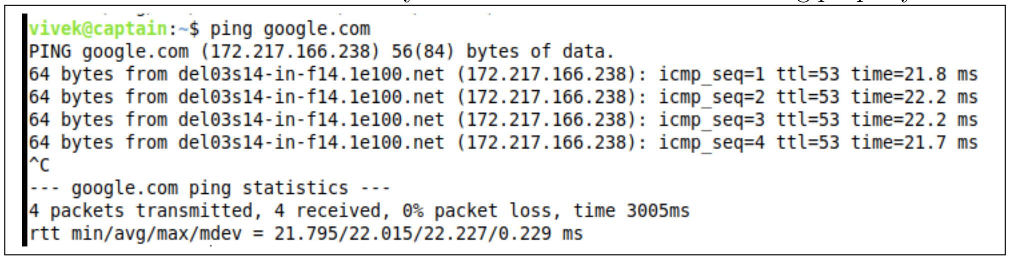
Introduction to Networking with *LINUX* *PART* – 1

1 INTRODUCTION

When it comes to networking, there is probably nothing that cannot be done with Linux.
Some one said once : Programming is learnt better with examples rather than with theories and books. Here are commands explained with example. But don't rely only on this article. This is only the insight into what can be done. See manual pages for more information.

2 ping

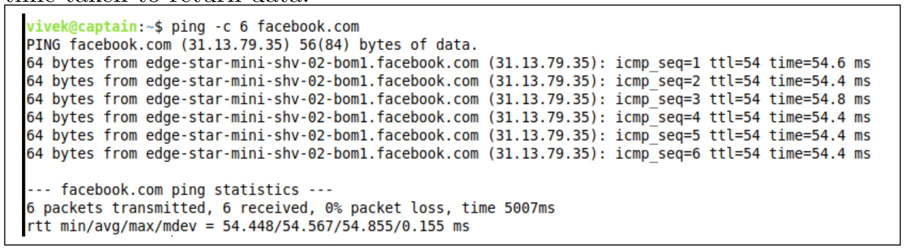
send ICMP ECHO REQUEST to network hosts
ping is the most basic command. And it is by default installed in all linux systems. This can be used to check if your internet connection is working properly.



```
vivek@captain:~$ ping google.com
PING google.com (172.217.166.238) 56(84) bytes of data.
64 bytes from del03s14-in-f14.1e100.net (172.217.166.238): icmp_seq=1 ttl=53 time=21.8 ms
64 bytes from del03s14-in-f14.1e100.net (172.217.166.238): icmp_seq=2 ttl=53 time=22.2 ms
64 bytes from del03s14-in-f14.1e100.net (172.217.166.238): icmp_seq=3 ttl=53 time=22.2 ms
64 bytes from del03s14-in-f14.1e100.net (172.217.166.238): icmp_seq=4 ttl=53 time=21.7 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 21.795/22.015/22.227/0.229 ms
```

Figure 2.1

It verifies the connection by sending packets of data. Use ctrl-C to stop it otherwise it will keep sending data. In the end it provides a brief summary of what has been done. It sends 1 packet each second by default. here you can see 172.217.166.238 is the exact IP address of server. It also provides the length of time taken to return data.

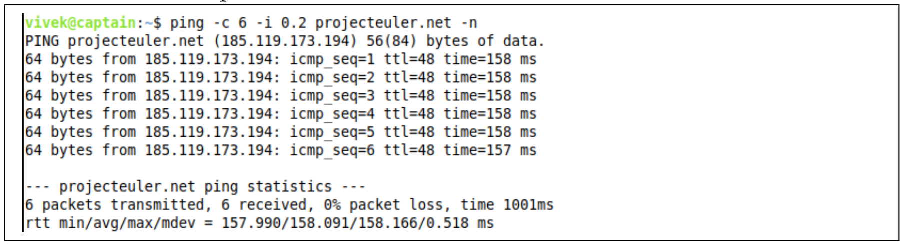


```
vivek@captain:~$ ping -c 6 facebook.com
PING facebook.com (31.13.79.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-02-bom1.facebook.com (31.13.79.35): icmp_seq=1 ttl=54 time=54.6 ms
64 bytes from edge-star-mini-shv-02-bom1.facebook.com (31.13.79.35): icmp_seq=2 ttl=54 time=54.4 ms
64 bytes from edge-star-mini-shv-02-bom1.facebook.com (31.13.79.35): icmp_seq=3 ttl=54 time=54.8 ms
64 bytes from edge-star-mini-shv-02-bom1.facebook.com (31.13.79.35): icmp_seq=4 ttl=54 time=54.4 ms
64 bytes from edge-star-mini-shv-02-bom1.facebook.com (31.13.79.35): icmp_seq=5 ttl=54 time=54.4 ms
64 bytes from edge-star-mini-shv-02-bom1.facebook.com (31.13.79.35): icmp_seq=6 ttl=54 time=54.4 ms

--- facebook.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 54.448/54.567/54.855/0.155 ms
```

Figure 2.2

Flag $-c$ `n` specifies exactly n number of packets to send. Afterwards the command auto stops.



```
vivek@captain:~$ ping -c 6 -i 0.2 projecteuler.net -n
PING projecteuler.net (185.119.173.194) 56(84) bytes of data.
64 bytes from 185.119.173.194: icmp_seq=1 ttl=48 time=158 ms
64 bytes from 185.119.173.194: icmp_seq=2 ttl=48 time=158 ms
64 bytes from 185.119.173.194: icmp_seq=3 ttl=48 time=158 ms
64 bytes from 185.119.173.194: icmp_seq=4 ttl=48 time=158 ms
64 bytes from 185.119.173.194: icmp_seq=5 ttl=48 time=158 ms
64 bytes from 185.119.173.194: icmp_seq=6 ttl=48 time=157 ms

--- projecteuler.net ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 157.990/158.091/158.166/0.518 ms
```

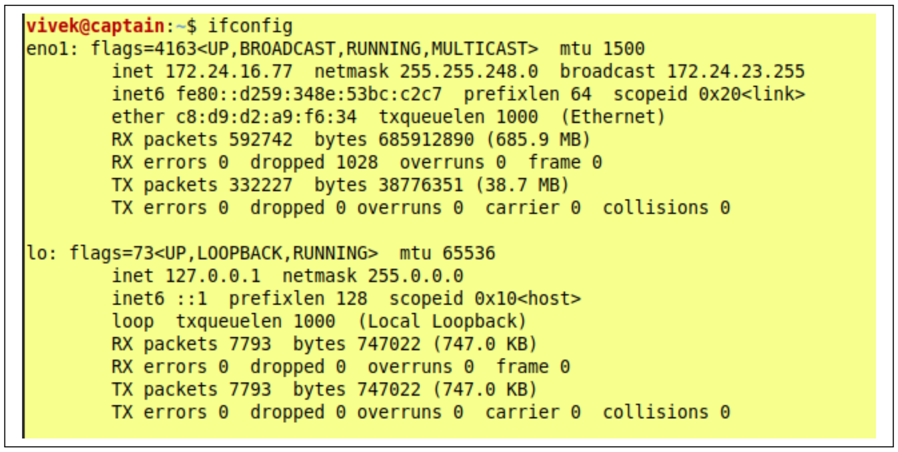
Figure 2.3

Flag $-i$ `n` specifies exactly n seconds wait before sending next packet. And Flag $-n$ says to print only numeric outputs. As manual says: only superuser can set interval less than 0.2 .
There even more options, like you can set a deadline time for ping to stop with $-w$ `deadline`. Default packet size is 55 but you can change it with $-s$ `packetsize`.

NOTE :- Some network devices are configured to ignore these packets for security reasons. Even some firewalls are configured to block ICMP traffic. example site : w3schools.com .
"ping w3schools.com" will fail to recieve any packets .

3 ifconfig

configure a network interface
Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

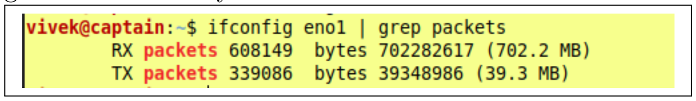


```
vivek@captain:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.24.16.77  netmask 255.255.248.0  broadcast 172.24.23.255
        inet6 fe80::d259:348e:53bc:c2c7  prefixlen 64  scopeid 0x20<link>
        ether c8:d9:d2:a9:f6:34  txqueuelen 1000  (Ethernet)
        RX packets 592742  bytes 685912890 (685.9 MB)
        RX errors 0  dropped 1028  overruns 0  frame 0
        TX packets 332227  bytes 38776351 (38.7 MB)
        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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 7793  bytes 747022 (747.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7793  bytes 747022 (747.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3.1

If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only.



