MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

DEPARTMENT OF ICT

Assignment No : 01

Course Code : ICT-3207

Course Title : Computer Networks

Assignment Name : Every Networking tools(Linux Exercise)

Submitted by

Md. Faruk Hosen
ID : IT-17035
Session : 2016-2017
Year : 3rd Semester : 2nd

Submitted to

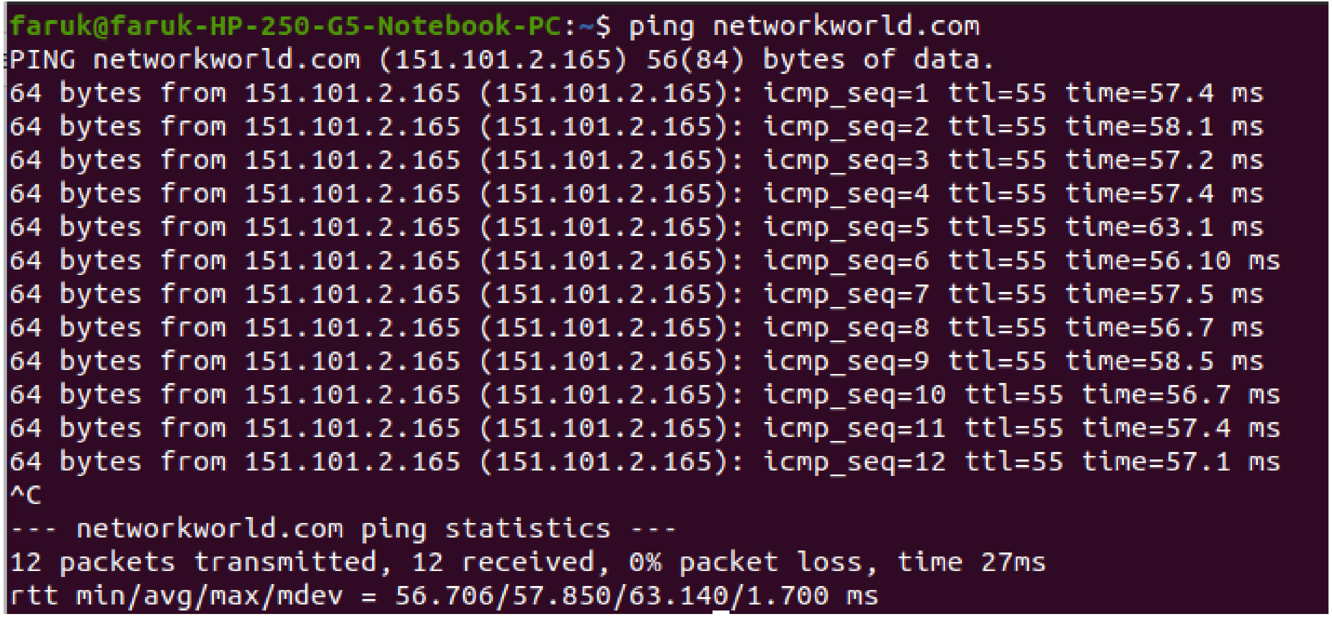
Nazrul Islam
Assistant Professor,
Department of ICT,MBSTU
Santosh,Tangail-1902

Date of Submission : 25 February 2020

Objective : **Learn about network tool:**
ping ,curl,httpie,wget,tc,dig/nslookup , whois ,ssh ,scp,rsinc, ngrip ,tcpdump,wireshark,tshark,tcpflow,ifconfig,route ,ip,arp,mitmproxy,nmap,zenmap,p0f,openvpn,wireguard,nc,socat,ftp/sftp,netstat/ss/lsof/fuser , iptables,nftables,hping3,traceroute/mtr,tcptraceroute , ethtool, iw/iwconfig , sysctl , openssl , stunnel,iptraf/nethogs/iftop/ntop , ab/nload/iperf , python -m SimpleHttpserver , ipcalc , nsenter

Every networking tool :

1. Ping : ping is sending out packets called echo requests that simply ask the remote host to respond we got several responses before we stop the requests. Ping is useful for troubleshooting because with one very simple command it tells you one that the remote system is running two your network connection is working.

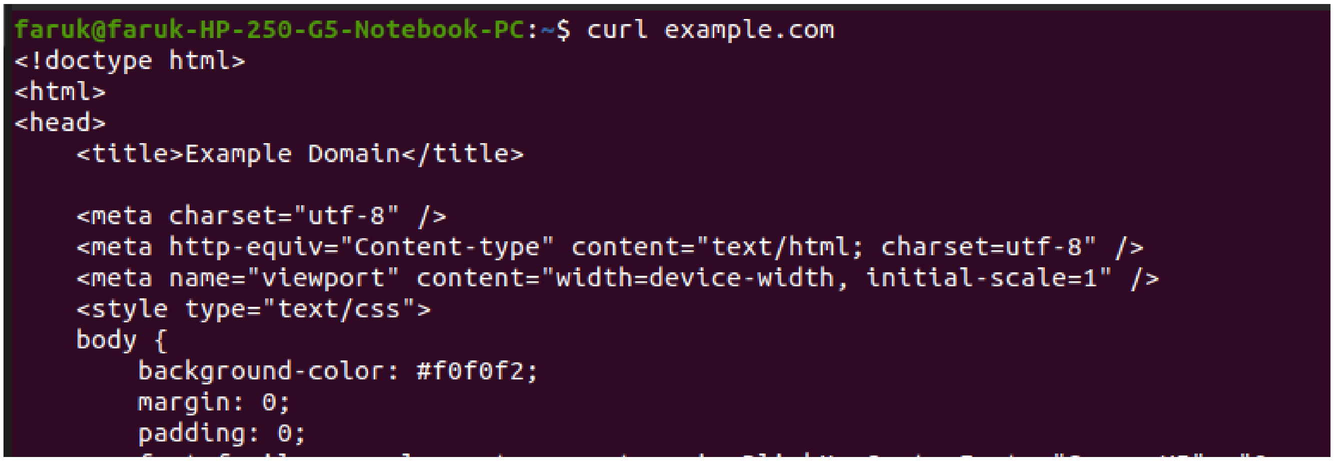


```
faruk@faruk-HP-250-G5-Notebook-PC:~$ ping networkworld.com
PING networkworld.com (151.101.2.165) 56(84) bytes of data.
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=1 ttl=55 time=57.4 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=2 ttl=55 time=58.1 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=3 ttl=55 time=57.2 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=4 ttl=55 time=57.4 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=5 ttl=55 time=63.1 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=6 ttl=55 time=56.10 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=7 ttl=55 time=57.5 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=8 ttl=55 time=56.7 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=9 ttl=55 time=58.5 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=10 ttl=55 time=56.7 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=11 ttl=55 time=57.4 ms
64 bytes from 151.101.2.165 (151.101.2.165): icmp_seq=12 ttl=55 time=57.1 ms
^C
--- networkworld.com ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 27ms
rtt min/avg/max/mdev = 56.706/57.850/63.140/1.700 ms
```

2. curl : make any HTTP request you want . *curl* is a command line tool to transfer data to or from a server, using any of the supported protocols (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP or FILE).

The most basic command we can give to `cURL` is to download a website or file.

```
curl example.com
```



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ curl example.com
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
```

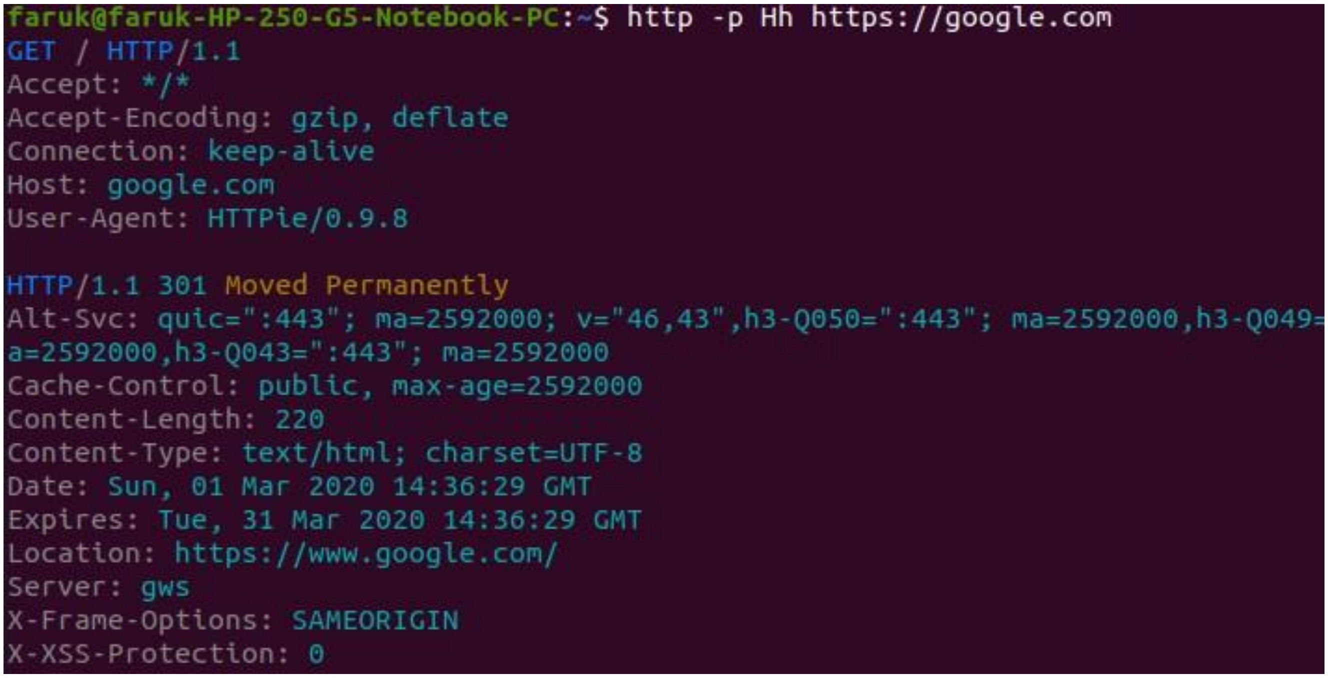
Save the Output to a File

To save the result of the `curl` command, use either the `-o` or `-O` option.

```
curl -O https://cdn.jsdelivr.net/npm/vue/dist/vue.js
```

You can resume a download by using the `-C -` option.

3.Httpie : HTTPie (pronounced aitch-tee-tee-pie) is a command line HTTP client. Its goal is to make CLI interaction with web services as human-friendly as possible. It provides a simple http command that allows for sending arbitrary HTTP requests using a simple and natural syntax, and displays colorized output.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ http -p Hh https://google.com
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: google.com
User-Agent: HTTPie/0.9.8

HTTP/1.1 301 Moved Permanently
Alt-Svc: quic=":443"; ma=2592000; v="46,43",h3-Q050=":443"; ma=2592000,h3-Q049=
a=2592000,h3-Q043=":443"; ma=2592000
Cache-Control: public, max-age=2592000
Content-Length: 220
Content-Type: text/html; charset=UTF-8
Date: Sun, 01 Mar 2020 14:36:29 GMT
Expires: Tue, 31 Mar 2020 14:36:29 GMT
Location: https://www.google.com/
Server: gws
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 0
```

4. wget : download files . Wget command is a Linux command line utility that helps us to download the files from the web. We can download the files from web servers using HTTP, HTTPS and FTP protocols. We can use wget in scripts and cronjobs.

1. To simply download a webpage:

```
wget http://example.com/sample.php
```

2. To download the file in background

```
wget -b http://www.example.com/samplepage.php
```

3. To overwrite the log wile of the wget command

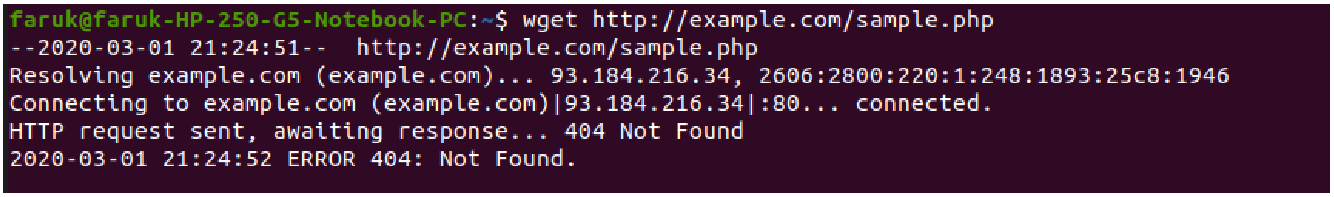
```
wget http://www.example.com/filename.txt -o /path/filename.txt
```

4. To resume a partially downloaded file

```
wget -c http://example.com/samplefile.tar.gz
```

5. To try a given number of times

```
wget --tries=10 http://example.com/samplefile.tar.gz
```



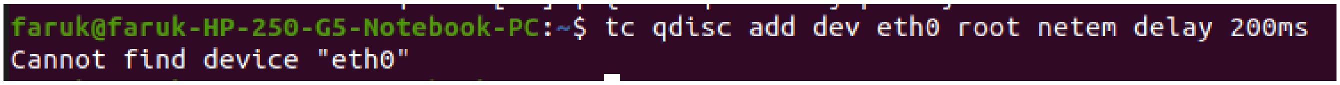
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ wget http://example.com/sample.php
--2020-03-01 21:24:51--  http://example.com/sample.php
Resolving example.com (example.com)... 93.184.216.34, 2606:2800:220:1:248:1893:25c8:1946
Connecting to example.com (example.com)|93.184.216.34|:80... connected.
HTTP request sent, awaiting response... 404 Not Found
2020-03-01 21:24:52 ERROR 404: Not Found.
```

5. tc : show/manipulate traffic control settings . **Tc** is used to configure Traffic Control in the Linux kernel.

Qdisc :

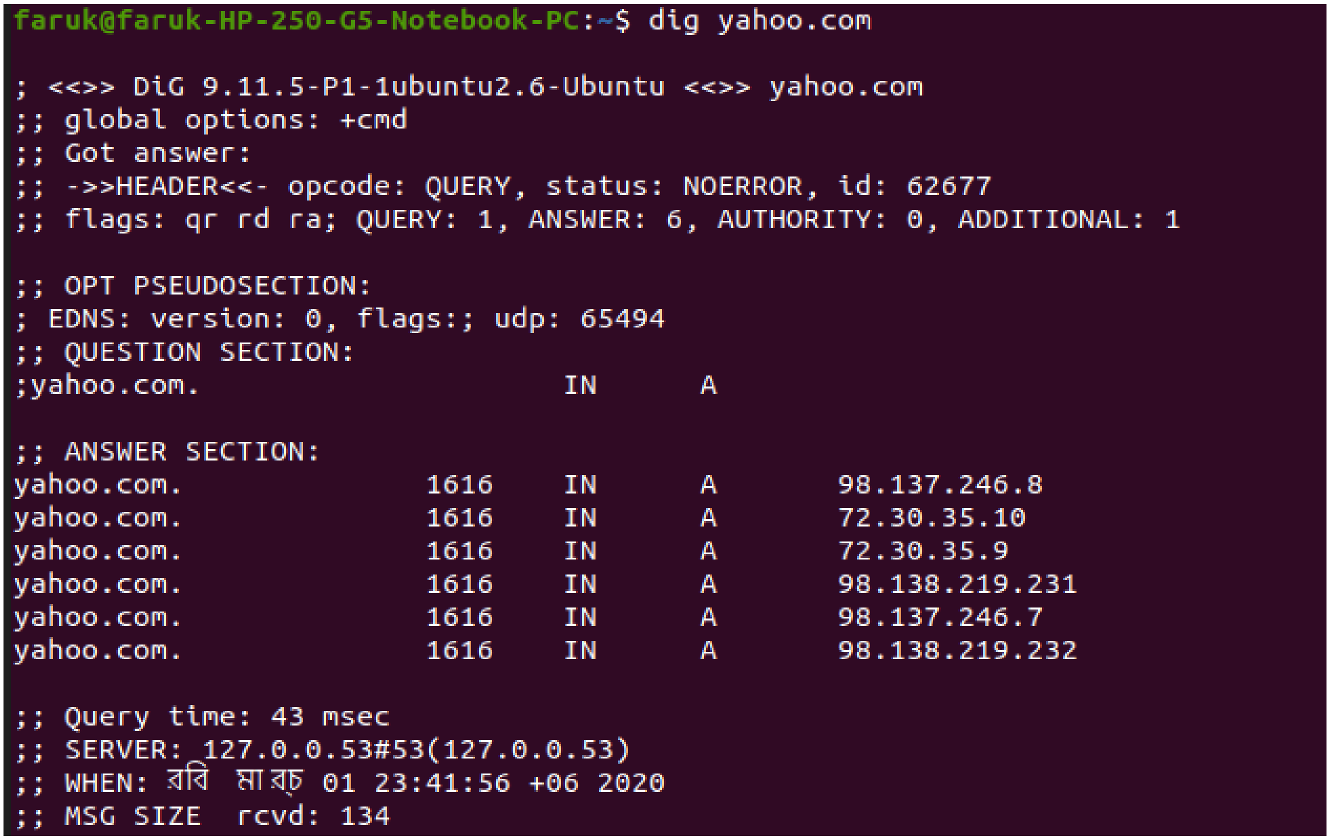
qdisc is short for 'queueing discipline' and it is elementary to understanding traffic control. Whenever the kernel needs to send a packet to an interface, it is **enqueued** to the qdisc configured for that interface. Immediately afterwards, the kernel tries to get as many packets as possible from the qdisc, for giving them to the network adaptor driver.

The first example is how to add constant delay to an interface. The syntax is as follows (run this as root):



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ tc qdisc add dev eth0 root netem delay 200ms
Cannot find device "eth0"
```

