804759402], length 0
10:55:05.527476 IP 192.168.0.2.59894 >
50.116.66.139.http: Flags [F.], seq 2521556029,
ack 2164168606, win 245, options [nop,nop,TS val
7959439 ecr 804759284], length 0
10:55:05.626027 IP 192.168.0.2.59894 >
50.116.66.139.http: Flags [.], ack 2, win 245,
options [nop,nop,TS val 7959537 ecr 804759787],
length 0
```

Sanfoundry Global Education & Learning Series – 1000 Linux Tutorials.