```
vivek@captain:~$ ifconfig eno1 | grep packets
        RX packets 608149  bytes 702282617 (702.2 MB)
        TX packets 339086  bytes 39348986 (39.3 MB)
```

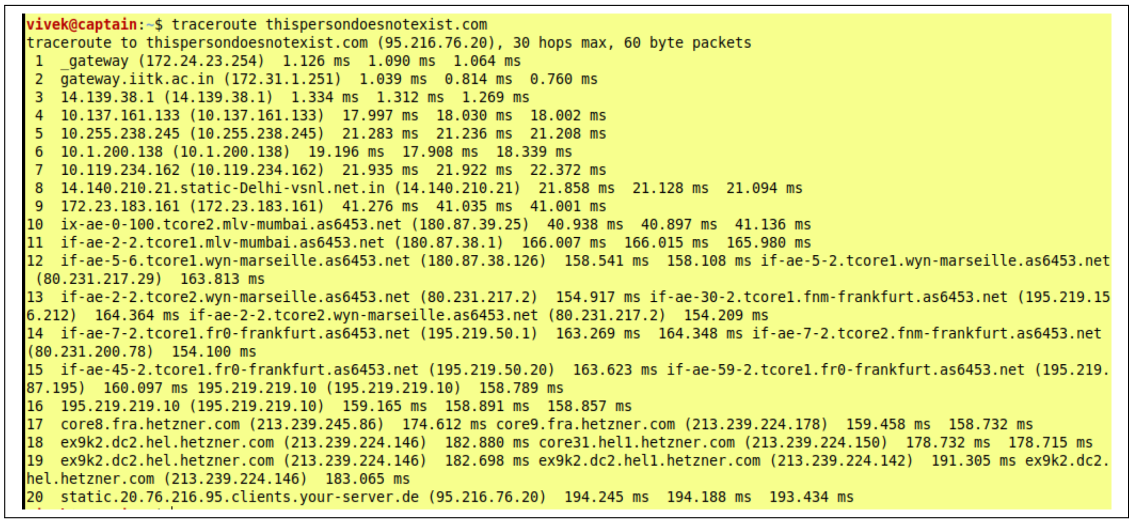
Figure 3.2

Using grep to see a specific result. Here I show data uses by the Ethernet Interface. RX is recieved amount of data and TX is transfered data since last login.

4 traceroute

print the route packets trace to network host

Trace every server that request jumps from or to in order to get to the host server.



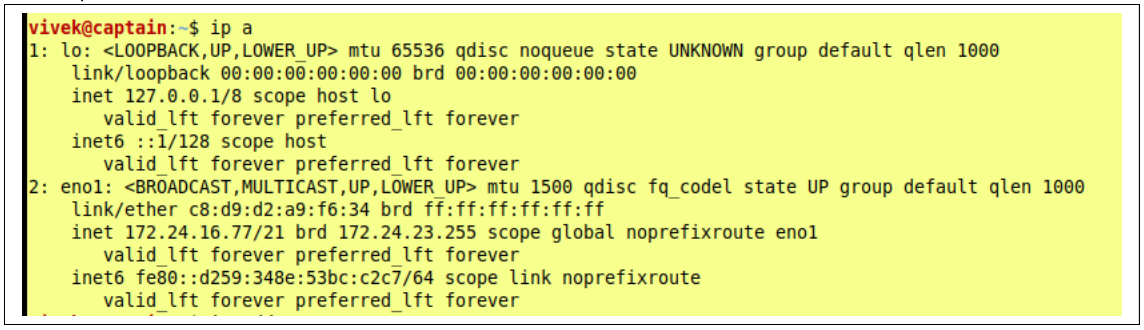
```
vivek@captain:~$ traceroute thispersondoesnotexist.com
traceroute to thispersondoesnotexist.com (95.216.76.20), 30 hops max, 60 byte packets
 1  _gateway (172.24.23.254)  1.126 ms  1.090 ms  1.064 ms
 2  gateway.iitk.ac.in (172.31.1.251)  1.039 ms  0.814 ms  0.760 ms
 3  14.139.38.1 (14.139.38.1)  1.334 ms  1.312 ms  1.269 ms
 4  10.137.161.133 (10.137.161.133)  17.997 ms  18.030 ms  18.002 ms
 5  10.255.238.245 (10.255.238.245)  21.283 ms  21.236 ms  21.208 ms
 6  10.1.200.138 (10.1.200.138)  19.196 ms  17.908 ms  18.339 ms
 7  10.119.234.162 (10.119.234.162)  21.935 ms  21.922 ms  22.372 ms
 8  14.140.210.21.static-Delhi-vsnl.net.in (14.140.210.21)  21.858 ms  21.128 ms  21.094 ms
 9  172.23.183.161 (172.23.183.161)  41.276 ms  41.035 ms  41.001 ms
10  ix-ae-0-100.tcore2.mlv-mumbai.as6453.net (180.87.39.25)  40.938 ms  40.897 ms  41.136 ms
11  if-ae-2-2.tcore1.mlv-mumbai.as6453.net (180.87.38.1)  166.007 ms  166.015 ms  165.980 ms
12  if-ae-5-6.tcore1.wyn-marseille.as6453.net (180.87.38.126)  158.541 ms  158.108 ms if-ae-5-2.tcore1.wyn-marseille.as6453.net
 (80.231.217.29)  163.813 ms
13  if-ae-2-2.tcore2.wyn-marseille.as6453.net (80.231.217.2)  154.917 ms if-ae-30-2.tcore1.fnm-frankfurt.as6453.net (195.219.15
6.212)  164.364 ms if-ae-2-2.tcore2.wyn-marseille.as6453.net (80.231.217.2)  154.209 ms
14  if-ae-7-2.tcore1.fr0-frankfurt.as6453.net (195.219.50.1)  163.269 ms  164.348 ms if-ae-7-2.tcore2.fnm-frankfurt.as6453.net
(80.231.200.78)  154.100 ms
15  if-ae-45-2.tcore1.fr0-frankfurt.as6453.net (195.219.50.20)  163.623 ms if-ae-59-2.tcore1.fr0-frankfurt.as6453.net (195.219.
87.195)  160.097 ms 195.219.219.10 (195.219.219.10)  158.789 ms
16  195.219.219.10 (195.219.219.10)  159.165 ms  158.891 ms  158.857 ms
17  core8.fra.hetzner.com (213.239.245.86)  174.612 ms core9.fra.hetzner.com (213.239.224.178)  159.458 ms  158.732 ms
18  ex9k2.dc2.hel.hetzner.com (213.239.224.146)  182.880 ms core31.hel1.hetzner.com (213.239.224.150)  178.732 ms  178.715 ms
19  ex9k2.dc2.hel.hetzner.com (213.239.224.146)  182.698 ms ex9k2.dc2.hel1.hetzner.com (213.239.224.142)  191.305 ms ex9k2.dc2.
hel.hetzner.com (213.239.224.146)  183.065 ms
20  static.20.76.216.95.clients.your-server.de (95.216.76.20)  194.245 ms  194.188 ms  193.434 ms
```

Figure 4.1

We see asterisks in the line when router do not provide identifying information. In cases where routing information is blocked, we can sometimes overcome this by adding either the -T or -I option to the traceroute command.

5 ip

show / manipulate routing, network devices, interfaces and tunnels



```
vivek@captain:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether c8:d9:d2:a9:f6:34 brd ff:ff:ff:ff:ff:ff
    inet 172.24.16.77/21 brd 172.24.23.255 scope global noprefixroute eno1
       valid_lft forever preferred_lft forever
    inet6 fe80::d259:348e:53bc:c2c7/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

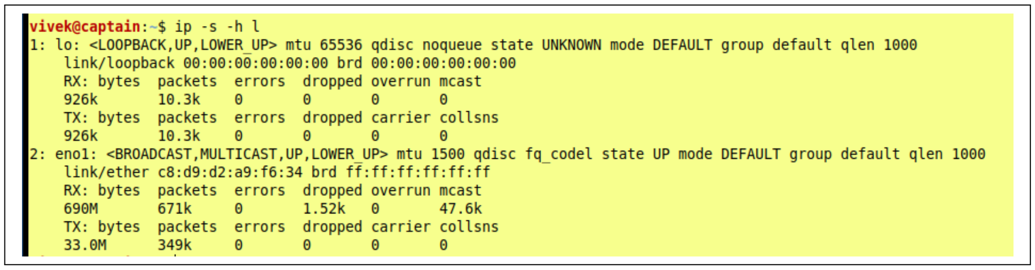
Figure 5.1

'a' is sort for address. In my case we see only 2 interfaces. The first called *lo* is the loopback interface, using this our system talks to itself. Second one *eno*1 is the ethernet interface. I don't have a wireless driver right now, but if you do , you will see another wireless interface named *wlo*0 or *wlo*1. All these listed commands will do the same thing.

```
vivek@captain:~$ ip a
```

```
vivek@captain:~$ ip addr
```