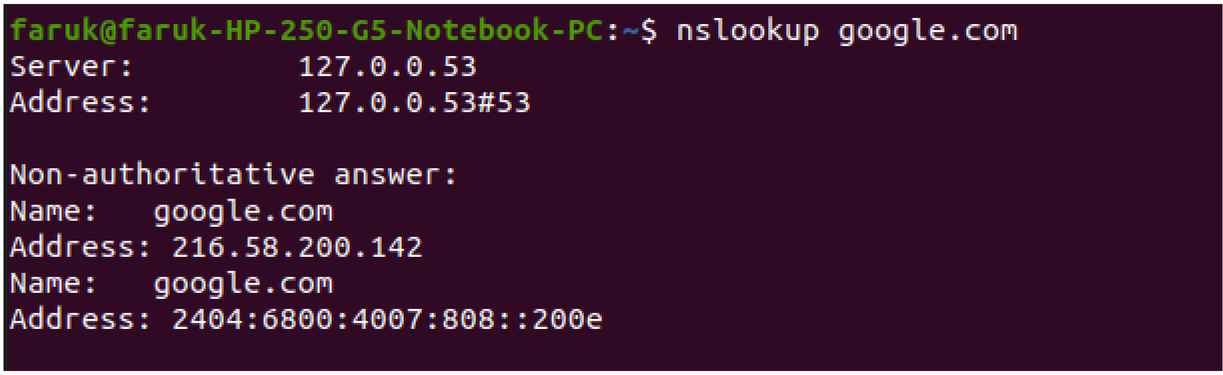
6.dig/nslookup : **Dig** stands for (**Domain Information Groper**) is a network administration command-line tool for querying **Domain Name System** (**DNS**) name servers. It is useful for verifying



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ dig yahoo.com

; <<>> DiG 9.11.5-P1-1ubuntu2.6-Ubuntu <<>> yahoo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62677
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;yahoo.com.                     IN      A

;; ANSWER SECTION:
yahoo.com.              1616    IN      A       98.137.246.8
yahoo.com.              1616    IN      A       72.30.35.10
yahoo.com.              1616    IN      A       72.30.35.9
yahoo.com.              1616    IN      A       98.138.219.231
yahoo.com.              1616    IN      A       98.137.246.7
yahoo.com.              1616    IN      A       98.138.219.232

;; Query time: 43 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: রবি মার্চ 01 23:41:56 +06 2020
;; MSG SIZE  rcvd: 134
```

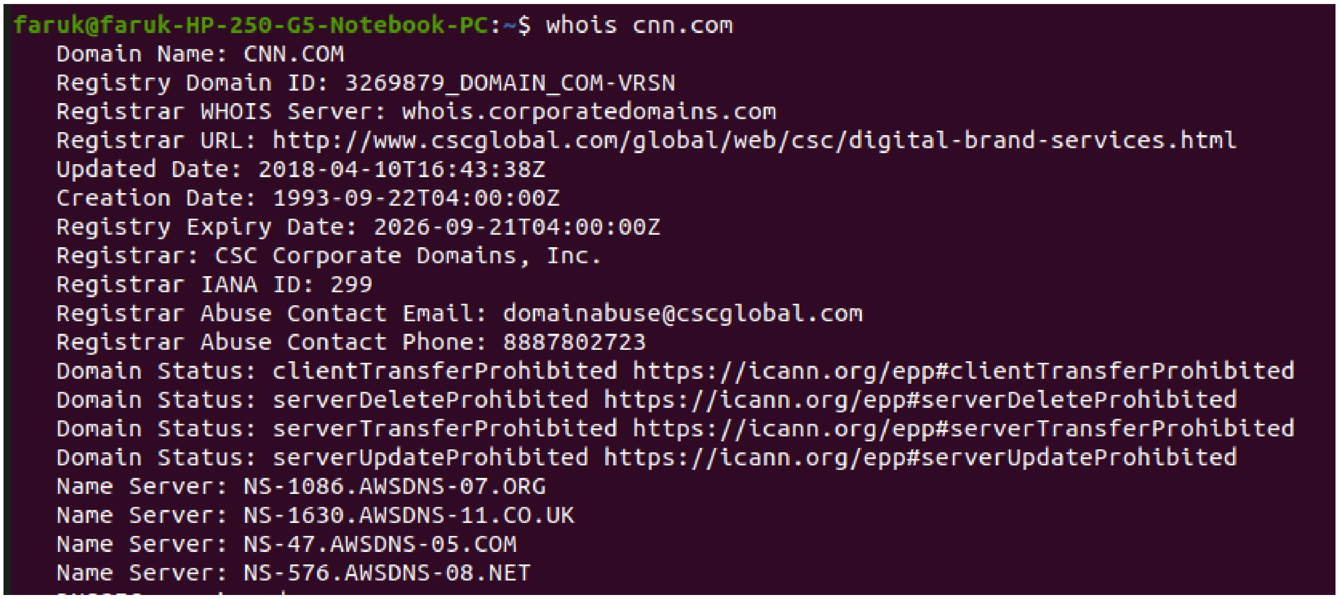
and troubleshooting **DNS** problems and also to perform **DNS** lookups and displays the answers that are returned from the name server that were queried. dig command replaces older tool such as **nslookup** and the host.

Nslookup (stands for "Name Server Lookup") is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems.



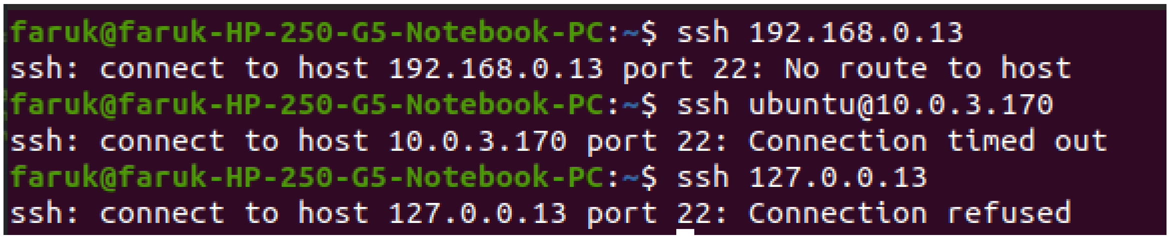
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.200.142
Name:   google.com
Address: 2404:6800:4007:808::200e
```

7.whois: You can use the *whois* command in Linux to find out information about a domain, such as the owner of the domain, the owner's contact information, and the nameservers that the domain is using. For example, to find out domain information of linux-bible.com, we can use the following command:



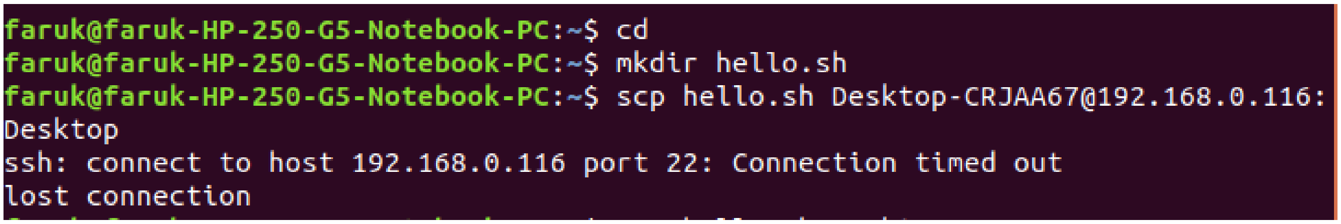
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ whois cnn.com
   Domain Name: CNN.COM
   Registry Domain ID: 3269879_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.corporatedomains.com
   Registrar URL: http://www.cscglobal.com/global/web/csc/digital-brand-services.html
   Updated Date: 2018-04-10T16:43:38Z
   Creation Date: 1993-09-22T04:00:00Z
   Registry Expiry Date: 2026-09-21T04:00:00Z
   Registrar: CSC Corporate Domains, Inc.
   Registrar IANA ID: 299
   Registrar Abuse Contact Email: domainabuse@cscglobal.com
   Registrar Abuse Contact Phone: 8887802723
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS-1086.AWSDNS-07.ORG
   Name Server: NS-1630.AWSDNS-11.CO.UK
   Name Server: NS-47.AWSDNS-05.COM
   Name Server: NS-576.AWSDNS-08.NET
```

8. ssh : *ssh* stands for “Secure Shell”. It is a protocol used to securely connect to a remote server/system. ssh is secure in the sense that it transfers the data in encrypted form between the host and the client. It transfers inputs from the client to the host and relays back the output. Ssh ankit@192.168.0.13



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ ssh 192.168.0.13
ssh: connect to host 192.168.0.13 port 22: No route to host
faruk@faruk-HP-250-G5-Notebook-PC:~$ ssh ubuntu@10.0.3.170
ssh: connect to host 10.0.3.170 port 22: Connection timed out
faruk@faruk-HP-250-G5-Notebook-PC:~$ ssh 127.0.0.13
ssh: connect to host 127.0.0.13 port 22: Connection refused
```

9. scp : stands for Secure copy. Copy files over a SSH connection.**scp** (secure copy) command in Linux system is used to copy file(s) between servers in a secure way. The SCP command or secure copy allows secure transferring of files in between the local host and the remote host or between two remote hosts.



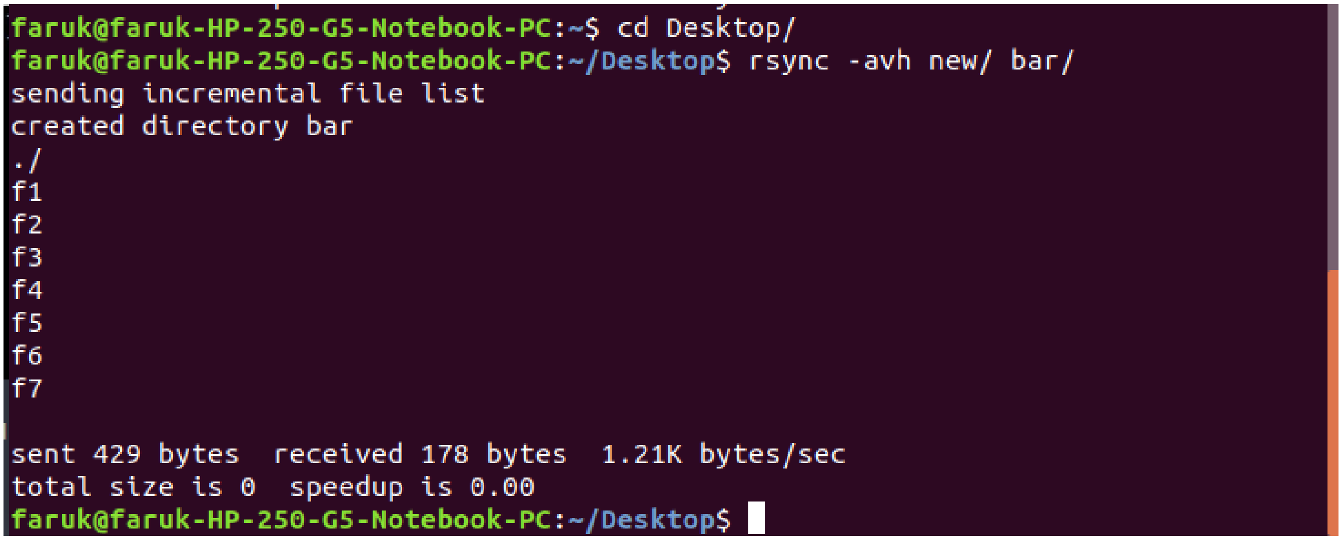
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ cd
faruk@faruk-HP-250-G5-Notebook-PC:~$ mkdir hello.sh
faruk@faruk-HP-250-G5-Notebook-PC:~$ scp hello.sh Desktop-CRJAA67@192.168.0.116:
Desktop
ssh: connect to host 192.168.0.116 port 22: Connection timed out
lost connection
```

10. rsync : synchronize file systems . Copy only changed files (works over ssh) .Rsync, short for **remote sync**, is a Linux command used for copying and synchronizing files, between 2 Linux systems. It works in such a way that any additional changes made to a file or directory are replicated on another Linux system.

With Linux rsync command, you can seamlessly copy and synchronize backups locally and remotely between Linux systems & servers. In this article, we will examine various command options that you can use alongside rsync command.

Rsync -avh new/bar/

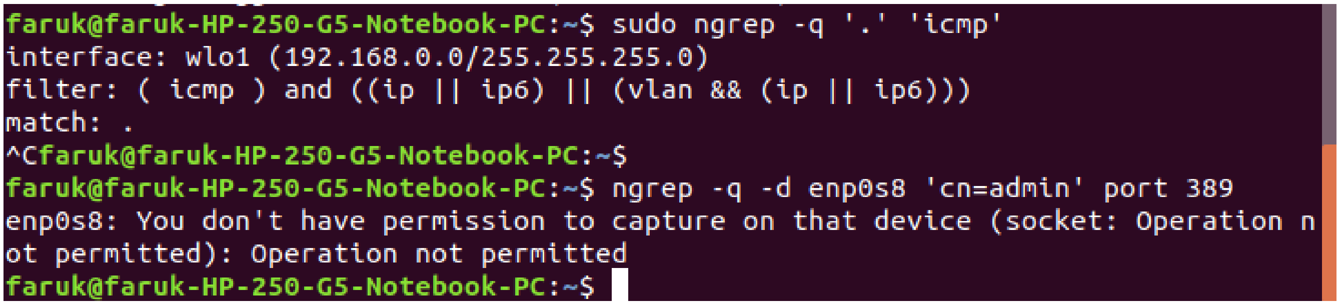
The above command will copy/sync all the files and directories present in directory `foo` to directory *bar*. If the destination directory is not present (here `bar`), rsync automatically creates one and copies all the data in it.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ cd Desktop/
faruk@faruk-HP-250-G5-Notebook-PC:~/Desktop$ rsync -avh new/ bar/
sending incremental file list
created directory bar
./
f1
f2
f3
f4
f5
f6
f7

sent 429 bytes  received 178 bytes  1.21K bytes/sec
total size is 0  speedup is 0.00
faruk@faruk-HP-250-G5-Notebook-PC:~/Desktop$ 
```

11.ngrep : grep for your network. **Ngrep** (**network grep**) is a simple yet powerful network packet analyzer. It is a grep-like tool applied to the network layer – it matches traffic passing over a network interface. The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

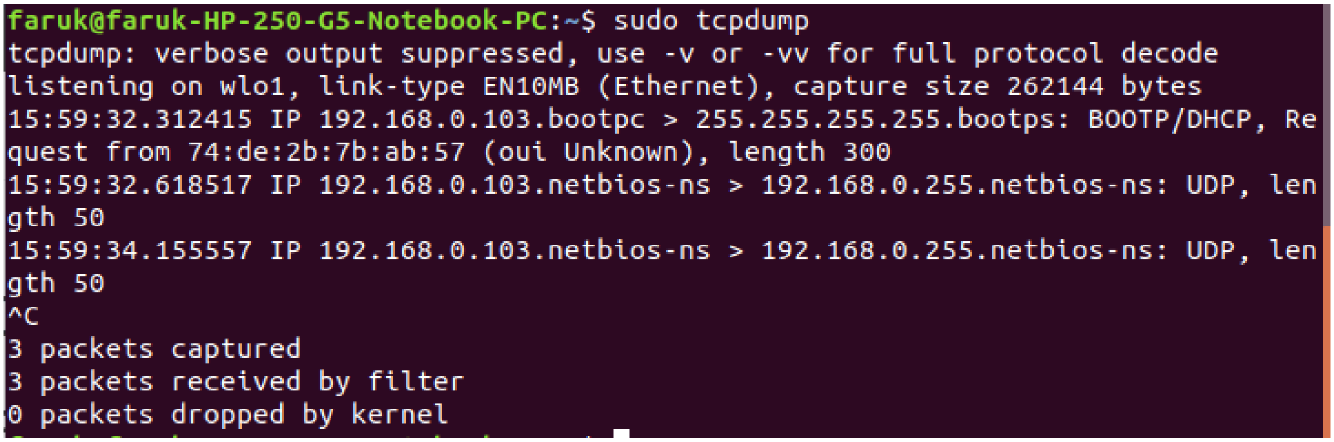
The following command will help you match all ping requests on the default working interface. You need to open another terminal and try to ping another remote machine. The `-q` flag tell **ngrep** to work quietly, to not output any information other than packet headers and their payloads.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo ngrep -q '.' 'icmp'
interface: wlo1 (192.168.0.0/255.255.255.0)
filter: ( icmp ) and ((ip || ip6) || (vlan && (ip || ip6)))
match: .
^Cfaruk@faruk-HP-250-G5-Notebook-PC:~$
faruk@faruk-HP-250-G5-Notebook-PC:~$ ngrep -q -d enp0s8 'cn=admin' port 389
enp0s8: You don't have permission to capture on that device (socket: Operation n
ot permitted): Operation not permitted
faruk@faruk-HP-250-G5-Notebook-PC:~$ 
```

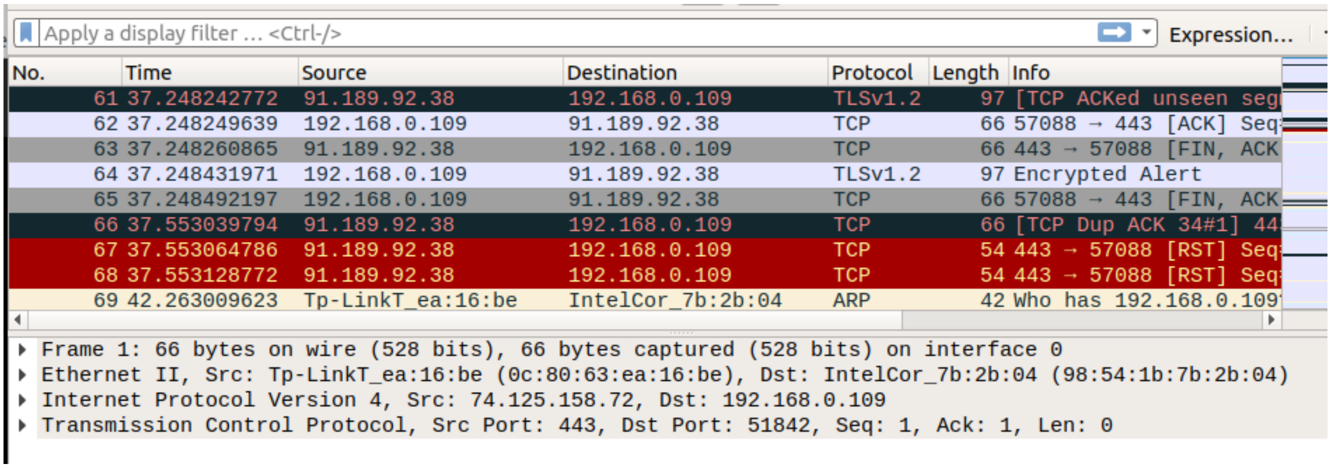
12.tcpdump : show me all packets on port 80 . tcpdumpis a command-line utility that you can use to capture and inspect network traffic going to and from your system. It is the most commonly used tool among network administrators for troubleshooting network issues and security testing.

Despite its name, with tcpdump, you can also capture non-TCP traffic such as UDP, ARP, or ICMP. The captured packets can be written to a file or standard output. One of the most powerful features of the tcpdump command is its ability to use filters and capture only the data you wish to analyze.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
15:59:32.312415 IP 192.168.0.103.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Re
quest from 74:de:2b:7b:ab:57 (oui Unknown), length 300
15:59:32.618517 IP 192.168.0.103.netbios-ns > 192.168.0.255.netbios-ns: UDP, len
gth 50
15:59:34.155557 IP 192.168.0.103.netbios-ns > 192.168.0.255.netbios-ns: UDP, len
gth 50
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel
```