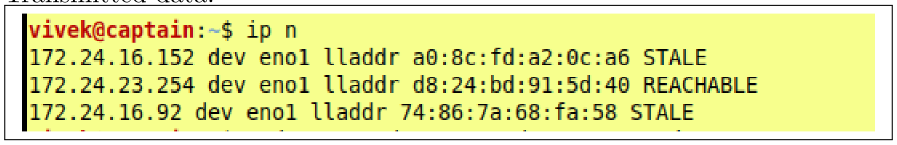
```
vivek@captain:~$ ip address
```



```
vivek@captain:~$ ip -s -h l
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    RX: bytes  packets  errors  dropped overrun mcast
    926k       10.3k    0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    926k       10.3k    0       0       0       0
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether c8:d9:d2:a9:f6:34 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    690M       671k     0       1.52k   0       47.6k
    TX: bytes  packets  errors  dropped carrier collsns
    33.0M      349k     0       0       0       0
```

Figure 5.2

$-s$ flag outputs more information, which is statistics or time value. $-h$ flag makes it more human readable. In output RX is recieved data and TX is Transmitted data.



```
vivek@captain:~$ ip n
172.24.16.152 dev eno1 lladdr a0:8c:fd:a2:0c:a6 STALE
172.24.23.254 dev eno1 lladdr d8:24:bd:91:5d:40 REACHABLE
172.24.16.92 dev eno1 lladdr 74:86:7a:68:fa:58 STALE
```

Figure 5.3

Address Resolution Protocol (ARP) is used to translate an IP address to its corresponding physical address, commonly known as MAC address. With ip command you can view the MAC address of the devices connected in your LAN by using the option neigh or neighbour or simply n.

You can activate or deactivate a network interface by:

```
$ sudo ip link set <interface> up
```

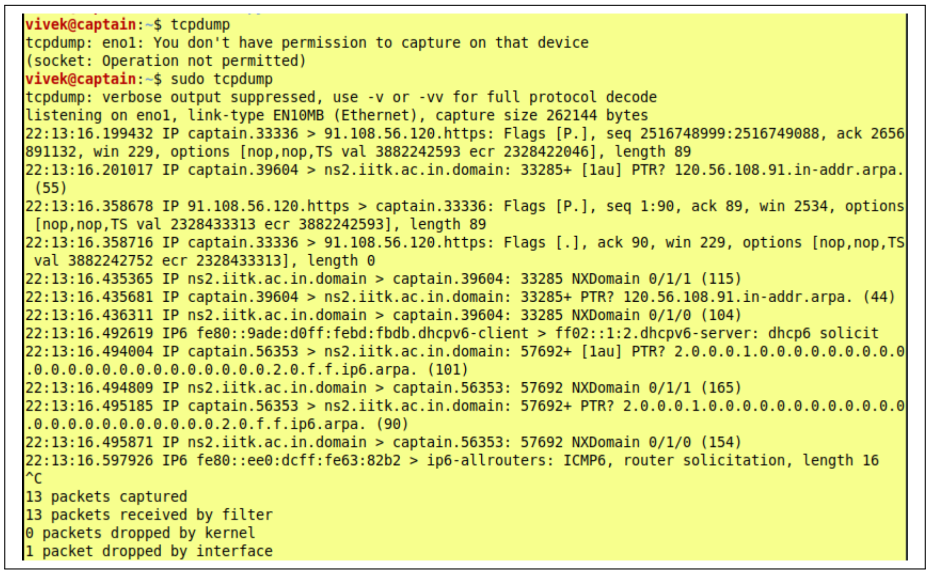
or

```
$ sudo ip link set <interface> down
```

6 tcpdump

dump traffic on a network

tcpdump is one of my favourite commands here. This program sees everything going out or coming to your system. tcp is transmission control protocol.



```
vivek@captain:~$ tcpdump
tcpdump: eno1: You don't have permission to capture on that device
(socket: Operation not permitted)
vivek@captain:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:13:16.199432 IP captain.33336 > 91.108.56.120.https: Flags [P.], seq 2516748999:2516749088, ack 2656
891132, win 229, options [nop,nop,TS val 3882242593 ecr 2328422046], length 89
22:13:16.201017 IP captain.39604 > ns2.iitk.ac.in.domain: 33285+ [1au] PTR? 120.56.108.91.in-addr.arpa.
 (55)
22:13:16.358678 IP 91.108.56.120.https > captain.33336: Flags [P.], seq 1:90, ack 89, win 2534, options
 [nop,nop,TS val 2328433313 ecr 3882242593], length 89
22:13:16.358716 IP captain.33336 > 91.108.56.120.https: Flags [.], ack 90, win 229, options [nop,nop,TS
 val 3882242752 ecr 2328433313], length 0
22:13:16.435365 IP ns2.iitk.ac.in.domain > captain.39604: 33285 NXDomain 0/1/1 (115)
22:13:16.435681 IP captain.39604 > ns2.iitk.ac.in.domain: 33285+ PTR? 120.56.108.91.in-addr.arpa. (44)
22:13:16.436311 IP ns2.iitk.ac.in.domain > captain.39604: 33285 NXDomain 0/1/0 (104)
22:13:16.492619 IP6 fe80::9ade:d0ff:febd:fbdb.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
22:13:16.494004 IP captain.56353 > ns2.iitk.ac.in.domain: 57692+ [1au] PTR? 2.0.0.0.1.0.0.0.0.0.0.0.0.0
.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (101)
22:13:16.494809 IP ns2.iitk.ac.in.domain > captain.56353: 57692 NXDomain 0/1/1 (165)
22:13:16.495185 IP captain.56353 > ns2.iitk.ac.in.domain: 57692+ PTR? 2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0
.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
22:13:16.495871 IP ns2.iitk.ac.in.domain > captain.56353: 57692 NXDomain 0/1/0 (154)
22:13:16.597926 IP6 fe80::ee0:dcff:fe63:82b2 > ip6-allrouters: ICMP6, router solicitation, length 16
^C
13 packets captured
13 packets received by filter
0 packets dropped by kernel
1 packet dropped by interface
```

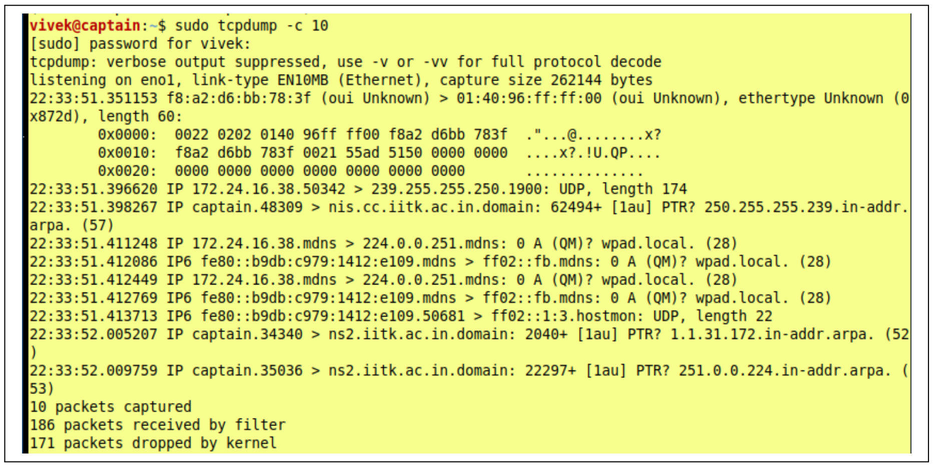
Figure 6.1

tcpdump may require sudo privilege to run, as you can see in figure 6.1 that I had to use sudo tcpdump. This command 'without any option' Listens on every port and keeps running forever. Use ctrl-C to stop. Afterwards it shows a brief summary. **'packets dropped by interface'** is a coincidence. It was capturing a packet when I stoped it. That's why it shows 1 packet dropped by interface. **'packets dropped by kernel'** is the buffer overflow. The incoming packets to tcpdump are buffered (queued) for processing. Sometimes there are too many packets in the buffer, and buffer runs out of memory. So further packets are dropped till some memory is freed.

You can increase the buffer size with the -B (–buffer-size) option like this:

```
$ tcpdump -B 4096
```

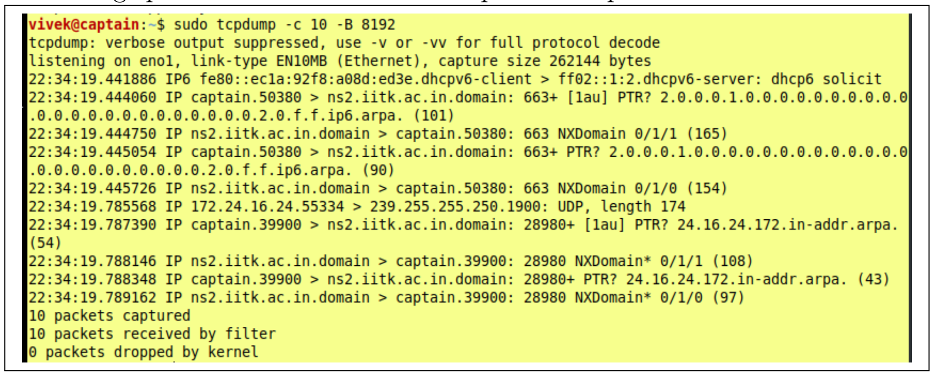
Note that the size is specified in kilobytes, so the line above sets the buffer size to 4MB.



```
vivek@captain:~$ sudo tcpdump -c 10
[sudo] password for vivek:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:33:51.351153 f8:a2:d6:bb:78:3f (oui Unknown) > 01:40:96:ff:ff:00 (oui Unknown), ethertype Unknown (0
x872d), length 60:
        0x0000:  0022 0202 0140 96ff ff00 f8a2 d6bb 783f  ."...@........x?
        0x0010:  f8a2 d6bb 783f 0021 55ad 5150 0000 0000  ....x?.!U.QP....
        0x0020:  0000 0000 0000 0000 0000 0000 0000       ..............
22:33:51.396620 IP 172.24.16.38.50342 > 239.255.255.250.1900: UDP, length 174
22:33:51.398267 IP captain.48309 > nis.cc.iitk.ac.in.domain: 62494+ [1au] PTR? 250.255.255.239.in-addr.
arpa. (57)
22:33:51.411248 IP 172.24.16.38.mdns > 224.0.0.251.mdns: 0 A (QM)? wpad.local. (28)
22:33:51.412086 IP6 fe80::b9db:c979:1412:e109.mdns > ff02::fb.mdns: 0 A (QM)? wpad.local. (28)
22:33:51.412449 IP 172.24.16.38.mdns > 224.0.0.251.mdns: 0 A (QM)? wpad.local. (28)
22:33:51.412769 IP6 fe80::b9db:c979:1412:e109.mdns > ff02::fb.mdns: 0 A (QM)? wpad.local. (28)
22:33:51.413713 IP6 fe80::b9db:c979:1412:e109.50681 > ff02::1:3.hostmon: UDP, length 22
22:33:52.005207 IP captain.34340 > ns2.iitk.ac.in.domain: 2040+ [1au] PTR? 1.1.31.172.in-addr.arpa. (52
)
22:33:52.009759 IP captain.35036 > ns2.iitk.ac.in.domain: 22297+ [1au] PTR? 251.0.0.224.in-addr.arpa. (
53)
10 packets captured
186 packets received by filter
171 packets dropped by kernel
```

Figure 6.2

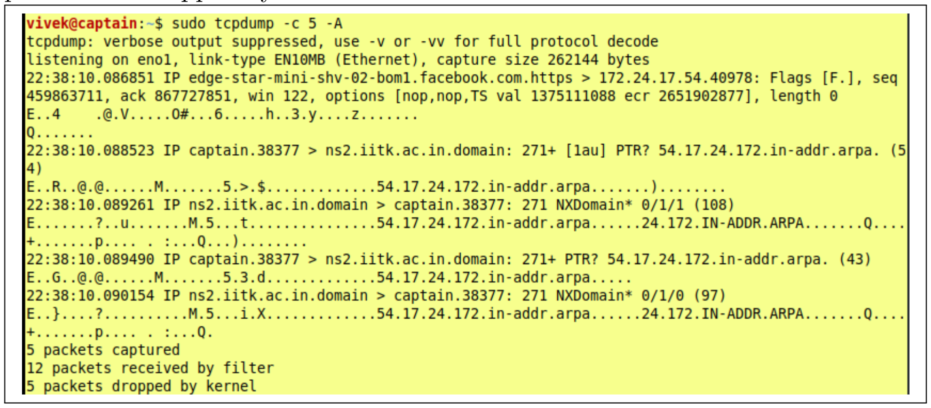
$-c$ flag specifies the exact number of packets to process.



```
vivek@captain:~$ sudo tcpdump -c 10 -B 8192
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:34:19.441886 IP6 fe80::ec1a:92f8:a08d:ed3e.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
22:34:19.444060 IP captain.50380 > ns2.iitk.ac.in.domain: 663+ [1au] PTR? 2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0
.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (101)
22:34:19.444750 IP ns2.iitk.ac.in.domain > captain.50380: 663 NXDomain 0/1/1 (165)
22:34:19.445054 IP captain.50380 > ns2.iitk.ac.in.domain: 663+ PTR? 2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0
.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
22:34:19.445726 IP ns2.iitk.ac.in.domain > captain.50380: 663 NXDomain 0/1/0 (154)
22:34:19.785568 IP 172.24.16.24.55334 > 239.255.255.250.1900: UDP, length 174
22:34:19.787390 IP captain.39900 > ns2.iitk.ac.in.domain: 28980+ [1au] PTR? 24.16.24.172.in-addr.arpa.
(54)
22:34:19.788146 IP ns2.iitk.ac.in.domain > captain.39900: 28980 NXDomain* 0/1/1 (108)
22:34:19.788348 IP captain.39900 > ns2.iitk.ac.in.domain: 28980+ PTR? 24.16.24.172.in-addr.arpa. (43)
22:34:19.789162 IP ns2.iitk.ac.in.domain > captain.39900: 28980 NXDomain* 0/1/0 (97)
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Figure 6.3

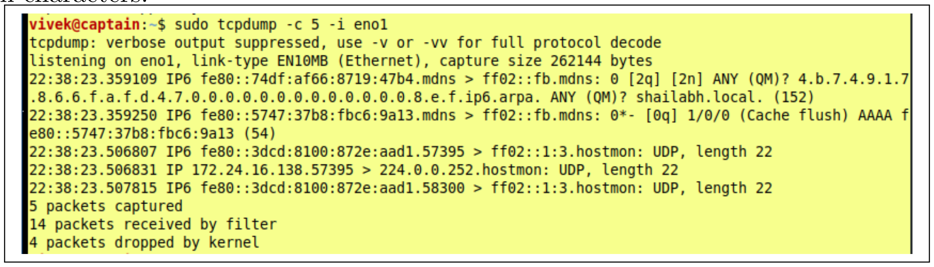
Compared to figure 6.2, We can see that after increasing the buffer size no packets are dropped by kernel.



```
vivek@captain:~$ sudo tcpdump -c 5 -A
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:38:10.086851 IP edge-star-mini-shv-02-bom1.facebook.com.https > 172.24.17.54.40978: Flags [F.], seq
459863711, ack 867727851, win 122, options [nop,nop,TS val 1375111088 ecr 2651902877], length 0
E..4    .@.V.....O#...6.....h..3.y....z.......
Q.......
22:38:10.088523 IP captain.38377 > ns2.iitk.ac.in.domain: 271+ [1au] PTR? 54.17.24.172.in-addr.arpa. (5
4)
E..R..@.@......M.......5.>.$............54.17.24.172.in-addr.arpa.......)........
22:38:10.089261 IP ns2.iitk.ac.in.domain > captain.38377: 271 NXDomain* 0/1/1 (108)
E.......?..u.......M.5...t..............54.17.24.172.in-addr.arpa......24.172.IN-ADDR.ARPA.......Q....
+.......p.... . :...Q...)........
22:38:10.089490 IP captain.38377 > ns2.iitk.ac.in.domain: 271+ PTR? 54.17.24.172.in-addr.arpa. (43)
E..G..@.@......M.......5.3.d............54.17.24.172.in-addr.arpa.....
22:38:10.090154 IP ns2.iitk.ac.in.domain > captain.38377: 271 NXDomain* 0/1/0 (97)
E..}....?..........M.5...i.X............54.17.24.172.in-addr.arpa......24.172.IN-ADDR.ARPA.......Q....
+.......p.... . :...Q.
5 packets captured
12 packets received by filter
5 packets dropped by kernel
```

Figure 6.4

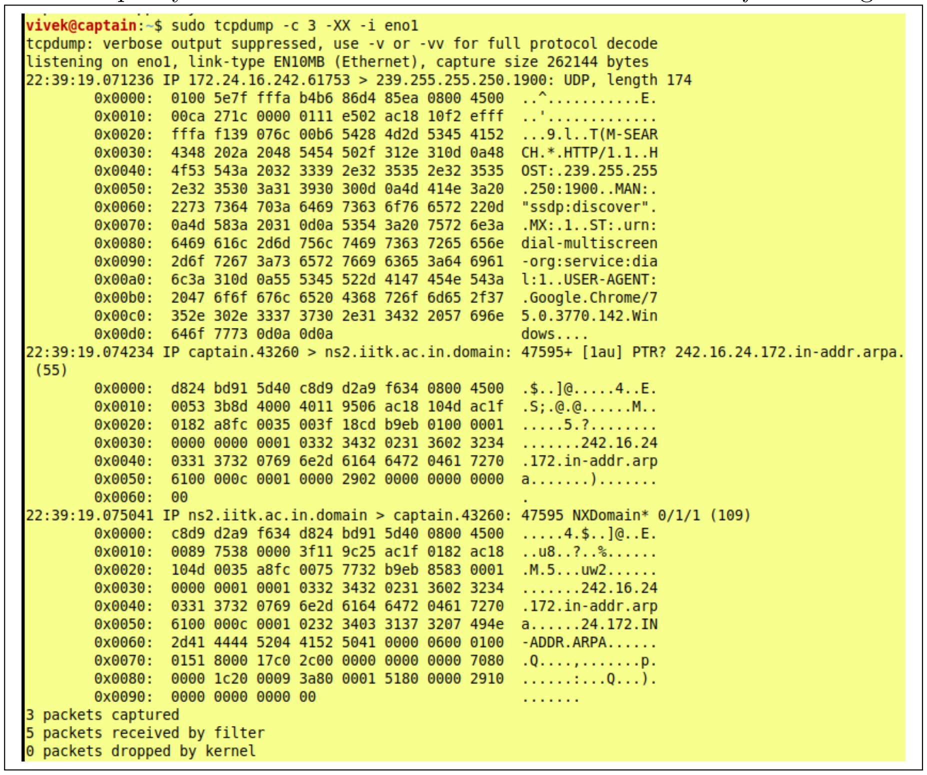
$-A$ flag tells the program to catenate the packets on standard output as ascii characters.