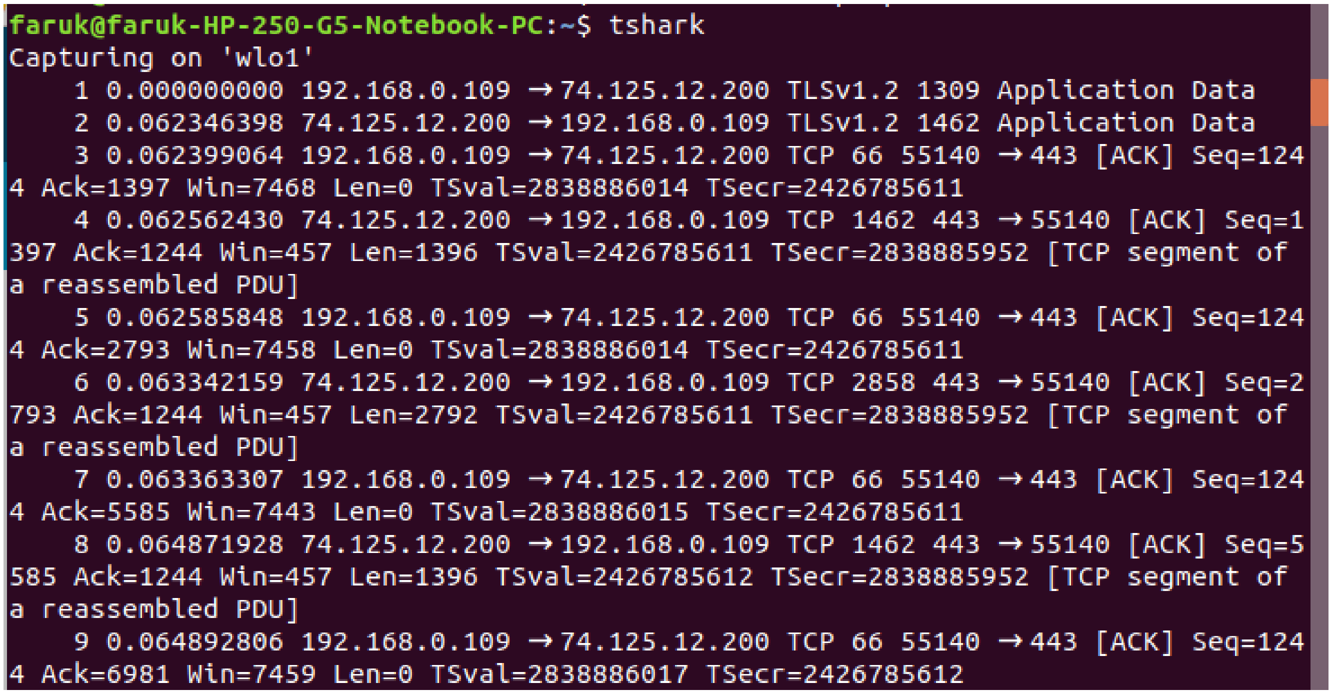
13. wireshark : look at those packets in a GUI . Wireshark is a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.



No.	Time	Source	Destination	Protocol	Length	Info
61	37.248242772	91.189.92.38	192.168.0.109	TLSv1.2	97	[TCP ACKed unseen seg
62	37.248249639	192.168.0.109	91.189.92.38	TCP	66	57088 → 443 [ACK] Seq
63	37.248260865	91.189.92.38	192.168.0.109	TCP	66	443 → 57088 [FIN, ACK
64	37.248431971	192.168.0.109	91.189.92.38	TLSv1.2	97	Encrypted Alert
65	37.248492197	192.168.0.109	91.189.92.38	TCP	66	57088 → 443 [FIN, ACK
66	37.553039794	91.189.92.38	192.168.0.109	TCP	66	[TCP Dup ACK 34#1] 44
67	37.553064786	91.189.92.38	192.168.0.109	TCP	54	443 → 57088 [RST] Seq
68	37.553128772	91.189.92.38	192.168.0.109	TCP	54	443 → 57088 [RST] Seq
69	42.263009623	Tp-LinkT_ea:16:be	IntelCor_7b:2b:04	ARP	42	Who has 192.168.0.109

```
▸ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▸ Ethernet II, Src: Tp-LinkT_ea:16:be (0c:80:63:ea:16:be), Dst: IntelCor_7b:2b:04 (98:54:1b:7b:2b:04)
▸ Internet Protocol Version 4, Src: 74.125.158.72, Dst: 192.168.0.109
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 51842, Seq: 1, Ack: 1, Len: 0
```

14. tshark : command line super powerful packet analysis. Dump and analyze network traffic. **TShark** is a network protocol analyzer. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. **TShark**'s native capture file format is **libpcap** format, which is also the format used by **tcpdump** and various other tools.



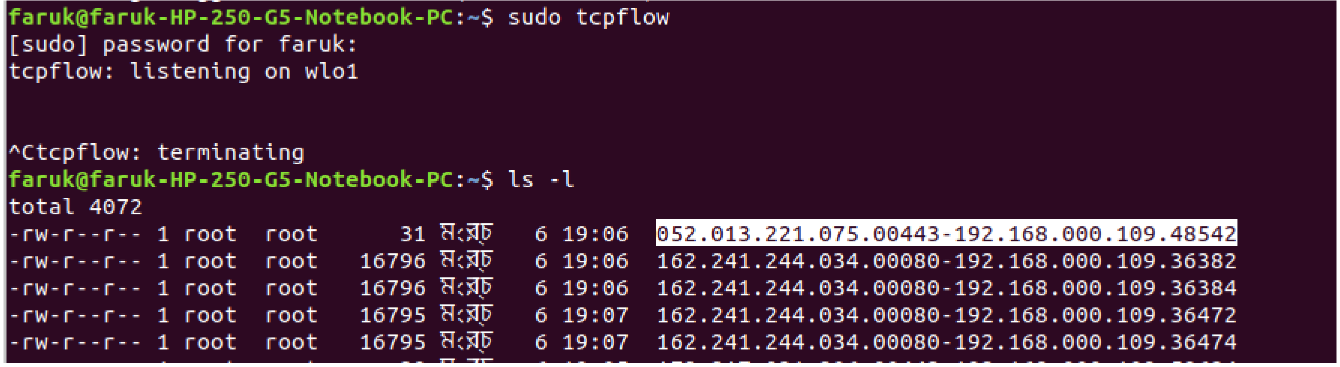
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ tshark
Capturing on 'wlo1'
    1 0.000000000 192.168.0.109 → 74.125.12.200 TLSv1.2 1309 Application Data
    2 0.062346398 74.125.12.200 → 192.168.0.109 TLSv1.2 1462 Application Data
    3 0.062399064 192.168.0.109 → 74.125.12.200 TCP 66 55140 → 443 [ACK] Seq=124
4 Ack=1397 Win=7468 Len=0 TSval=2838886014 TSecr=2426785611
    4 0.062562430 74.125.12.200 → 192.168.0.109 TCP 1462 443 → 55140 [ACK] Seq=1
397 Ack=1244 Win=457 Len=1396 TSval=2426785611 TSecr=2838885952 [TCP segment of
a reassembled PDU]
    5 0.062585848 192.168.0.109 → 74.125.12.200 TCP 66 55140 → 443 [ACK] Seq=124
4 Ack=2793 Win=7458 Len=0 TSval=2838886014 TSecr=2426785611
    6 0.063342159 74.125.12.200 → 192.168.0.109 TCP 2858 443 → 55140 [ACK] Seq=2
793 Ack=1244 Win=457 Len=2792 TSval=2426785611 TSecr=2838885952 [TCP segment of
a reassembled PDU]
    7 0.063363307 192.168.0.109 → 74.125.12.200 TCP 66 55140 → 443 [ACK] Seq=124
4 Ack=5585 Win=7443 Len=0 TSval=2838886015 TSecr=2426785611
    8 0.064871928 74.125.12.200 → 192.168.0.109 TCP 1462 443 → 55140 [ACK] Seq=5
585 Ack=1244 Win=457 Len=1396 TSval=2426785612 TSecr=2838885952 [TCP segment of
a reassembled PDU]
    9 0.064892806 192.168.0.109 → 74.125.12.200 TCP 66 55140 → 443 [ACK] Seq=124
4 Ack=6981 Win=7459 Len=0 TSval=2838886017 TSecr=2426785612
```

15. tcpflow: capture and assemble tcpstreams . **tcpflow** is a program that captures data transmitted as part of TCP connections (flows), and stores the data in a way that is convenient for protocol analysis or debugging. A program like ***tcpdump(4)*** shows a summary of packets seen on the wire, but usually doesn't store the data that's actually being transmitted. In contrast, tcpflow reconstructs the actual data streams and stores each flow in a separate file for later analysis. tcpflow understands TCP sequence numbers and will correctly reconstruct data streams regardless of retransmissions or out-of-order delivery.

By default tcpflow stores all captured data in files that have names in the form.

```
sourceip.sourceport-destip.destport
192.168.043.031.52920-216.058.210.034.00443
```

Now let’s do a directory listing to see if tcp flow has been captured in any files.



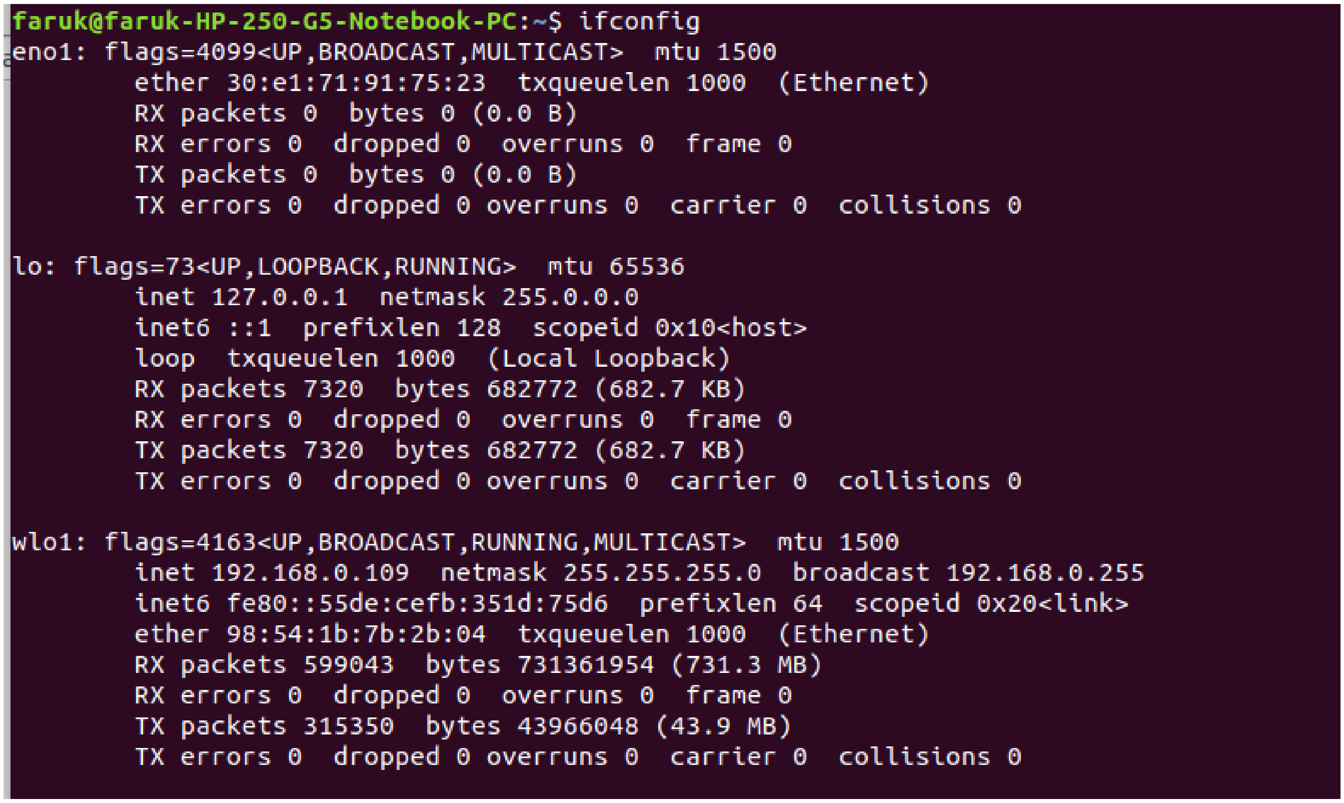
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo tcpflow
[sudo] password for faruk:
tcpflow: listening on wlo1


^Ctcpflow: terminating
faruk@faruk-HP-250-G5-Notebook-PC:~$ ls -l
total 4072
-rw-r--r-- 1 root  root      31 মার্চ   6 19:06  052.013.221.075.00443-192.168.000.109.48542
-rw-r--r-- 1 root  root   16796 মার্চ   6 19:06  162.241.244.034.00080-192.168.000.109.36382
-rw-r--r-- 1 root  root   16796 মার্চ   6 19:06  162.241.244.034.00080-192.168.000.109.36384
-rw-r--r-- 1 root  root   16795 মার্চ   6 19:07  162.241.244.034.00080-192.168.000.109.36472
-rw-r--r-- 1 root  root   16795 মার্চ   6 19:07  162.241.244.034.00080-192.168.000.109.36474
```

The first file **192.168.043.031.52920-216.058.210.034.00443** contains data transfered from host **192.168.043.031** (the localhost on which tcpflow was run) via port **52920**, to host **216.058.210.034** (the remote host) via port **443**.

16. ***ifconfig :*** ifconfig command is used for displaying current network configuration information, setting up an ip address ,netmask or broadcast address to an network interface,creating an alias for network interface, setting up hardware address and enable or disable network interfaces .

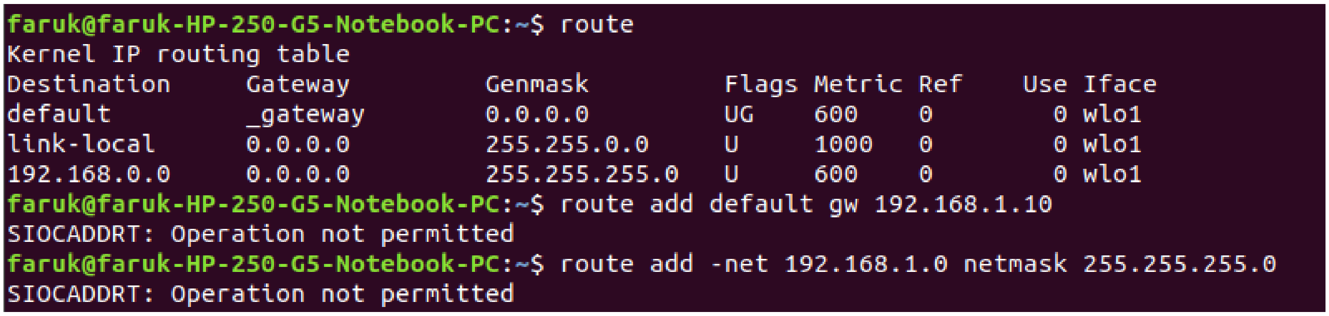
Basic information displayed upon using ifconfig are :
 ip address
Mac address
MTU(Maximum Transmission Unit)

ifconfig output show the ip address of 3 networks i.e. Ethernet,local network and WLAN.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ ifconfig
eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 30:e1:71:91:75:23  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        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 7320  bytes 682772 (682.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7320  bytes 682772 (682.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.109  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::55de:cefb:351d:75d6  prefixlen 64  scopeid 0x20<link>
        ether 98:54:1b:7b:2b:04  txqueuelen 1000  (Ethernet)
        RX packets 599043  bytes 731361954 (731.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 315350  bytes 43966048 (43.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

17. **route :** Route command is used to show/manipulate the IP routing table. It is primarily used to setup static routes to specific host or networks via an interface.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlo1
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 wlo1
192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0 wlo1
faruk@faruk-HP-250-G5-Notebook-PC:~$ route add default gw 192.168.1.10
SIOCADDRT: Operation not permitted
faruk@faruk-HP-250-G5-Notebook-PC:~$ route add -net 192.168.1.0 netmask 255.255.255.0
SIOCADDRT: Operation not permitted
```

18. ip : The ip command is used to assign an address to a network interface and/or configure network interface parameters on Linux operating systems.
Type the following command to list and show all ip address associated on on all network interfaces:

```
ip a or
```

ip addr

You can select between IPv4 and IPv6 using the following syntax:

```
ip -4 a
ip -6 a
```

Assigns the IP address to the interface

The syntax is as follows to add an IPv4/IPv6 address:

```
ip a add {ip_addr/mask} dev {interface}
```

To assign 192.168.1.200/255.255.255.0 to eth0, enter:

```
ip a add 192.168.1.200/255.255.255.0 dev eth0
```

OR

```
ip a add 192.168.1.200/24 dev eth0
```

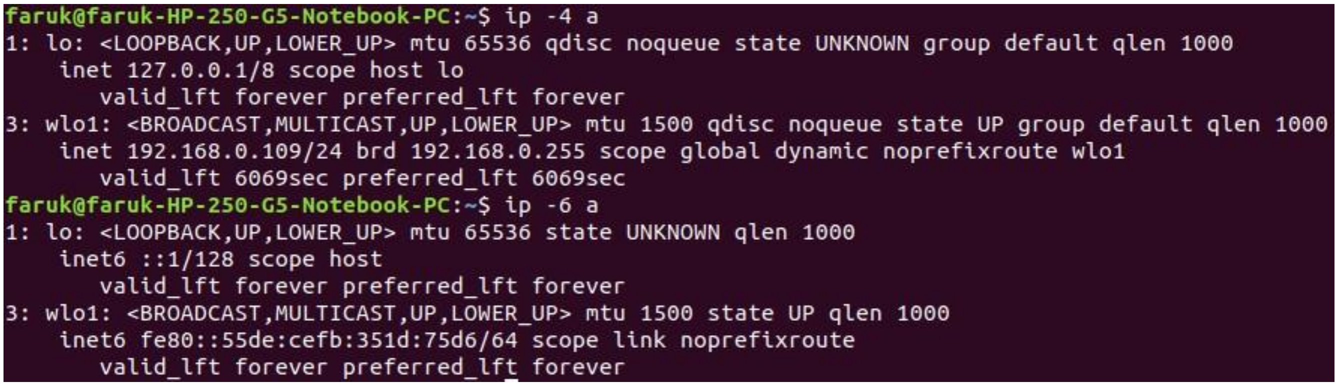
Remove / Delete the IP address from the interface

The syntax is as follows to remove an IPv4/IPv6 address:

```
ip a del {ipv6_addr_OR_ipv4_addr} dev {interface}
```

To delete 192.168.1.200/24 from eth0, enter:

```
ip a del 192.168.1.200/24 dev eth0
```

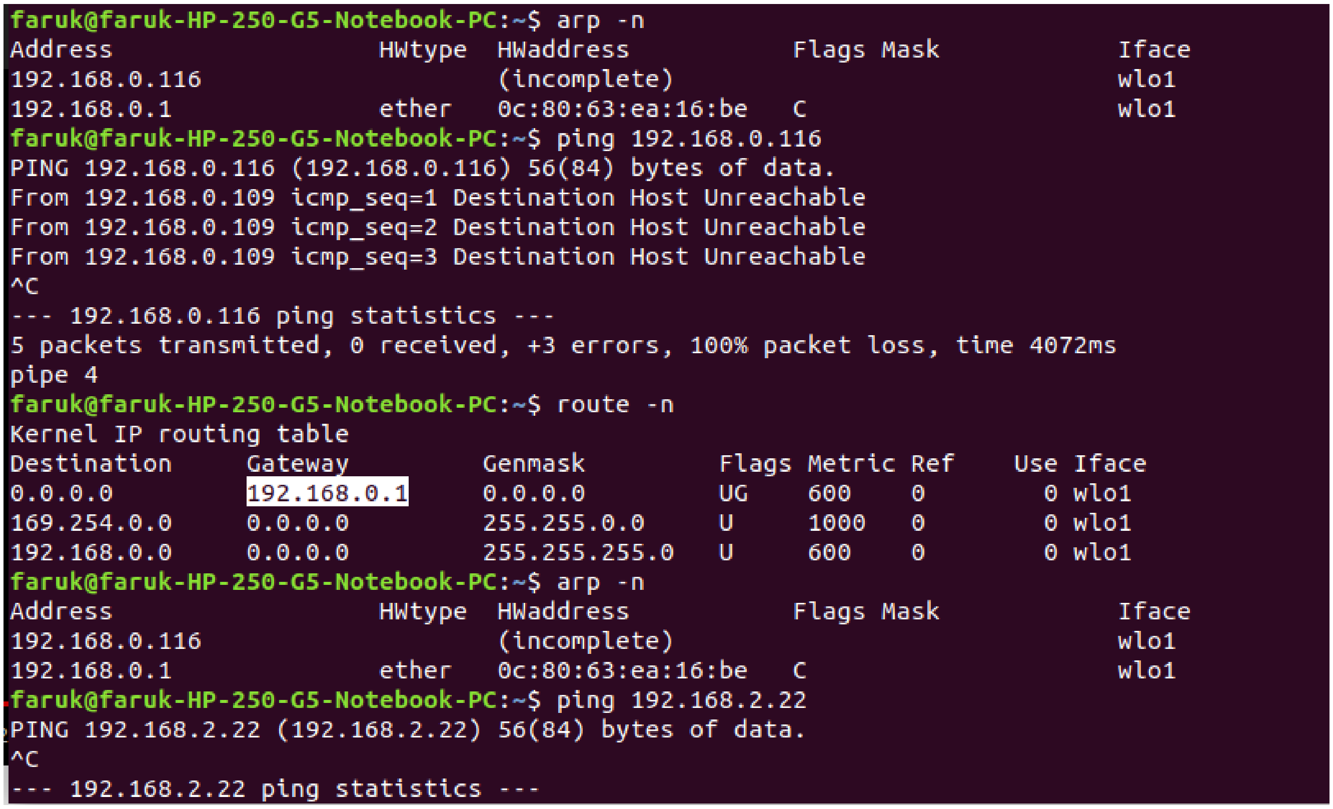


```
faruk@faruk-HP-250-G5-Notebook-PC:~$ ip -4 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    inet 192.168.0.109/24 brd 192.168.0.255 scope global dynamic noprefixroute wlo1
       valid_lft 6069sec preferred_lft 6069sec
faruk@faruk-HP-250-G5-Notebook-PC:~$ ip -6 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1000
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 fe80::55de:cefb:351d:75d6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

19. arp : ARP is the abbreviation for Address Resolution Protocol, which is used to find the address of a network neighbor for a given IPv4 address. This protocol is used by network nodes to resolve IP addresses to their corresponding MAC addresses.

The arp command could be used for the following purposes:

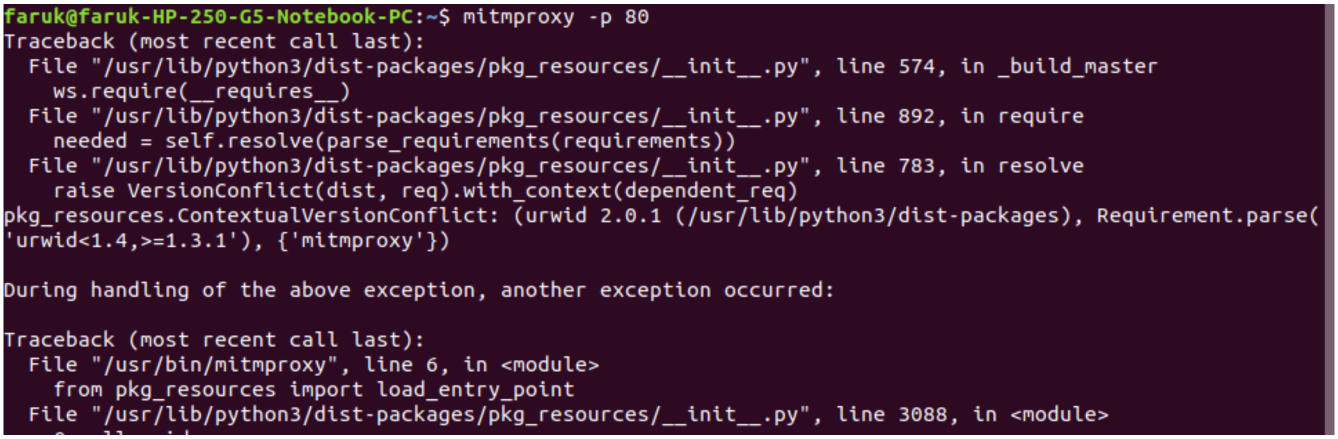
- Display IP address to MAC address resolution information for neighboring devices.
- Clear address mapping entries and set them up manually.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.116                    (incomplete)                              wlo1
192.168.0.1              ether   0c:80:63:ea:16:be   C                     wlo1
faruk@faruk-HP-250-G5-Notebook-PC:~$ ping 192.168.0.116
PING 192.168.0.116 (192.168.0.116) 56(84) bytes of data.
From 192.168.0.109 icmp_seq=1 Destination Host Unreachable
From 192.168.0.109 icmp_seq=2 Destination Host Unreachable
From 192.168.0.109 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.0.116 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4072ms
pipe 4
faruk@faruk-HP-250-G5-Notebook-PC:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    600    0        0 wlo1
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlo1
192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0 wlo1
faruk@faruk-HP-250-G5-Notebook-PC:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.116                    (incomplete)                              wlo1
192.168.0.1              ether   0c:80:63:ea:16:be   C                     wlo1
faruk@faruk-HP-250-G5-Notebook-PC:~$ ping 192.168.2.22
PING 192.168.2.22 (192.168.2.22) 56(84) bytes of data.
^C
--- 192.168.2.22 ping statistics ---
```

20. <u>mitmproxy :</u> spy on ssl connections your programs are making . **mitmproxy** is an SSL-capable man-in-the-middle HTTP proxy. It provides a console interface that allows traffic flows to be inspected and edited on the fly.

To open it, go to the terminal and type "**mitmproxy -parameter"** and for getting help on commands, type "**mitmproxy –h"**. To start the mitmproxy, type "**mitmproxy –p portnumber"**. In this case, it is "mitmproxy –p 80".

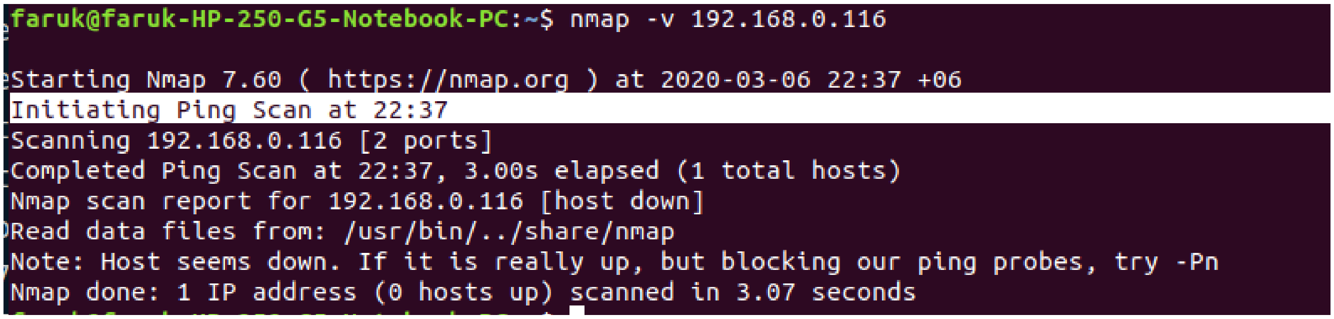


```
faruk@faruk-HP-250-G5-Notebook-PC:~$ mitmproxy -p 80
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/pkg_resources/__init__.py", line 574, in _build_master
    ws.require(__requires__)
  File "/usr/lib/python3/dist-packages/pkg_resources/__init__.py", line 892, in require
    needed = self.resolve(parse_requirements(requirements))
  File "/usr/lib/python3/dist-packages/pkg_resources/__init__.py", line 783, in resolve
    raise VersionConflict(dist, req).with_context(dependent_req)
pkg_resources.ContextualVersionConflict: (urwid 2.0.1 (/usr/lib/python3/dist-packages), Requirement.parse(
'urwid<1.4,>=1.3.1'), {'mitmproxy'})

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/bin/mitmproxy", line 6, in <module>
    from pkg_resources import load_entry_point
  File "/usr/lib/python3/dist-packages/pkg_resources/__init__.py", line 3088, in <module>
```

21. <u>nmap :</u> in ur network scanning ur ports . **Nmap** is a tool used for determining the hosts that are running and what services the hosts are running . The **Nmap** aka **Network Mapper** is an open source and a very versatile tool for Linux system/network administrators. **Nmap** is used for **exploring networks**, **perform security scans**, **network audit** and **finding open ports** on remote machine. It scans for Live hosts, Operating systems, packet filters and open ports running on remote hosts.

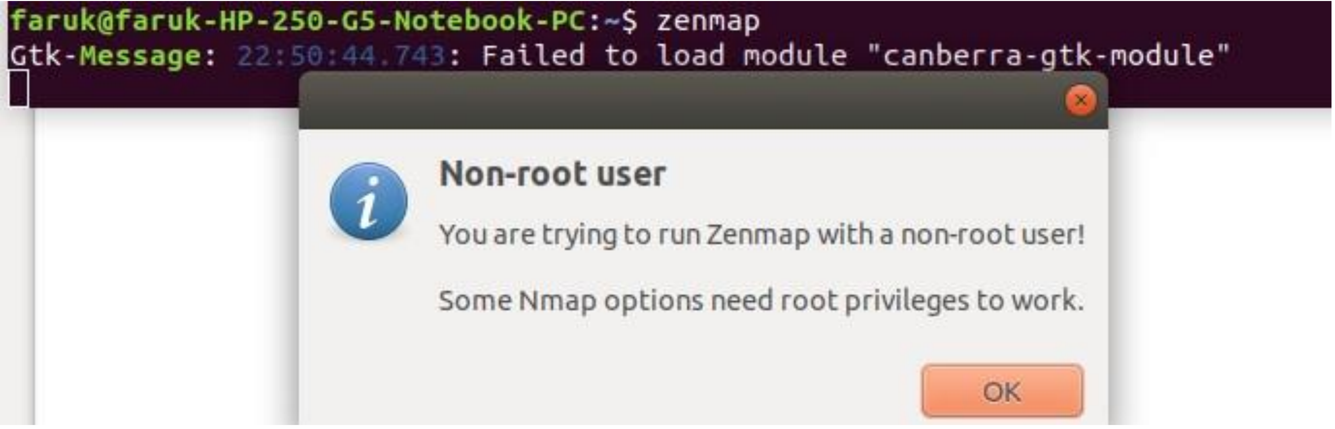
Syntax : nmap [Scan Type(s)] [Options] {target specification}

You can see that the below command with “-v” option is giving more detailed information about the remote machine.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ nmap -v 192.168.0.116

Starting Nmap 7.60 ( https://nmap.org ) at 2020-03-06 22:37 +06
Initiating Ping Scan at 22:37
Scanning 192.168.0.116 [2 ports]
Completed Ping Scan at 22:37, 3.00s elapsed (1 total hosts)
Nmap scan report for 192.168.0.116 [host down]
Read data files from: /usr/bin/../share/nmap
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.07 seconds
```

22.zenmap : GUI for nmap. Graphical Nmap frontend and results viewer. Zenmap is a multi-platform graphical Nmap frontend and results viewer. Zenmap aims to make Nmap easy for beginners to use while giving experienced Nmap users advanced features. Frequently used scans can be saved as profiles to make them easy to run repeatedly.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ zenmap
Gtk-Message: 22:50:44.743: Failed to load module "canberra-gtk-module"
```

Non-root user

You are trying to run Zenmap with a non-root user!

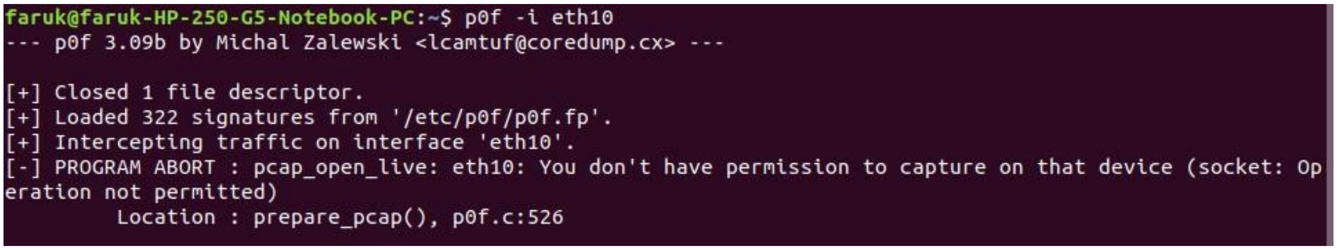
Some Nmap options need root privileges to work.

23.p0f : identify OS of hosts connecting to you .identify remote systems passively. P0f is an OS Fingerprinting and Forensics Tool that utilizes an array of sophisticated, purely passive traffic fingerprinting mechanisms to identify the players behind any incidental TCP/IP communications.**p0f** uses a fingerprinting technique based on analyzing the structure of a TCP/IP packet to determine the operating system and other configuration properties of a remote host.
In this Forensics Tool, To Lanch p0f use this comment faruk**@faruk$p0f -i -eth0**

Use interface eth0 ***(-i eth0)***

promiscuous mode ***(-p)***

saving the results to a file ***(-o /tmp/p0f.log)***:



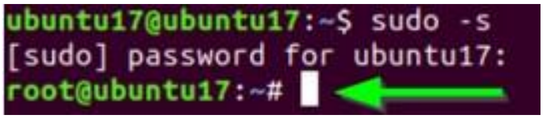
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ p0f -i eth10
--- p0f 3.09b by Michal Zalewski <lcamtuf@coredump.cx> ---

[+] Closed 1 file descriptor.
[+] Loaded 322 signatures from '/etc/p0f/p0f.fp'.
[+] Intercepting traffic on interface 'eth10'.
[-] PROGRAM ABORT : pcap_open_live: eth10: You don't have permission to capture on that device (socket: Op
eration not permitted)
         Location : prepare_pcap(), p0f.c:526
```

<u>24.</u> **<u>openvpn :</u>** **OpenVPN** open source **OpenVPN** CLI program. The open source project client program is the main method of getting your **Linux** system connected to the Access Server. The package is available in most distributions and is known simply as **openvpn**.