```
vivek@captain:~$ sudo tcpdump -c 5 -i eno1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:38:23.359109 IP6 fe80::74df:af66:8719:47b4.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? 4.b.7.4.9.1.7
.8.6.6.f.a.f.d.4.7.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. ANY (QM)? shailabh.local. (152)
22:38:23.359250 IP6 fe80::5747:37b8:fbc6:9a13.mdns > ff02::fb.mdns: 0*- [0q] 1/0/0 (Cache flush) AAAA f
e80::5747:37b8:fbc6:9a13 (54)
22:38:23.506807 IP6 fe80::3dcd:8100:872e:aad1.57395 > ff02::1:3.hostmon: UDP, length 22
22:38:23.506831 IP 172.24.16.138.57395 > 224.0.0.252.hostmon: UDP, length 22
22:38:23.507815 IP6 fe80::3dcd:8100:872e:aad1.58300 > ff02::1:3.hostmon: UDP, length 22
5 packets captured
14 packets received by filter
4 packets dropped by kernel
```

Figure 6.5

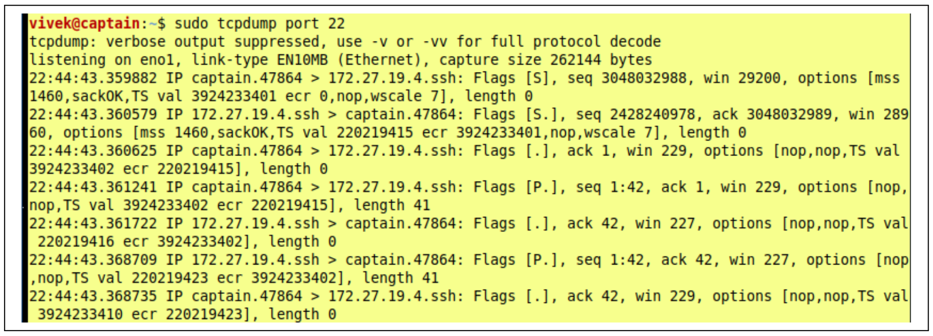
You can specify a network interface to look for with $-i < interface >$ flag.



```
vivek@captain:~$ sudo tcpdump -c 3 -XX -i eno1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:39:19.071236 IP 172.24.16.242.61753 > 239.255.255.250.1900: UDP, length 174
        0x0000:  0100 5e7f fffa b4b6 86d4 85ea 0800 4500  ..^..........E.
        0x0010:  00ca 271c 0000 0111 e502 ac18 10f2 efff  ..'............
        0x0020:  fffa f139 076c 00b6 5428 4d2d 5345 4152  ...9.l..T(M-SEAR
        0x0030:  4348 202a 2048 5454 502f 312e 310d 0a48  CH.*.HTTP/1.1..H
        0x0040:  4f53 543a 2032 3339 2e32 3535 2e32 3535  OST:.239.255.255
        0x0050:  2e32 3530 3a31 3930 300d 0a4d 414e 3a20  .250:1900..MAN:.
        0x0060:  2273 7364 703a 6469 7363 6f76 6572 220d  "ssdp:discover".
        0x0070:  0a4d 583a 2031 0d0a 5354 3a20 7572 6e3a  .MX:.1..ST:.urn:
        0x0080:  6469 616c 2d6d 756c 7469 7363 7265 656e  dial-multiscreen
        0x0090:  2d6f 7267 3a73 6572 7669 6365 3a64 6961  -org:service:dia
        0x00a0:  6c3a 310d 0a55 5345 522d 4147 454e 543a  l:1..USER-AGENT:
        0x00b0:  2047 6f6f 676c 6520 4368 726f 6d65 2f37  .Google.Chrome/7
        0x00c0:  352e 302e 3337 3730 2e31 3432 2057 696e  5.0.3770.142.Win
        0x00d0:  646f 7773 0d0a 0d0a                      dows....
22:39:19.074234 IP captain.43260 > ns2.iitk.ac.in.domain: 47595+ [1au] PTR? 242.16.24.172.in-addr.arpa.
 (55)
        0x0000:  d824 bd91 5d40 c8d9 d2a9 f634 0800 4500  .$..]@.....4..E.
        0x0010:  0053 3b8d 4000 4011 9506 ac18 104d ac1f  .S;.@.@......M..
        0x0020:  0182 a8fc 0035 003f 18cd b9eb 0100 0001  .....5.?........
        0x0030:  0000 0000 0001 0332 3432 0231 3602 3234  .......242.16.24
        0x0040:  0331 3732 0769 6e2d 6164 6472 0461 7270  .172.in-addr.arp
        0x0050:  6100 000c 0001 0000 2902 0000 0000 0000  a.......).......
        0x0060:  00                                       .
22:39:19.075041 IP ns2.iitk.ac.in.domain > captain.43260: 47595 NXDomain* 0/1/1 (109)
        0x0000:  c8d9 d2a9 f634 d824 bd91 5d40 0800 4500  .....4.$..]@..E.
        0x0010:  0089 7538 0000 3f11 9c25 ac1f 0182 ac18  ..u8..?..%......
        0x0020:  104d 0035 a8fc 0075 7732 b9eb 8583 0001  .M.5...uw2......
        0x0030:  0000 0001 0001 0332 3432 0231 3602 3234  .......242.16.24
        0x0040:  0331 3732 0769 6e2d 6164 6472 0461 7270  .172.in-addr.arp
        0x0050:  6100 000c 0001 0232 3403 3137 3207 494e  a......24.172.IN
        0x0060:  2d41 4444 5204 4152 5041 0000 0600 0100  -ADDR.ARPA......
        0x0070:  0151 8000 17c0 2c00 0000 0000 0000 7080  .Q....,.......p.
        0x0080:  0000 1c20 0009 3a80 0001 5180 0000 2910  ......:...Q...).
        0x0090:  0000 0000 0000 00                        .......
3 packets captured
5 packets received by filter
0 packets dropped by kernel
```

Figure 6.6

$-XX$ flag tells the program to catenate the packets in hex as well as in ascii on standard output.



```
vivek@captain:~$ sudo tcpdump port 22
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:44:43.359882 IP captain.47864 > 172.27.19.4.ssh: Flags [S], seq 3048032988, win 29200, options [mss
1460,sackOK,TS val 3924233401 ecr 0,nop,wscale 7], length 0
22:44:43.360579 IP 172.27.19.4.ssh > captain.47864: Flags [S.], seq 2428240978, ack 3048032989, win 289
60, options [mss 1460,sackOK,TS val 220219415 ecr 3924233401,nop,wscale 7], length 0
22:44:43.360625 IP captain.47864 > 172.27.19.4.ssh: Flags [.], ack 1, win 229, options [nop,nop,TS val
3924233402 ecr 220219415], length 0
22:44:43.361241 IP captain.47864 > 172.27.19.4.ssh: Flags [P.], seq 1:42, ack 1, win 229, options [nop,
nop,TS val 3924233402 ecr 220219415], length 41
22:44:43.361722 IP 172.27.19.4.ssh > captain.47864: Flags [.], ack 42, win 227, options [nop,nop,TS val
 220219416 ecr 3924233402], length 0
22:44:43.368709 IP 172.27.19.4.ssh > captain.47864: Flags [P.], seq 1:42, ack 42, win 227, options [nop
,nop,TS val 220219423 ecr 3924233402], length 41
22:44:43.368735 IP captain.47864 > 172.27.19.4.ssh: Flags [.], ack 42, win 229, options [nop,nop,TS val
 3924233410 ecr 220219423], length 0
```

Figure 6.7

You can also specify a port to look for with *port* < *portnumber* >. Use ctrl-C to exit. Here a look on port 22 (special port for ssh). But if you are not using this port, you will not see any output packets here.

One more thing to consider/try is that tcpdump may be spending a lot of time doing DNS queries to resolve IPs to domain names. If you don't need those, try throwing in the -n (no lookups) flag. This will speedup your work. e.g.:

```
$ tcpdump -n port 80
```

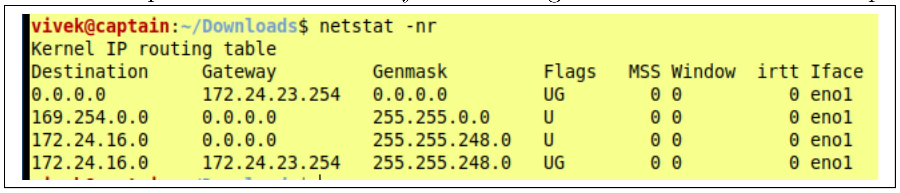
There are much more to this command than what I have shown you. Keep tinkering to find more.

These days with increasing power of processors and security, almost all programs send and recieve encrypted packets. It's not easy to decrypt a messages, and get something like a password out of it. For new people the above results of the tcpdump command has no much use. Therefore it's mainly used for troubleshooting network activities.

7 netstat

Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

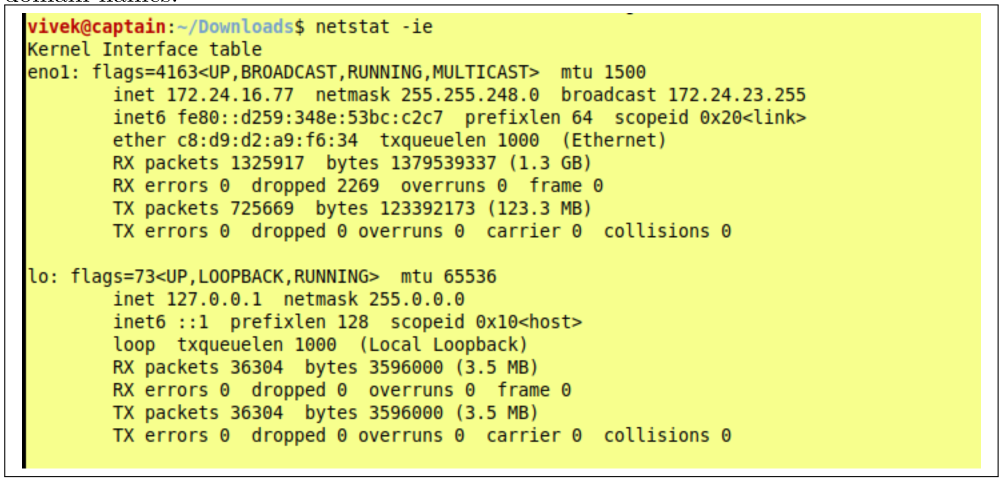
Netstat prints information about the Linux networking subsystem. The type of information printed is controlled by the first argument. Here are few examples.



```
vivek@captain:~/Downloads$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         172.24.23.254   0.0.0.0         UG        0 0          0 eno1
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eno1
172.24.16.0     0.0.0.0         255.255.248.0   U         0 0          0 eno1
172.24.16.0     172.24.23.254   255.255.248.0   UG        0 0          0 eno1
```

Figure 7.1

$-r$ flag specifies kernel IP table. $-n$ for numerical IP address instead of domain names.

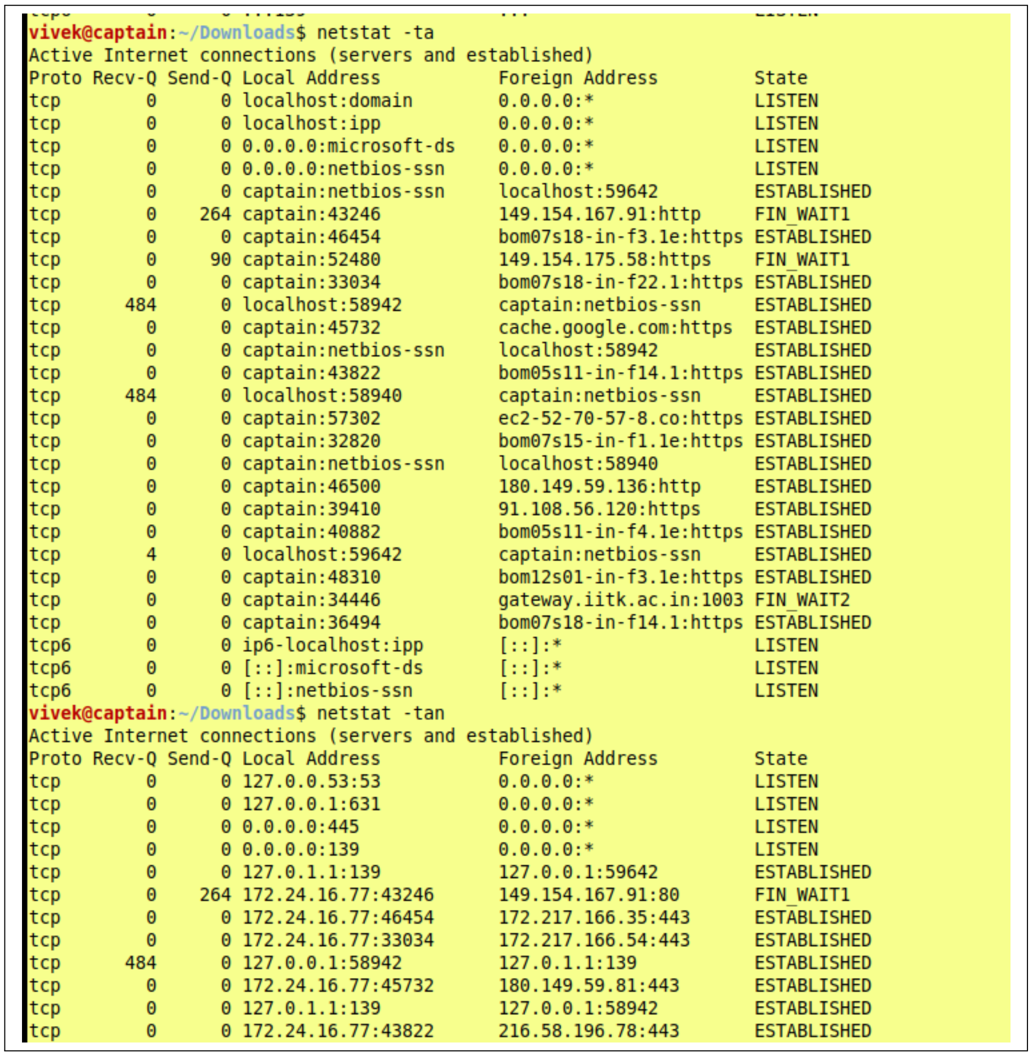


```
vivek@captain:~/Downloads$ netstat -ie
Kernel Interface table
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.24.16.77  netmask 255.255.248.0  broadcast 172.24.23.255
        inet6 fe80::d259:348e:53bc:c2c7  prefixlen 64  scopeid 0x20<link>
        ether c8:d9:d2:a9:f6:34  txqueuelen 1000  (Ethernet)
        RX packets 1325917  bytes 1379539337 (1.3 GB)
        RX errors 0  dropped 2269  overruns 0  frame 0
        TX packets 725669  bytes 123392173 (123.3 MB)
        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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 36304  bytes 3596000 (3.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 36304  bytes 3596000 (3.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 7.2

$-i$ flag gives uses of each of our device. $-e$ flag gives extended result.



```
vivek@captain:~/Downloads$ netstat -ta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:microsoft-ds    0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:netbios-ssn     0.0.0.0:*               LISTEN
tcp        0      0 captain:netbios-ssn     localhost:59642         ESTABLISHED
tcp        0    264 captain:43246           149.154.167.91:http     FIN_WAIT1
tcp        0      0 captain:46454           bom07s18-in-f3.1e:https ESTABLISHED
tcp        0     90 captain:52480           149.154.175.58:https    FIN_WAIT1
tcp        0      0 captain:33034           bom07s18-in-f22.1:https ESTABLISHED
tcp      484      0 localhost:58942         captain:netbios-ssn     ESTABLISHED
tcp        0      0 captain:45732           cache.google.com:https  ESTABLISHED
tcp        0      0 captain:netbios-ssn     localhost:58942         ESTABLISHED
tcp        0      0 captain:43822           bom05s11-in-f14.1:https ESTABLISHED
tcp      484      0 localhost:58940         captain:netbios-ssn     ESTABLISHED
tcp        0      0 captain:57302           ec2-52-70-57-8.co:https ESTABLISHED
tcp        0      0 captain:32820           bom07s15-in-f1.1e:https ESTABLISHED
tcp        0      0 captain:netbios-ssn     localhost:58940         ESTABLISHED
tcp        0      0 captain:46500           180.149.59.136:http     ESTABLISHED
tcp        0      0 captain:39410           91.108.56.120:https     ESTABLISHED
tcp        0      0 captain:40882           bom05s11-in-f4.1e:https ESTABLISHED
tcp        4      0 localhost:59642         captain:netbios-ssn     ESTABLISHED
tcp        0      0 captain:48310           bom12s01-in-f3.1e:https ESTABLISHED
tcp        0      0 captain:34446           gateway.iitk.ac.in:1003 FIN_WAIT2
tcp        0      0 captain:36494           bom07s18-in-f14.1:https ESTABLISHED
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
tcp6       0      0 [::]:microsoft-ds       [::]:*                  LISTEN
tcp6       0      0 [::]:netbios-ssn        [::]:*                  LISTEN
vivek@captain:~/Downloads$ netstat -tan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN
tcp        0      0 127.0.1.1:139           127.0.0.1:59642         ESTABLISHED
tcp        0    264 172.24.16.77:43246      149.154.167.91:80       FIN_WAIT1
tcp        0      0 172.24.16.77:46454      172.217.166.35:443      ESTABLISHED
tcp        0      0 172.24.16.77:33034      172.217.166.54:443      ESTABLISHED
tcp      484      0 127.0.0.1:58942         127.0.1.1:139           ESTABLISHED
tcp        0      0 172.24.16.77:45732      180.149.59.81:443       ESTABLISHED
tcp        0      0 127.0.1.1:139           127.0.0.1:58942         ESTABLISHED
tcp        0      0 172.24.16.77:43822      216.58.196.78:443       ESTABLISHED
```

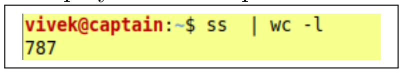
Figure 7.3

flag $-ta$ shows all ports using tcp protocol, the local address and connected foreign address.

NOTE :- This program is mostly obsolete. Replacement for netstat is ss. Replacement for netstat -r is ip route. Replacement for netstat -i is ip -s link. Replacement for netstat -g is ip maddr.

8 ss

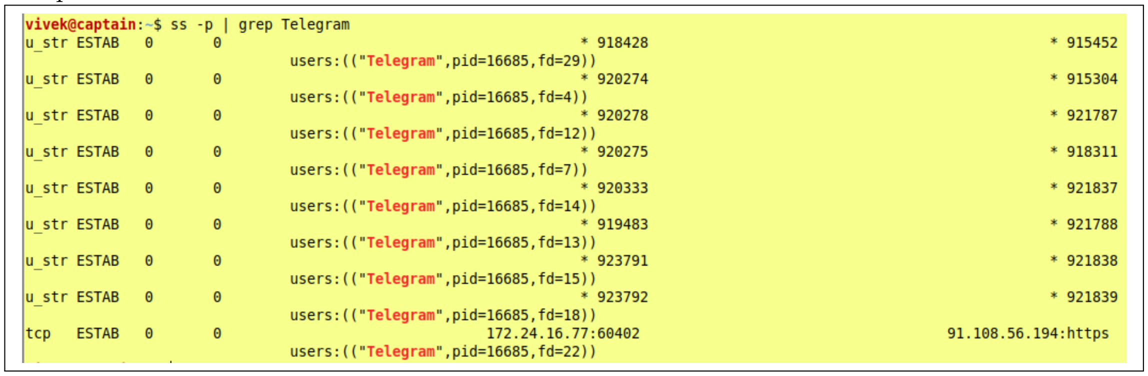
an utility to investigate sockets
It can display more TCP and state information than other tools. With no option it displays a list of open non-listening sockets.



```
vivek@captain:~$ ss  | wc -l
787
```

Figure 8.1

This result would have been nasty. *wc* $-l$ only counts the number of lines and prints it.

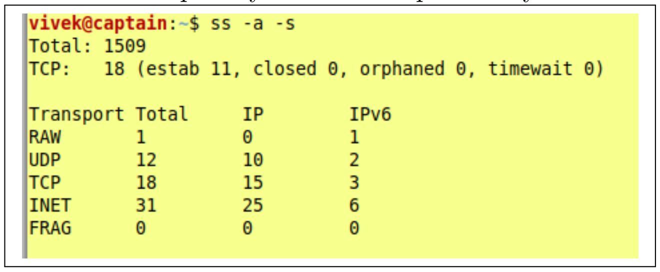


```
vivek@captain:~$ ss -p | grep Telegram
u_str ESTAB   0      0                                    * 918428                                           * 915452
                     users:(("Telegram",pid=16685,fd=29))
u_str ESTAB   0      0                                    * 920274                                           * 915304
                     users:(("Telegram",pid=16685,fd=4))
u_str ESTAB   0      0                                    * 920278                                           * 921787
                     users:(("Telegram",pid=16685,fd=12))
u_str ESTAB   0      0                                    * 920275                                           * 918311
                     users:(("Telegram",pid=16685,fd=7))
u_str ESTAB   0      0                                    * 920333                                           * 921837
                     users:(("Telegram",pid=16685,fd=14))
u_str ESTAB   0      0                                    * 919483                                           * 921788
                     users:(("Telegram",pid=16685,fd=13))
u_str ESTAB   0      0                                    * 923791                                           * 921838
                     users:(("Telegram",pid=16685,fd=15))
u_str ESTAB   0      0                                    * 923792                                           * 921839
                     users:(("Telegram",pid=16685,fd=18))
tcp   ESTAB   0      0                              172.24.16.77:60402                                 91.108.56.194:https
                     users:(("Telegram",pid=16685,fd=22))
```

Figure 8.2

With $-p$ option it shows processes using the sockets. This result was really big. I only grepped The sockets used by Telegram.

With $-K$ option you can attempt forcibly close a socket.

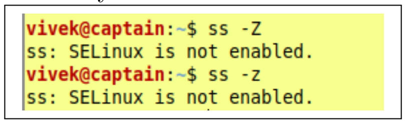


```
vivek@captain:~$ ss -a -s
Total: 1509
TCP:   18 (estab 11, closed 0, orphaned 0, timewait 0)

Transport Total     IP        IPv6
RAW       1         0         1
UDP       12        10        2
TCP       18        15        3
INET      31        25        6
FRAG      0         0         0
```

Figure 8.3

By default ss only displays listening sockets, but with $-a$ option it will display all listening and non-listening. $-s$ option is good if you just want a summary of result.



```
vivek@captain:~$ ss -Z
ss: SELinux is not enabled.
vivek@captain:~$ ss -z
ss: SELinux is not enabled.
```

Figure 8.4

$-Z$ option is same as $-p$ but it also displays process security information. $-z$ option is same as $-Z$ but also includes socket context. Oh, but here it says "SELinux is not enabled". On most systems 'selinux-utils' is not installed. It enhances the security of your system if you wish to use it as server. If not installed you can install it with command:

```
$ sudo apt install selinux-utils
```

on debian based systems. But be warned It might crush your system, so better ask someone before installing.

9 iw

show / manipulate wireless devices and their configuration

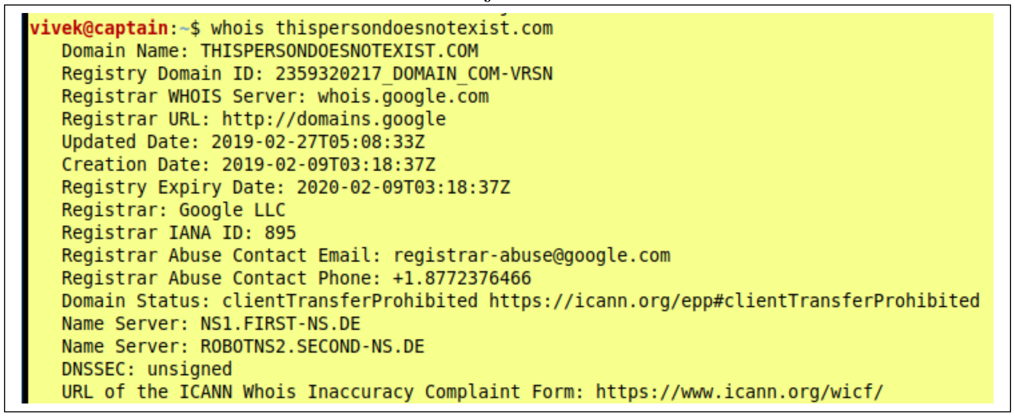
Take a look at manual page. Coming Soon from my side.

Or just type "iw" and you will see a lot of options.

10 whois

client for the whois directory service

The WHOIS is a query/response protocol that is widely used to query databases that hold information about internet resources such as domain names and IP address allocations. whois searches for an object in a RFC 3912 database.