1. Log in as a root user. If you are not a root user, then run the following command and tap the Enter key.

sudo -s

Type your root password and tap the **Enter** key.



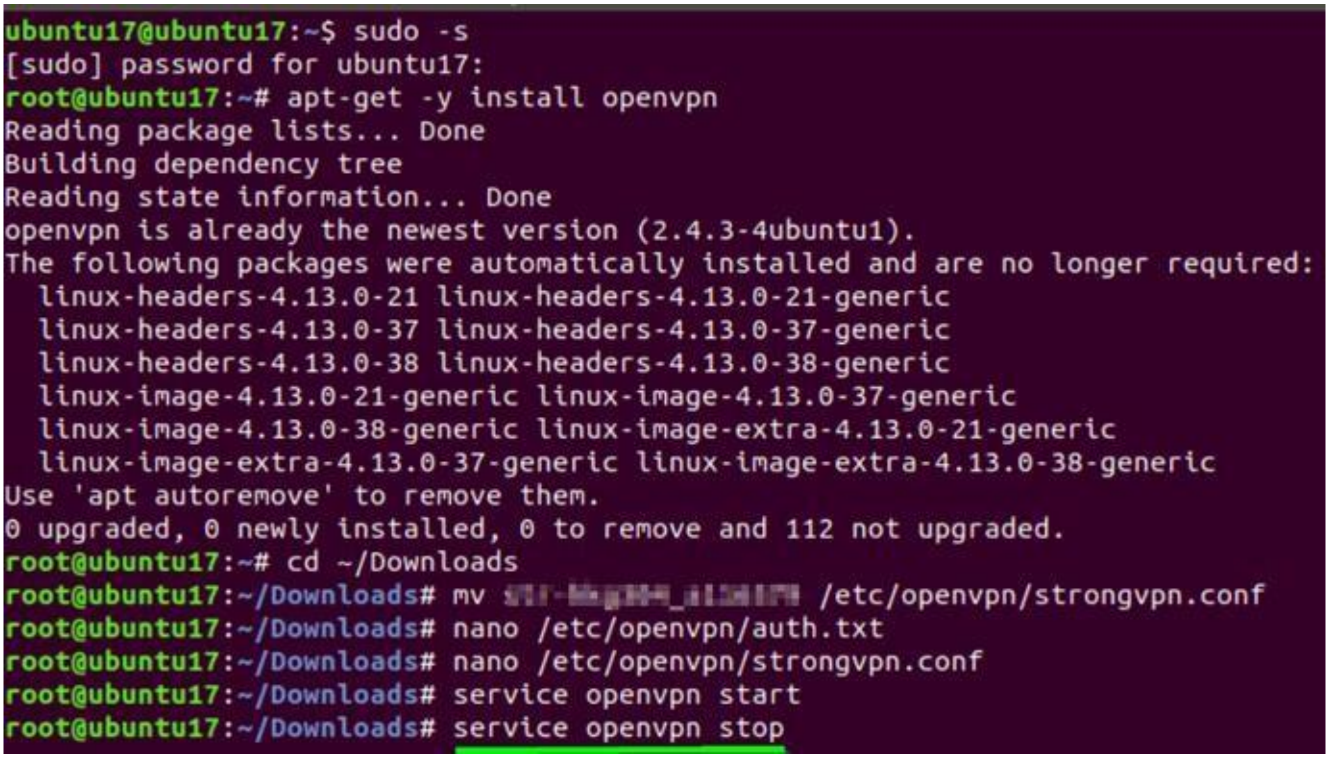
```
ubuntu17@ubuntu17:~$ sudo -s
[sudo] password for ubuntu17:
root@ubuntu17:~#
```

To connect the VPN, type the below command:

service openvpn start

To disconnect StrongVPN, type the following command:

service openvpn stop



```
ubuntu17@ubuntu17:~$ sudo -s
[sudo] password for ubuntu17:
root@ubuntu17:~# apt-get -y install openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
openvpn is already the newest version (2.4.3-4ubuntu1).
The following packages were automatically installed and are no longer required:
  linux-headers-4.13.0-21 linux-headers-4.13.0-21-generic
  linux-headers-4.13.0-37 linux-headers-4.13.0-37-generic
  linux-headers-4.13.0-38 linux-headers-4.13.0-38-generic
  linux-image-4.13.0-21-generic linux-image-4.13.0-37-generic
  linux-image-4.13.0-38-generic linux-image-extra-4.13.0-21-generic
  linux-image-extra-4.13.0-37-generic linux-image-extra-4.13.0-38-generic
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 112 not upgraded.
root@ubuntu17:~# cd ~/Downloads
root@ubuntu17:~/Downloads# mv [redacted] /etc/openvpn/strongvpn.conf
root@ubuntu17:~/Downloads# nano /etc/openvpn/auth.txt
root@ubuntu17:~/Downloads# nano /etc/openvpn/strongvpn.conf
root@ubuntu17:~/Downloads# service openvpn start
root@ubuntu17:~/Downloads# service openvpn stop
```

25. **wireguard :** a newer vpn. WireGuard is a new, experimental VPN protocol that aims to offer a simpler, faster, and more secure solution for VPN tunneling than the existing VPN protocols. WireGuard has some major differences when compared to OpenVPN and IPSec, such as the code size (under 4,000 lines!), speed, and encryption standards. it's very **secure.**
The installation process is based on Ubuntu. Documentation regarding other platforms is available on the **WireGuard website**.

1 . Connect to your server via SSH.

2 . Install Linux kernel headers and WireGuard.

$ sudo apt-get install linux-headers-$(uname --kernel-release) # installs the right kernel headers for your version
$ sudo add-apt-repository ppa:wireguard/wireguard
$ sudo apt-get update
$ sudo apt-get install wireguard

26. **nc :** Netcat (nc) command is a powerful tool to analyze network connections, scan for open ports, transfer data etc. It is a networking utility for reading from and writing to network connections using TCP or UDP protocols.

Test if a particular TCP port of a remote host is open

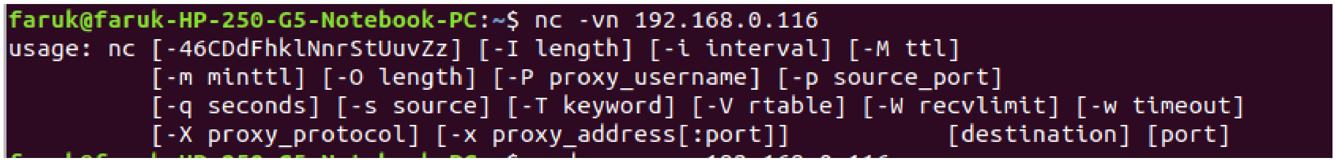
```
nc -vn 192.168.40.146 2424
```

Output if the 2424 port on remote server is closed

```
nc: connect to 192.168.40.146 port 2424 (tcp) failed: Connection refused
```

Output if the port on remote server is opened (e.g. 22 port)

```
Connection to 192.168.40.146 22 port [tcp/*] succeeded!
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4
```



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ nc -vn 192.168.0.116
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s source] [-T keyword] [-V rtable] [-W recvlimit] [-w timeout]
          [-X proxy_protocol] [-x proxy_address[:port]]           [destination] [port]
```

27. **socat :** proxy a TCP socket to a unix domain socket+lot more. **Socat** is a command line based utility that establishes two bidirectional byte streams and transfers data between them. Because the streams can be constructed from a large set of different types of data sinks and sources (see address types), and because lots of address options may be applied to the streams, socat can be used for many different purposes.

socat Usage:

1. TCP port forwarder
2. External socksifier
3. Attacking weak firewalls (security testing)

4. A shell interface to UNIX sockets
5. IP6 relay
6. For redirecting TCP oriented programs to a serial line
7. Logically connect serial lines on different computers
8. Security testing and research
9. Establish a relatively secure environment (su and chroot) for running client or server shell scripts with network connections etc

WARNING! These examples may open your computer ports and sockets to other Internet users. You must have a good understanding of TCP/IP and UNIX networking to use this tool.

Install socat Under Ubuntu Linux

Type the following command:

```
$ sudo apt-get update && sudo apt-get install socat
```

To redirect all port 80 conenctions to ip 202.54.1.5, enter:

```
# socat TCP-LISTEN:80,fork TCP:202.54.1.5:80
```

28. telnet : like SSH but insecure . Telnet helps to -

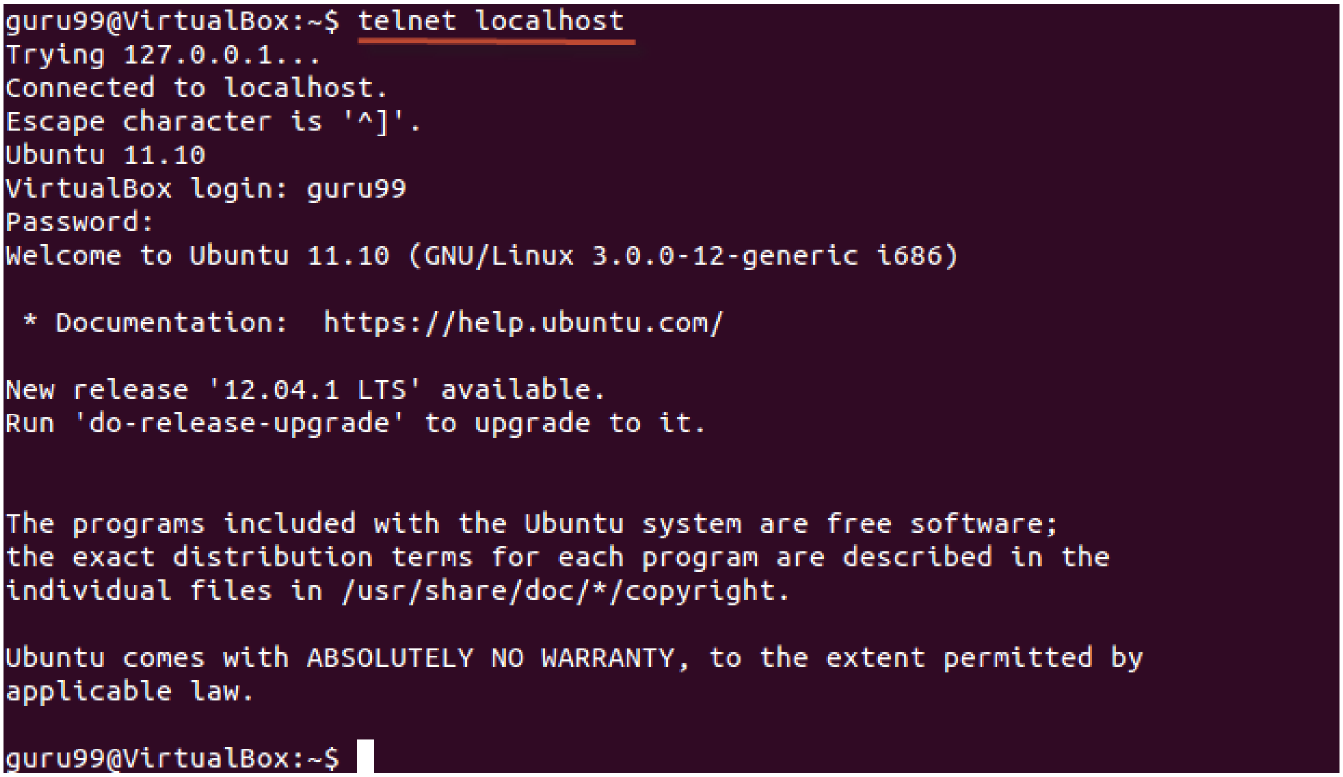
- connect to a remote Linux computer
- run programs remotely and conduct administration

The syntax for this utility is: t

elnet hostname="" or=""

Example: `telnet localhost`

For demonstration purpose, we will connect to your computer (localhost). The utility will ask your username and password.



```
guru99@VirtualBox:~$ telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 11.10
VirtualBox login: guru99
Password:
Welcome to Ubuntu 11.10 (GNU/Linux 3.0.0-12-generic i686)

 * Documentation:  https://help.ubuntu.com/

New release '12.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

guru99@VirtualBox:~$ 
```

29. ftp/sftp : The ftp stands for **F**ile **T**ransfer **P**rotocol. It connect to the remote host to exchange files and directories from one host to another over a network which can be LAN or any other.
The sftp stands for **S**ecure **ftp.**

Most of the ftp commands are applicable to sftp. So wherever, you need to use sftp, you can use it at the place of ftp.

Ftp is the **most preferred protocol for data transfer** amongst computers.

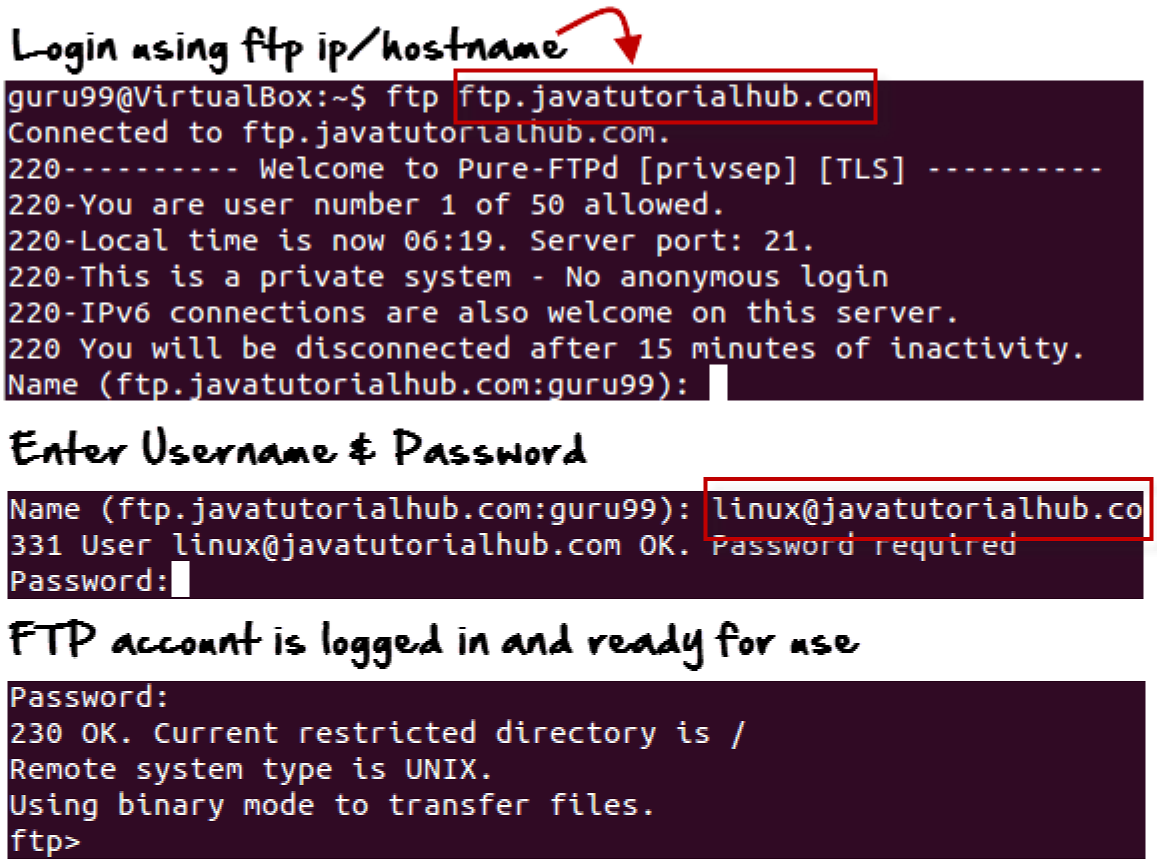
You can use FTP to -

- Logging in and establishing a connection with a remote host
- Upload and download files
- Navigating through directories
- Browsing contents of the directories

The syntax to establish an FTP connection to a remote host is -

```
ftp hostname=”” or=””
```

Once you enter this command, it will ask you for **authentication** via username and password.



Login using ftp ip/hostname

```
guru99@VirtualBox:~$ ftp ftp.javatutorialhub.com
Connected to ftp.javatutorialhub.com.
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 06:19. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
Name (ftp.javatutorialhub.com:guru99):
```

Enter Username & Password

```
Name (ftp.javatutorialhub.com:guru99): linux@javatutorialhub.co
331 User linux@javatutorialhub.com OK. Password required
Password:
```

FTP account is logged in and ready for use

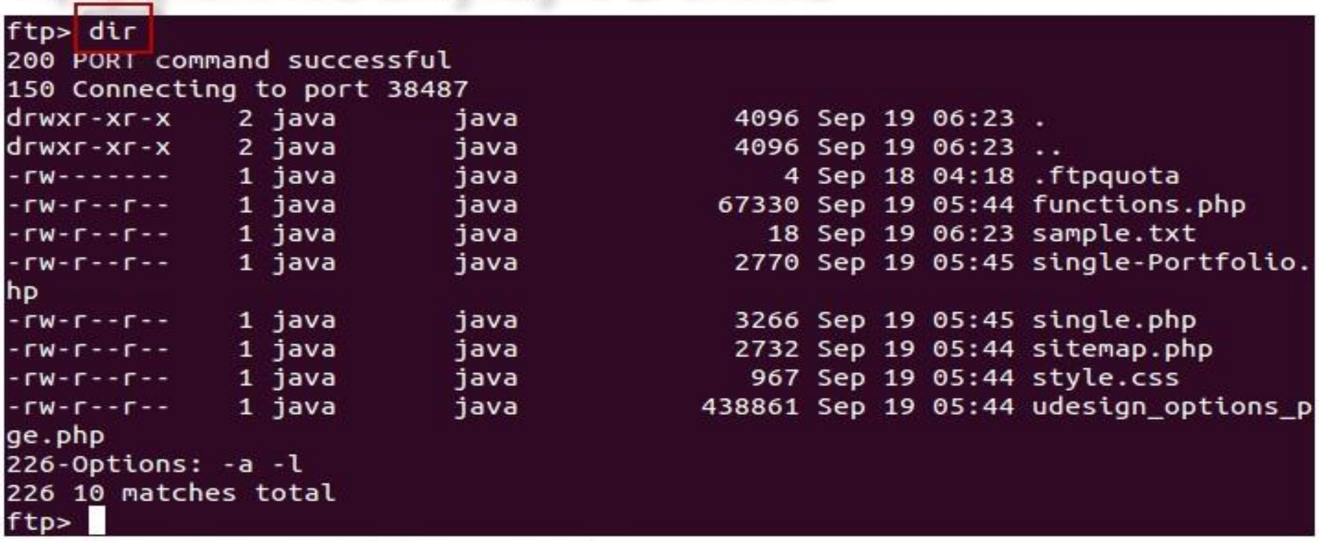
```
Password:
230 OK. Current restricted directory is /
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Once a connection is established, and you are logged in, you may use the following commands to perform different actions.

Command	Function
dir	Display files in the current directory of a remote computer
cd "dirname"	change directory to "dirname" on a remote computer
put file	upload 'file' from local to remote computer
get file	Download 'file' from remote to local computer
quit	Logout

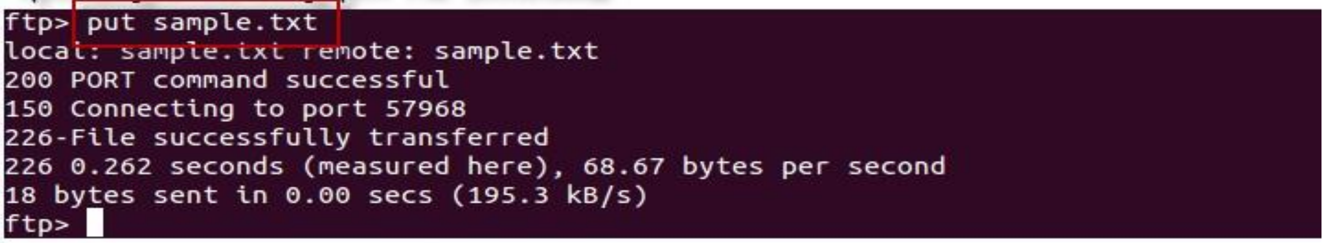
Let us run some of the important commands.

Displaying files in the directory using the 'dir' command



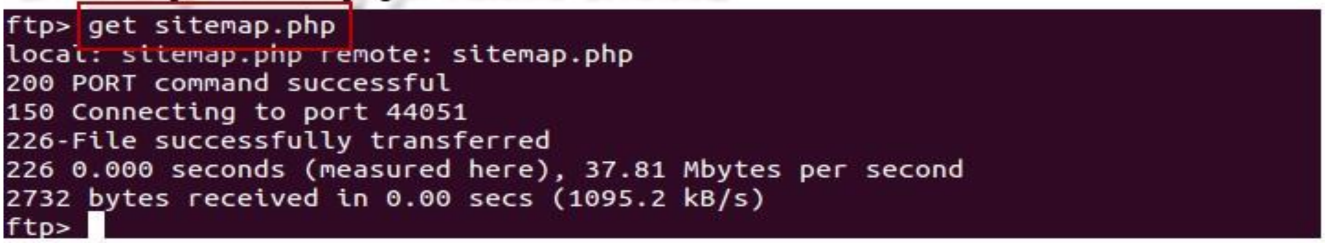
```
ftp> dir
200 PORT command successful
150 Connecting to port 38487
drwxr-xr-x    2 java       java             4096 Sep 19 06:23 .
drwxr-xr-x    2 java       java             4096 Sep 19 06:23 ..
-rw-------    1 java       java                4 Sep 18 04:18 .ftpquota
-rw-r--r--    1 java       java            67330 Sep 19 05:44 functions.php
-rw-r--r--    1 java       java               18 Sep 19 06:23 sample.txt
-rw-r--r--    1 java       java             2770 Sep 19 05:45 single-Portfolio.
hp
-rw-r--r--    1 java       java             3266 Sep 19 05:45 single.php
-rw-r--r--    1 java       java             2732 Sep 19 05:44 sitemap.php
-rw-r--r--    1 java       java              967 Sep 19 05:44 style.css
-rw-r--r--    1 java       java           438861 Sep 19 05:44 udesign_options_p
ge.php
226-Options: -a -l
226 10 matches total
ftp>
```

uploading a file using 'put file' command



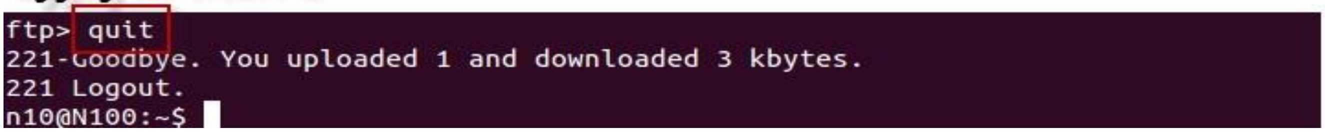
```
ftp> put sample.txt
local: sample.txt remote: sample.txt
200 PORT command successful
150 Connecting to port 57968
226-File successfully transferred
226 0.262 seconds (measured here), 68.67 bytes per second
18 bytes sent in 0.00 secs (195.3 kB/s)
ftp>
```

downloading a file using 'get filename' command



```
ftp> get sitemap.php
local: sitemap.php remote: sitemap.php
200 PORT command successful
150 Connecting to port 44051
226-File successfully transferred
226 0.000 seconds (measured here), 37.81 Mbytes per second
2732 bytes received in 0.00 secs (1095.2 kB/s)
ftp>
```

logging out from FTP



```
ftp> quit
221-Goodbye. You uploaded 1 and downloaded 3 kbytes.
221 Logout.
n10@N100:~$
```

Sftp -->By default, same SSH protocol is used to authenticate and establish a SFTP connection. To start an SFTP session, enter the username and remote hostname or IP address at the command prompt. Once authentication successful, you will see a shell with an **sftp>** prompt.

```
sftp tecmint@27.48.137.6
sftp> ?
```

Upload File

Put single or multiple files in remote system.

```
sftp> put local.profile
```

```
Uploading local.profile to /tecmint/local.profile
```

Upload Mutiple Files

Putting multiple files on in remote system.

```
sftp> mput *.xls
```

Download Files

Getting single or multiple files in local system.

```
sftp> get SettlementReport_1-10th.xls
Fetching /tecmint/SettlementReport_1-10th.xls to SettlementReport_1-10th.xls
```

Get multiple files on a local system.

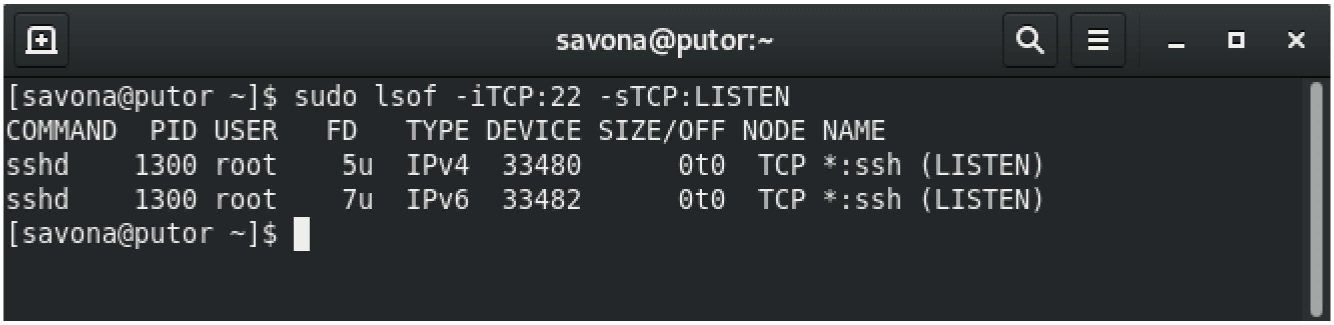
```
sftp> mget *.xls
```

30. netstat /ss / lsof /fuser : "what ports are server using?" 4 Ways to Find Which Process is Listening on a Specific Port .

i. Find Listening Process with lsof Command

The lsof command stands for "list open files". Since everything in Linux is a file, including ports and sockets, we can get all the information we need. To find the processes listening on a specific port with lsof, run the following command:

```
lsof -iTCP:22 -sTCP:LISTEN
```



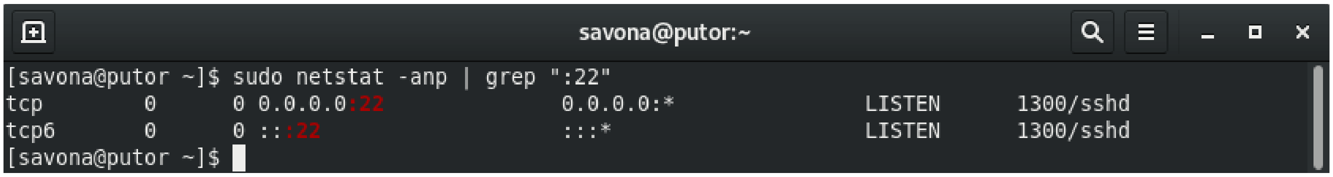
```
[savona@putor ~]$ sudo lsof -iTCP:22 -sTCP:LISTEN
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd    1300 root    5u  IPv4  33480      0t0  TCP *:ssh (LISTEN)
sshd    1300 root    7u  IPv6  33482      0t0  TCP *:ssh (LISTEN)
[savona@putor ~]$
```

ii. Use netstat Command to Find Process Listening on Port

The netstat command is an oldie but a goody. It has been around since the early 80's and is available not only on every Linux system, but also any UNIX or UNIX variant and Microsoft Windows since at least XP.

To find the processes listening on a specific port with netstat, use the following command:

```
netstat -anp | grep ":22"
```



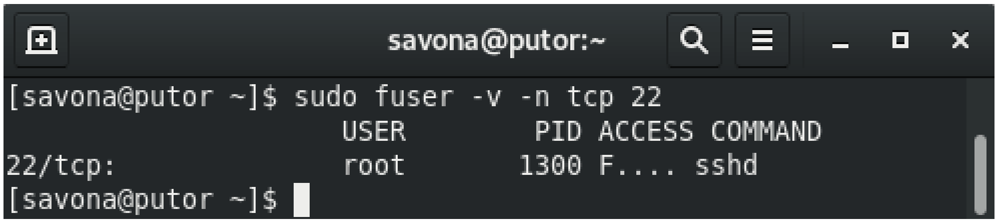
```
[savona@putor ~]$ sudo netstat -anp | grep ":22"
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1300/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      1300/sshd
[savona@putor ~]$
```

The -a option means all, -n means show ports numerically not names, -p means show process id and name.

Iii.Find Listening Process with fuser Command

The fuser command identifies which process is using a file or socket. With some clever options we can find all the information we need about which process is listening on a port.

sudo fuser 22/tcp



```
[savona@putor ~]$ sudo fuser -v -n tcp 22
                     USER        PID ACCESS COMMAND
22/tcp:              root       1300 F.... sshd
[savona@putor ~]$
```

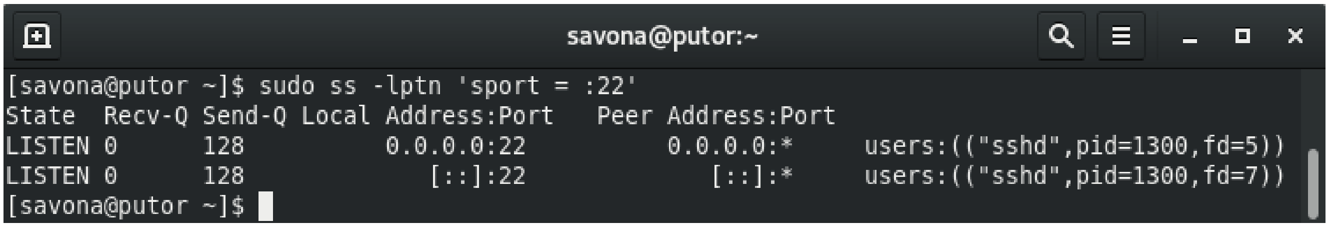
Here we use the -v option for verbose mode, and the -n option to select the corresponding namespace, followed by TCP port 22.

iv.Use the ss Command to Find Process Listening on Port

The ss command is pegged as the replacement for netstat. However, I think netstat is so ingrained in people that it will be a while before it goes anywhere. The ss commands syntax in my opinion is a little more convoluted, but that may be because I am old and set in my ways.

To find processes listening on a specific port with the ss command, we can use the following command:

```
sudo ss –lptn ’sport = :22’.
```



```
[savona@putor ~]$ sudo ss -lptn 'sport = :22'
State  Recv-Q Send-Q Local Address:Port   Peer Address:Port
LISTEN 0      128          0.0.0.0:22          0.0.0.0:*     users:(("sshd",pid=1300,fd=5))
LISTEN 0      128             [::]:22             [::]:*     users:(("sshd",pid=1300,fd=7))
[savona@putor ~]$
```

31. iptables : set up firewalls and NAT . IPTables is a rule based firewall and it is pre-installed on most of Linux operating system. By default it runs without any rules. IPTables was included in **Kernel 2.4**, prior it was called **ipchains** or **ipfwadm**. IPTables is a front-end tool to talk to the kernel and decides the packets to filter.**Iptables** is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel.

How to list all iptables rules on Linux

The procedure to list all rules on Linux is as follows:

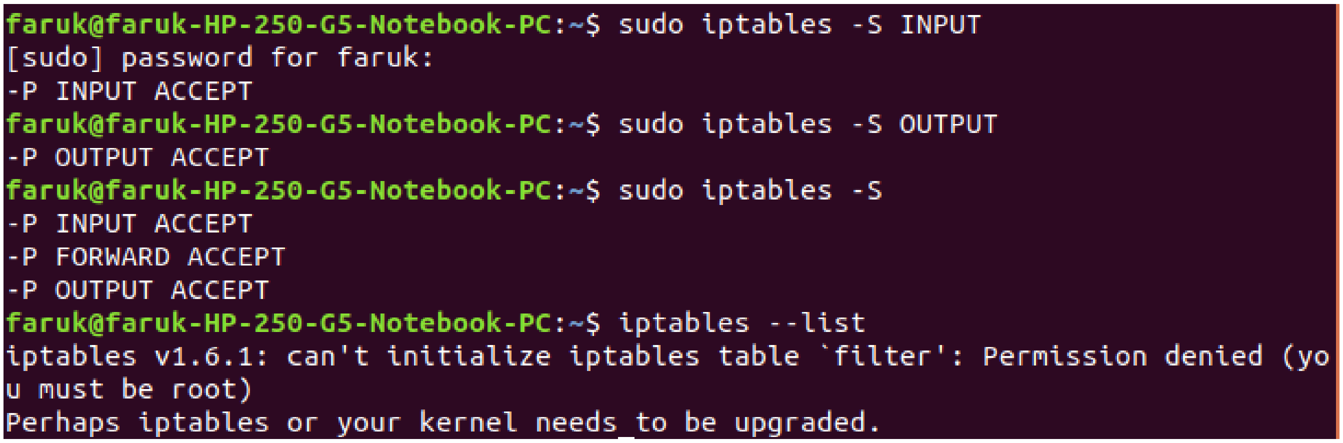
1. Open the terminal app or login using ssh:
   ```
   ssh user@server-name
   ```
2. To list all IPv4 rules :
   ```
   sudo iptables -S
   ```
3. To list all IPv6 rules :
   ```
   sudo ip6tables -S
   ```
4. To list all tables rules :
   ```
   sudo iptables -L -v -n | more
   ```
5. To list all rules for INPUT tables :
   ```
   sudo iptables -L INPUT -v -n
   sudo iptables -S INPUT
   ```

Viewing all iptables rules in Linux

The syntax is:

```
iptables -S
iptables --list
iptables -L
iptables -S TABLE_NAME
iptables --table NameHere --list
iptables -t NameHere -L -n -v --line-numbers
```