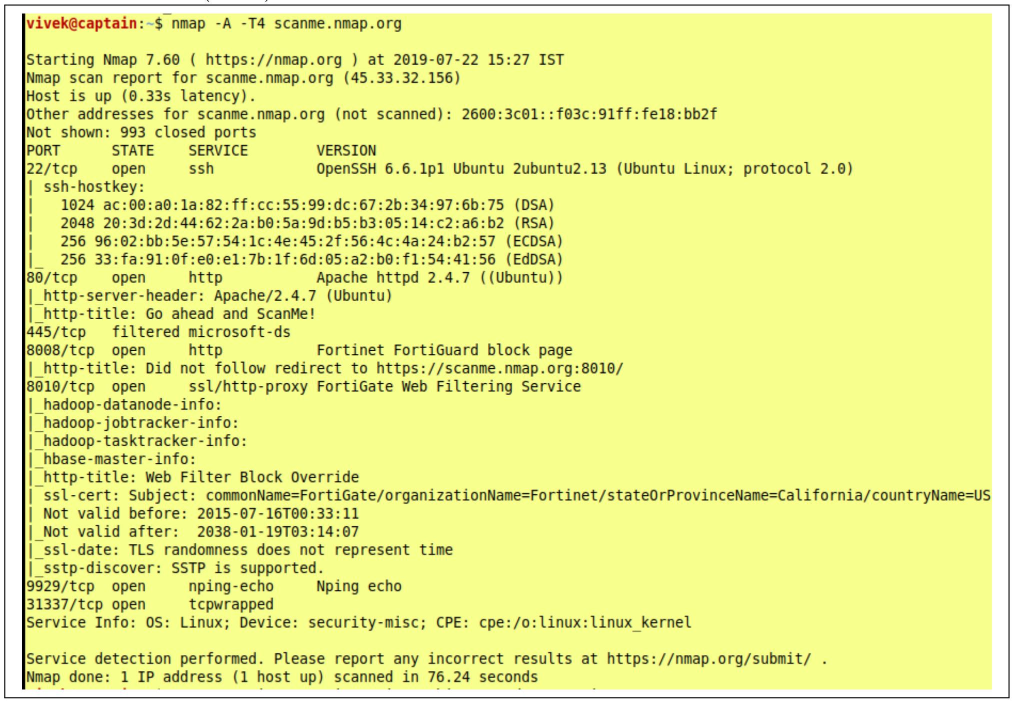


```
vivek@captain:~$ whois thispersondoesnotexist.com
   Domain Name: THISPERSONDOESNOTEXIST.COM
   Registry Domain ID: 2359320217_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.google.com
   Registrar URL: http://domains.google
   Updated Date: 2019-02-27T05:08:33Z
   Creation Date: 2019-02-09T03:18:37Z
   Registry Expiry Date: 2020-02-09T03:18:37Z
   Registrar: Google LLC
   Registrar IANA ID: 895
   Registrar Abuse Contact Email: registrar-abuse@google.com
   Registrar Abuse Contact Phone: +1.8772376466
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Name Server: NS1.FIRST-NS.DE
   Name Server: ROBOTNS2.SECOND-NS.DE
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
```

Figure 10.1

11 nmap

Network exploration tool and security / port scanner
nmap (”network mapper”) is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, and can also be used against a single host. In addition to the interesting ports table, Nmap can provide further information on targets, including reverse DNS names, operating system guesses, device types, MAC addresses application name and version. This command has been used in two of the modern movies Elysium (2013), G.I. Joe: Retaliation (2013).

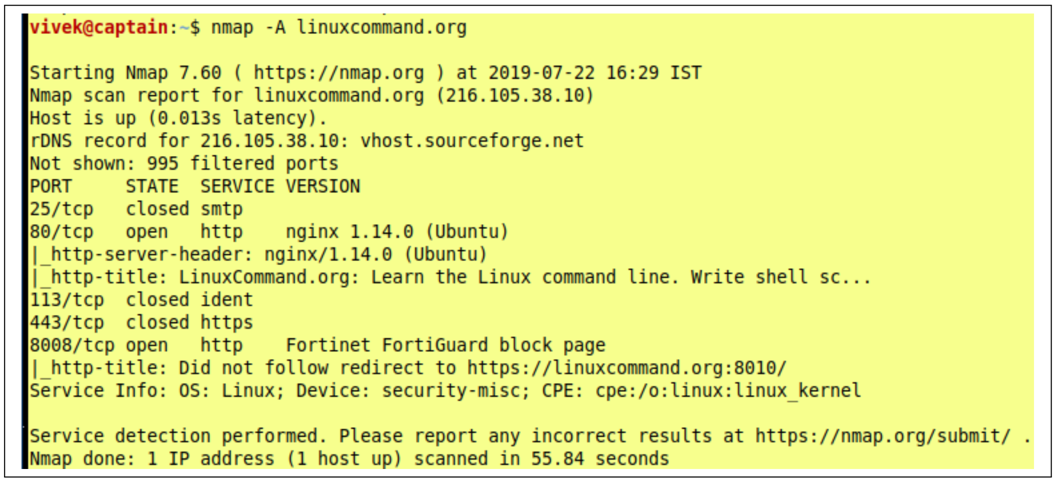


```
vivek@captain:~$ nmap -A -T4 scanme.nmap.org

Starting Nmap 7.60 ( https://nmap.org ) at 2019-07-22 15:27 IST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.33s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 993 closed ports
PORT      STATE    SERVICE        VERSION
22/tcp    open     ssh            OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (EdDSA)
80/tcp    open     http           Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Go ahead and ScanMe!
445/tcp   filtered microsoft-ds
8008/tcp  open     http           Fortinet FortiGuard block page
|_http-title: Did not follow redirect to https://scanme.nmap.org:8010/
8010/tcp  open     ssl/http-proxy FortiGate Web Filtering Service
|_hadoop-datanode-info:
|_hadoop-jobtracker-info:
|_hadoop-tasktracker-info:
|_hbase-master-info:
|_http-title: Web Filter Block Override
| ssl-cert: Subject: commonName=FortiGate/organizationName=Fortinet/stateOrProvinceName=California/countryName=US
| Not valid before: 2015-07-16T00:33:11
|_Not valid after:  2038-01-19T03:14:07
|_ssl-date: TLS randomness does not represent time
|_sstp-discover: SSTP is supported.
9929/tcp  open     nping-echo     Nping echo
31337/tcp open     tcpwrapped
Service Info: OS: Linux; Device: security-misc; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 76.24 seconds
```

Figure 11.1



```
vivek@captain:~$ nmap -A linuxcommand.org

Starting Nmap 7.60 ( https://nmap.org ) at 2019-07-22 16:29 IST
Nmap scan report for linuxcommand.org (216.105.38.10)
Host is up (0.013s latency).
rDNS record for 216.105.38.10: vhost.sourceforge.net
Not shown: 995 filtered ports
PORT     STATE  SERVICE VERSION
25/tcp   closed smtp
80/tcp   open   http    nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: LinuxCommand.org: Learn the Linux command line. Write shell sc...
113/tcp  closed ident
443/tcp  closed https
8008/tcp open   http    Fortinet FortiGuard block page
|_http-title: Did not follow redirect to https://linuxcommand.org:8010/
Service Info: OS: Linux; Device: security-misc; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 55.84 seconds
```

Figure 11.2

flag $-A$ provides more information, like os version. flag $-T4$ makes work faster.

We will discus about this command again, later in network exploitation. We will take a deep dive later.

12 conclusion

I you are Interested in networking with linux, then for you Things are just getting started. These simple commands have a supporting role in actual networking. On diving deeper You will find much more interresting things. You will be able to write your own scripts. Scripts strong enough to download full data base of a server. Some of the funny parts include hacking wifi passwords, making a server down, Creating your own cypher chatting protocol, etc.

13 further reading

linuxcommand.org
A well organised website to get started with linux. Also Bash-Scripting.
linux-training.be
Comtains a series of 6 books till now. from very fundamentals of linux to Administration, Servers, Security and Networking.
DNS
Learn about Domain Name System (DNS) from wikipedia.
DHCP

Learn about Dynamic Host Configuration Protocol (dhcp) .
TCP
Learn about one of the main protocols of the internet protocol suite Transmission Control Protocol (tcp).
URL
Learn about Uniform Resource Locator (URL).
IPv4
Learn about one of the core protocols of standards-based internetworking methods in the Internet. The Fourth version of the Internet Protocol (IP) which still routes most Internet traffic today.
ICMP
Learn about Internet Control Message Protocol (ICMP) a supporting protocol in internet protocol suite.
nmap
Oficial Website of nmap. A lot more than the manual. Also there a is a book about nmap.

---

14 The Post Credit Scene

If you liked this blog, You are definitely going to like my upcoming blogs in this series. There we will talk about how to connect to a server , how to download from or upload data to a server. We will learn to create our own server, and to maintain it. We will learn to share data between computers using Terminal. etc..etc.. A lot to come. stay tunned.

If you find anything wrong or you have any sugestions. Mail me at vivekkumargupta680@gmail.com . I would love to hear from you.