Print all rules in the selected chain



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo iptables -S INPUT
[sudo] password for faruk:
-P INPUT ACCEPT
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo iptables -S OUTPUT
-P OUTPUT ACCEPT
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
faruk@faruk-HP-250-G5-Notebook-PC:~$ iptables --list
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (yo
u must be root)
Perhaps iptables or your kernel needs to be upgraded.
```

32. nftables: new version of iptables.nftables is a new subsystem of the Linux kernel that replaces several parts of the Netfilter framework (upon which IPtables

is based), which allows for improved functionality. In IPtables, there are several chains and tables that are loaded by default.
In nftables, there are no default chains or tables.

In IPtables, there is only one target per rule.
In nftables, you can perform multiple actions within a single rule.

In nftables, there is a tool called ipset. Using ipset allows for the listing of multiple networks or addresses which can be matched in a single rule.

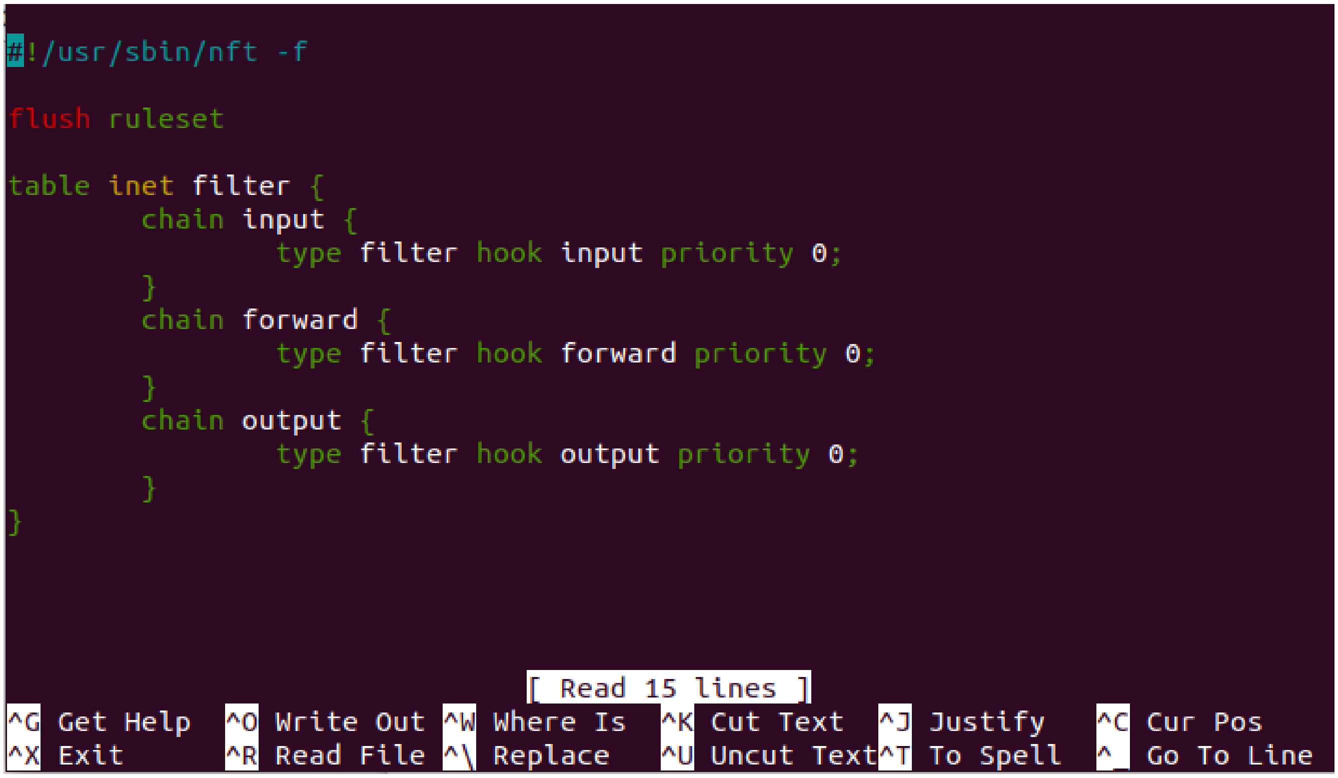
In the iptables structure, there are four tools per family:

- iptables
- ip6tables
- arptables
- ebtables

nftables contain a compatibility layer that encompasses all of these tools, which allows the use of the old iptables rules syntax.

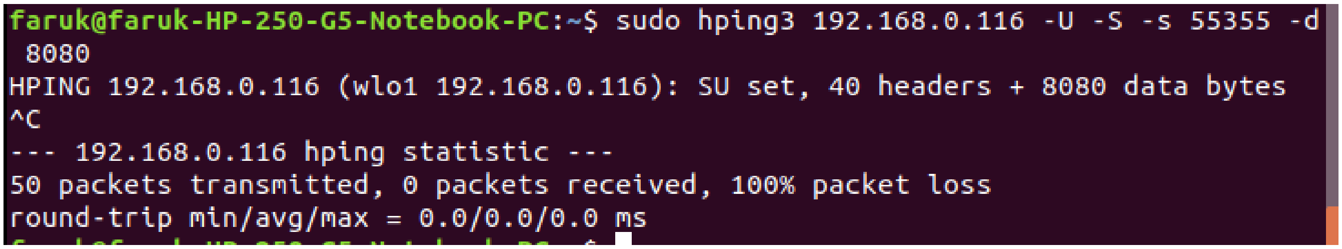
Nftables is a framework by the netfilter project that provides packet filtering,network address translation(NAT) and other packet mangling.

Nftables replaces the iptables framework.



```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
        chain input {
                type filter hook input priority 0;
        }
        chain forward {
                type filter hook forward priority 0;
        }
        chain output {
                type filter hook output priority 0;
        }
}
```

33. hping3 : construct any TCP packet you want. Its a new version of hping,one of the most popular and free packet crafting tool avialable. Hping3 tools help to create any type of packet that we need inorder to launch an attack, as different packets will results in different responses from the OS TCP/IP stack,gives an idea about the ports and service and the IDS and firewall testing using different protocol fragmenting packets etc.
Hping3 can create TCP, RAWIP, ICMP and UDP packets.However,its default packet is TCP.
Most network admins block or drop ICMP packets for security reasons,But with the help of hping3 tools,we can do almost the same job with the tcp protocol.
hping3 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies.



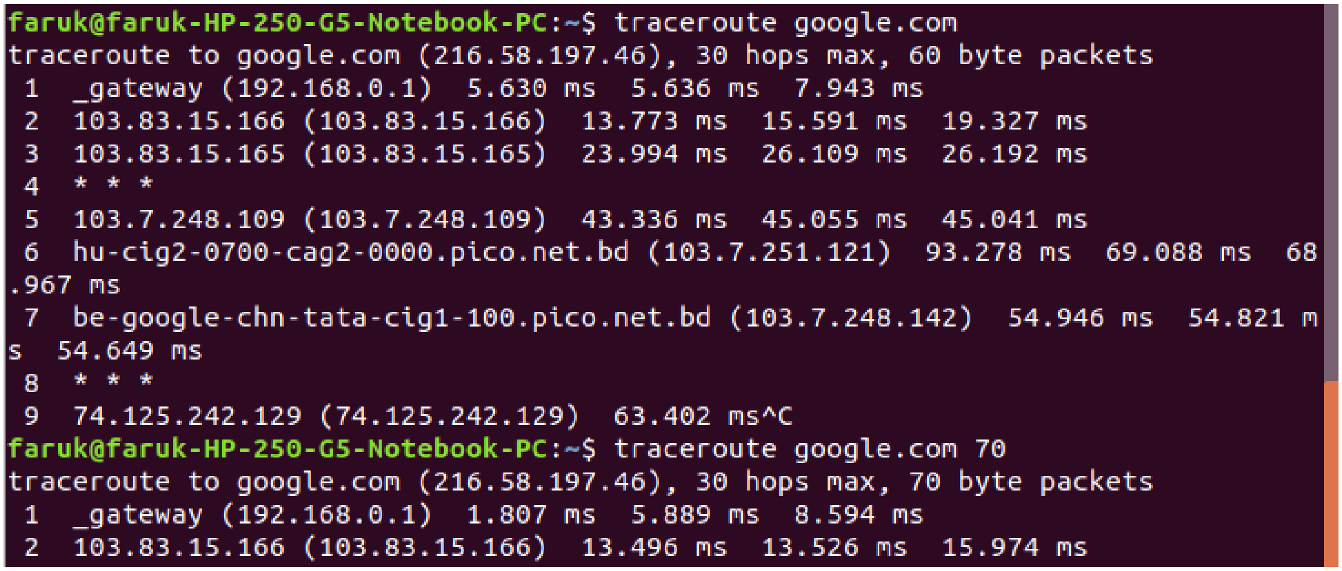
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo hping3 192.168.0.116 -U -S -s 55355 -d
 8080
HPING 192.168.0.116 (wlo1 192.168.0.116): SU set, 40 headers + 8080 data bytes
^C
--- 192.168.0.116 hping statistic ---
50 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

34. traceroute/mtr : what servers are on the way to that server?
Traceroute is a simple tool to show the pathway to a remote server. This can be anything from a website that you are attempting to visit, to a printer on your local network.
To call it, we simply need to provide a website or IP address that we would like to explore:

```
traceroute google.com
```

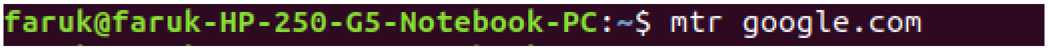
You can adjust the size of the packet that is sent to each hop by giving the integer after the hostname:

```
traceroute google.com 70
```

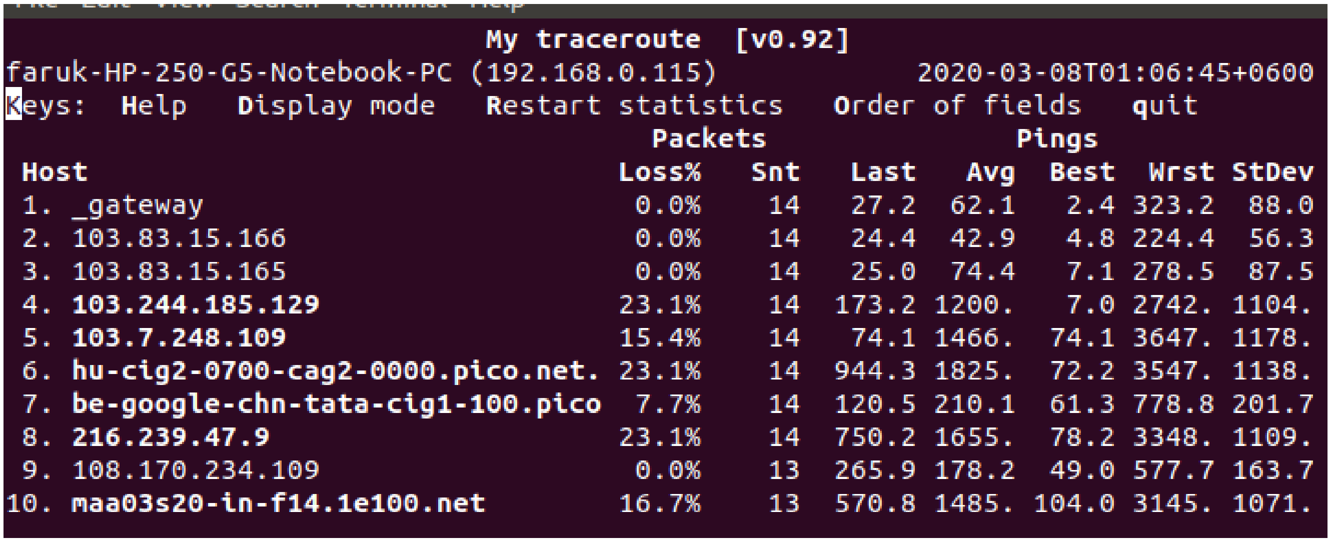


```
faruk@faruk-HP-250-G5-Notebook-PC:~$ traceroute google.com
traceroute to google.com (216.58.197.46), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  5.630 ms  5.636 ms  7.943 ms
 2  103.83.15.166 (103.83.15.166)  13.773 ms  15.591 ms  19.327 ms
 3  103.83.15.165 (103.83.15.165)  23.994 ms  26.109 ms  26.192 ms
 4  * * *
 5  103.7.248.109 (103.7.248.109)  43.336 ms  45.055 ms  45.041 ms
 6  hu-cig2-0700-cag2-0000.pico.net.bd (103.7.251.121)  93.278 ms  69.088 ms  68
.967 ms
 7  be-google-chn-tata-cig1-100.pico.net.bd (103.7.248.142)  54.946 ms  54.821 m
s  54.649 ms
 8  * * *
 9  74.125.242.129 (74.125.242.129)  63.402 ms^C
faruk@faruk-HP-250-G5-Notebook-PC:~$ traceroute google.com 70
traceroute to google.com (216.58.197.46), 30 hops max, 70 byte packets
 1  _gateway (192.168.0.1)  1.807 ms  5.889 ms  8.594 ms
 2  103.83.15.166 (103.83.15.166)  13.496 ms  13.526 ms  15.974 ms
```

Mtr(my traceroute) is a command line network diagnostic tool that provides the functionality of both the ping and traceroute commands. It is a simple and cross-platform tool that prints information about the entire route that the network packets take, right from the host system to the specified destination system. The mtr command takes an edge over the traceroute command as it also prints the response percentage and the response times for all network hops between the two systems.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ mtr google.com
```



```
                          My traceroute  [v0.92]
faruk-HP-250-G5-Notebook-PC (192.168.0.115)           2020-03-08T01:06:45+0600
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                       Packets               Pings
 Host                                Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. _gateway                          0.0%    14   27.2  62.1   2.4 323.2  88.0
 2. 103.83.15.166                     0.0%    14   24.4  42.9   4.8 224.4  56.3
 3. 103.83.15.165                     0.0%    14   25.0  74.4   7.1 278.5  87.5
 4. 103.244.185.129                  23.1%    14  173.2 1200.   7.0 2742. 1104.
 5. 103.7.248.109                    15.4%    14   74.1 1466.  74.1 3647. 1178.
 6. hu-cig2-0700-cag2-0000.pico.net. 23.1%    14  944.3 1825.  72.2 3547. 1138.
 7. be-google-chn-tata-cig1-100.pico  7.7%    14  120.5 210.1  61.3 778.8 201.7
 8. 216.239.47.9                     23.1%    14  750.2 1655.  78.2 3348. 1109.
 9. 108.170.234.109                   0.0%    13  265.9 178.2  49.0 577.7 163.7
10. maa03s20-in-f14.1e100.net        16.7%    13  570.8 1485. 104.0 3145. 1071.
```

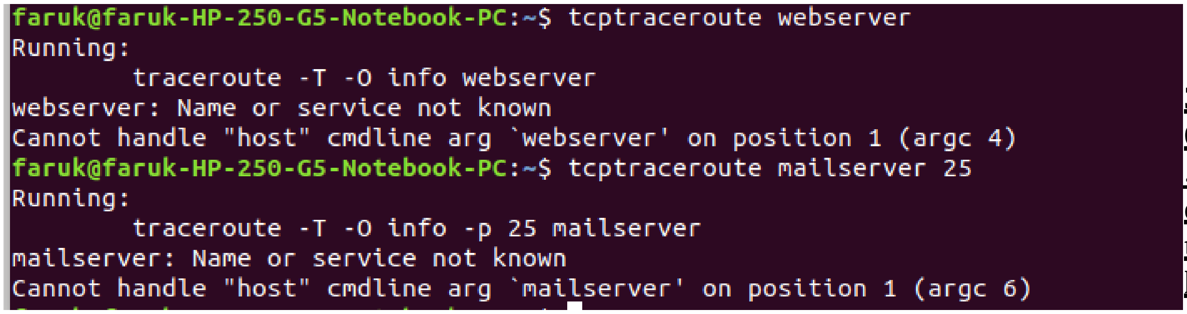
35. tcptraceroute : use tcp packets instead of icmp to traceroute . **tcptraceroute** is a traceroute implementation using TCP packets. **t**raceroute is a tool used to identify the path used by a packet to reach the destination. This tool uses ICMP messages, but unlike *ping*, identifies every router in the path. traceroute is useful when troubleshooting network problems.

To trace the path to a web server listening for connections on port 80:

tcptraceroute webserver

To trace the path to a mail server listening for connections on port 25:

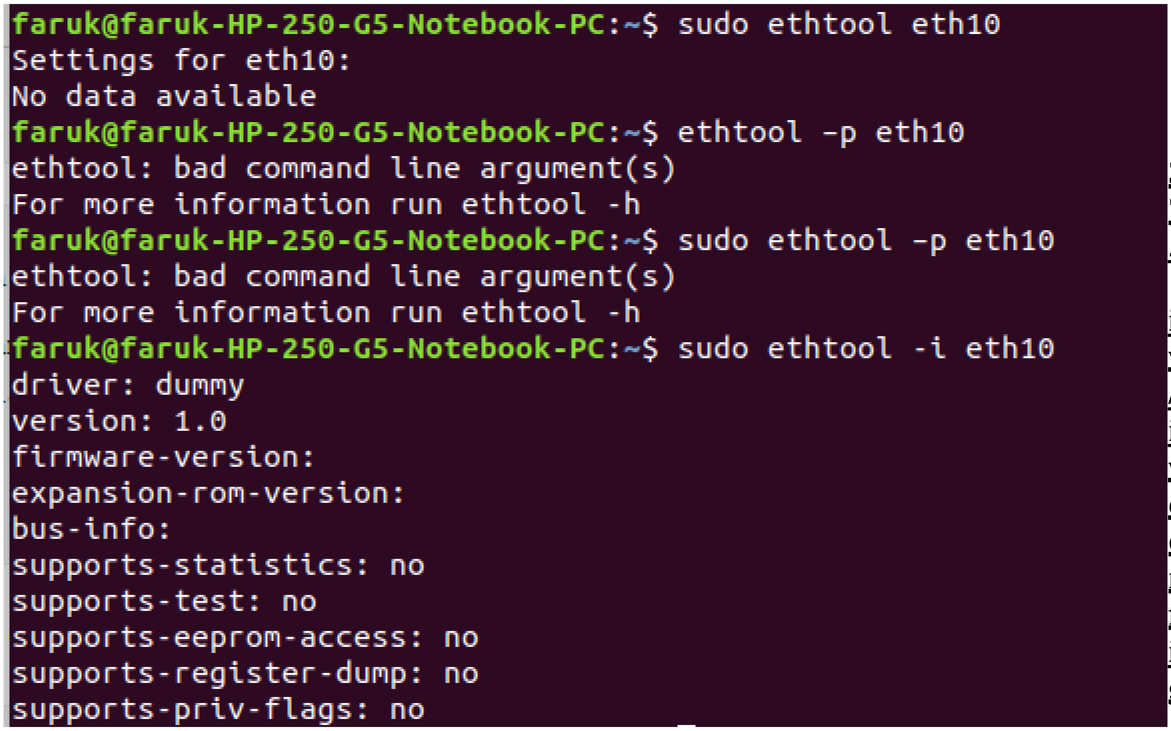
tcptraceroute mailserver 25



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ tcptraceroute webserver
Running:
        traceroute -T -O info webserver
webserver: Name or service not known
Cannot handle "host" cmdline arg `webserver' on position 1 (argc 4)
faruk@faruk-HP-250-G5-Notebook-PC:~$ tcptraceroute mailserver 25
Running:
        traceroute -T -O info -p 25 mailserver
mailserver: Name or service not known
Cannot handle "host" cmdline arg `mailserver' on position 1 (argc 6)
```

36. ethtool : manage physical ethernet connections + network cards. **ethtool** is used to query and control network device driver and hardware settings, particularly for wired Ethernet devices.

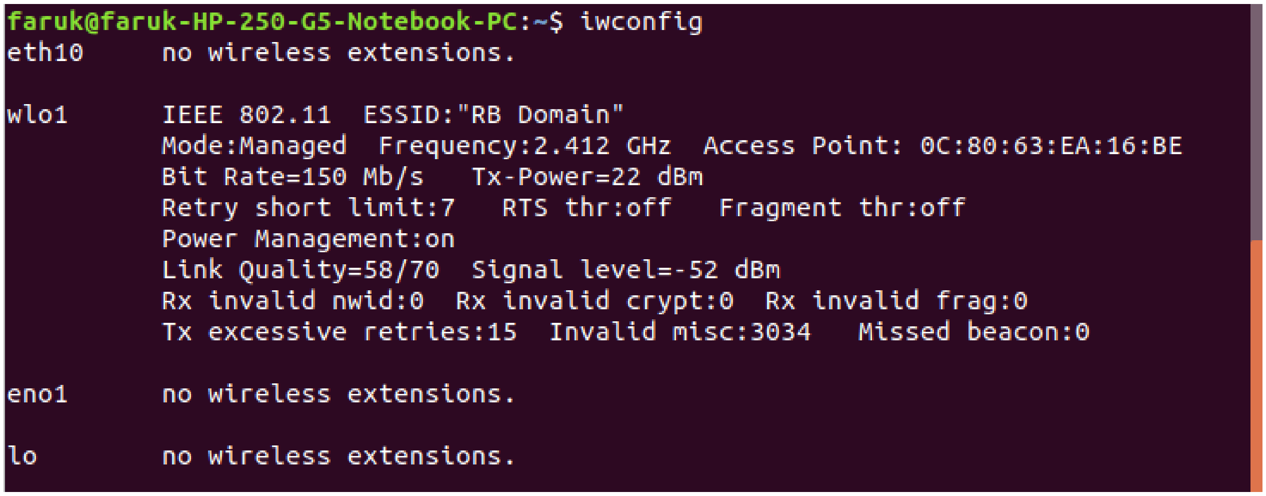
ethtool is a networking utility on Linux. It is used to configure Ethernet devices on Linux. **ethtool** can also be used to find a lot of information about connected Ethernet devices on your Linux computer.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo ethtool eth10
Settings for eth10:
No data available
faruk@faruk-HP-250-G5-Notebook-PC:~$ ethtool -p eth10
ethtool: bad command line argument(s)
For more information run ethtool -h
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo ethtool -p eth10
ethtool: bad command line argument(s)
For more information run ethtool -h
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo ethtool -i eth10
driver: dummy
version: 1.0
firmware-version:
expansion-rom-version:
bus-info:
supports-statistics: no
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

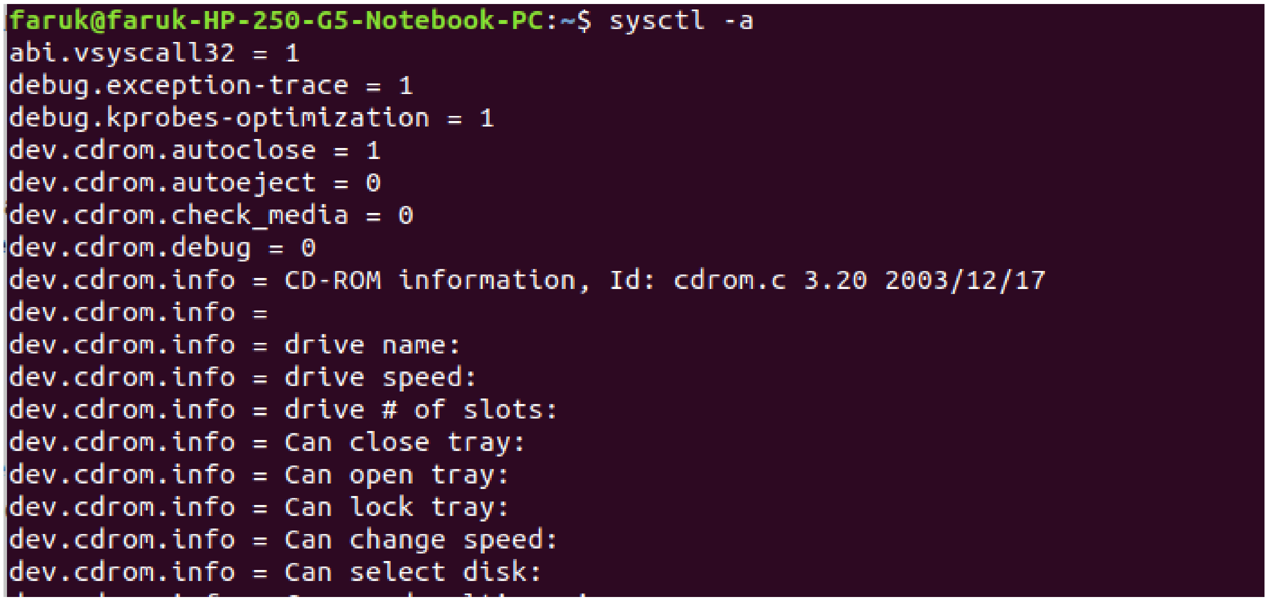
37. iw/iwconfig : manage wireless network settings.(see speed/frequency) . **Iwconfig** is similar to ***ifconfig***, but is dedicated to the wireless interfaces. It is used to set the parameters of the network interface which are specific to the wireless operation (for example : the frequency). **Iwconfig** may also be used to display those parameters, and the wireless statistics (extracted from */proc/net/wireless*) .

Below command is used to display all the wireless interfaces.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ iwconfig
eth10     no wireless extensions.

wlo1      IEEE 802.11  ESSID:"RB Domain"
          Mode:Managed  Frequency:2.412 GHz  Access Point: 0C:80:63:EA:16:BE
          Bit Rate=150 Mb/s   Tx-Power=22 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=58/70  Signal level=-52 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:15  Invalid misc:3034   Missed beacon:0

eno1      no wireless extensions.

lo        no wireless extensions.
```

38. sysctl : configure Linux kernel's network stack. **sysctl** is used to modify kernel parameters at runtime. The parameters available are those listed under /proc/sys/. Procfs is required for **sysctl(8)** support in Linux. You can use **sysctl(8)** to both read and write sysctl data.



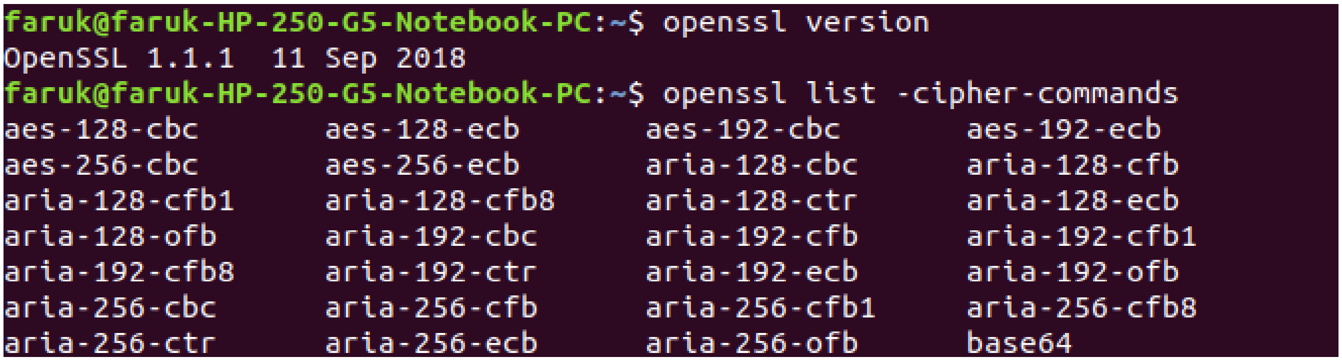
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ sysctl -a
abi.vsyscall32 = 1
debug.exception-trace = 1
debug.kprobes-optimization = 1
dev.cdrom.autoclose = 1
dev.cdrom.autoeject = 0
dev.cdrom.check_media = 0
dev.cdrom.debug = 0
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info =
dev.cdrom.info = drive name:
dev.cdrom.info = drive speed:
dev.cdrom.info = drive # of slots:
dev.cdrom.info = Can close tray:
dev.cdrom.info = Can open tray:
dev.cdrom.info = Can lock tray:
dev.cdrom.info = Can change speed:
dev.cdrom.info = Can select disk:
```

39. openssl : do literally anything with SSL certificates. **OpenSSL** is an open-source **command** line tool that is commonly used to generate private keys, create CSRs, install your SSL/TLS certificate, and identify certificate information.

The **openssl** program is a command line tool for using the various cryptography functions of OpenSSL's **crypto** library from the shell. It can be used for

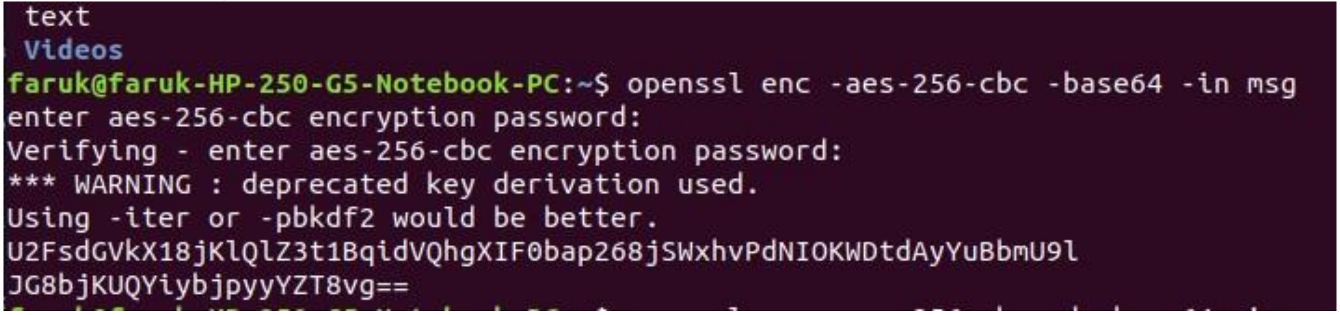
- Creation and management of private keys, public keys and parameters
- Public key cryptographic operations
- Creation of X.509 certificates, CSRs and CRLs
- Calculation of Message Digests

- Encryption and Decryption with Ciphers
- SSL/TLS Client and Server Tests

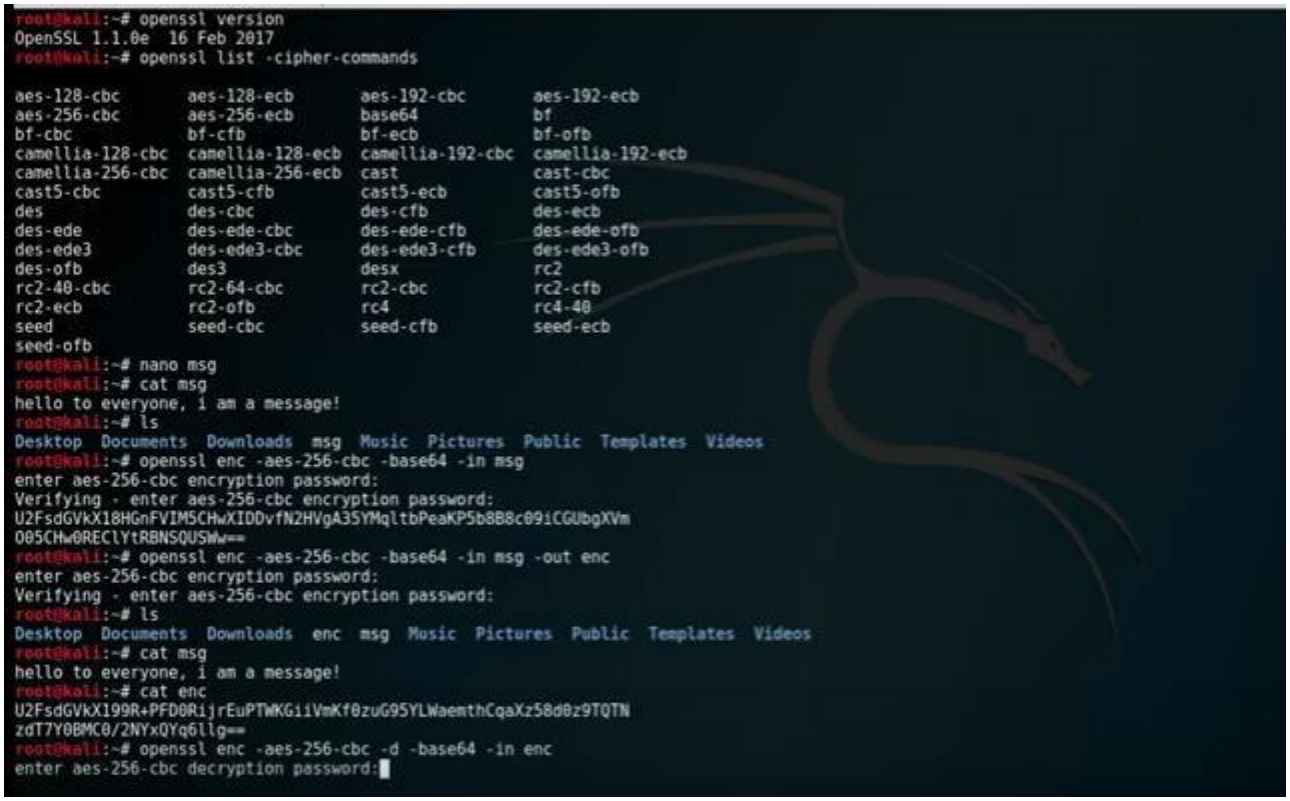


```
faruk@faruk-HP-250-G5-Notebook-PC:~$ openssl version
OpenSSL 1.1.1  11 Sep 2018
faruk@faruk-HP-250-G5-Notebook-PC:~$ openssl list -cipher-commands
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      base64
```

then type **nano msg** and press ctrl+X and yes and enter
then type **cat msg**   then type ls



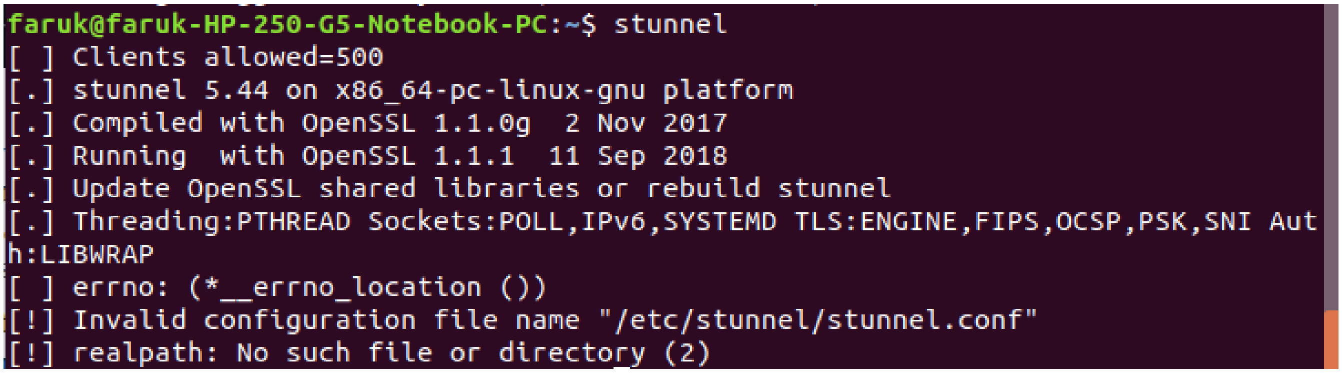
```
 text
 Videos
faruk@faruk-HP-250-G5-Notebook-PC:~$ openssl enc -aes-256-cbc -base64 -in msg
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
U2FsdGVkX18jKlQlZ3t1BqidVQhgXIF0bap268jSWxhvPdNIOKWDtdAyYuBbmU9l
JG8bjKUQYiybjpyyYZT8vg==
```



```
root@kali:~# openssl version
OpenSSL 1.1.0e  16 Feb 2017
root@kali:~# openssl list -cipher-commands

aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       base64            bf
bf-cbc            bf-cfb            bf-ecb            bf-ofb
camellia-128-cbc  camellia-128-ecb  camellia-192-cbc  camellia-192-ecb
camellia-256-cbc  camellia-256-ecb  cast              cast-cbc
cast5-cbc         cast5-cfb         cast5-ecb         cast5-ofb
des               des-cbc           des-cfb           des-ecb
des-ede           des-ede-cbc       des-ede-cfb       des-ede-ofb
des-ede3          des-ede3-cbc      des-ede3-cfb      des-ede3-ofb
des-ofb           des3              desx              rc2
rc2-40-cbc        rc2-64-cbc        rc2-cbc           rc2-cfb
rc2-ecb           rc2-ofb           rc4               rc4-40
seed              seed-cbc          seed-cfb          seed-ecb
seed-ofb
root@kali:~# nano msg
root@kali:~# cat msg
hello to everyone, i am a message!
root@kali:~# ls
Desktop  Documents  Downloads  msg  Music  Pictures  Public  Templates  Videos
root@kali:~# openssl enc -aes-256-cbc -base64 -in msg
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
U2FsdGVkX18HGnFVIM5CHwXIDDvfN2HVgA35YMqltbPeaKP5b8B8c09iCGUbgXVm
O05CHw0RECLYtRBNSQUSWw==
root@kali:~# openssl enc -aes-256-cbc -base64 -in msg -out enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
root@kali:~# ls
Desktop  Documents  Downloads  enc  msg  Music  Pictures  Public  Templates  Videos
root@kali:~# cat msg
hello to everyone, i am a message!
root@kali:~# cat enc
U2FsdGVkX199R+PFD0RijrEuPTWKGiiVmKf0zuG95YLWaemthCqaXz58d0z9TQTN
zdT7Y0BMC0/2NYxQYq6llg==
root@kali:~# openssl enc -aes-256-cbc -d -base64 -in enc
enter aes-256-cbc decryption password:
```

40. stunnel : make a SSL proxy for an insecure server . The **stunnel** program is designed to work as SSL encryption wrapper between remote clients and local (*inetd*-startable) or remote servers. The concept is that having non-SSL aware daemons running on your system you can easily set them up to communicate with clients over secure SSL channels.

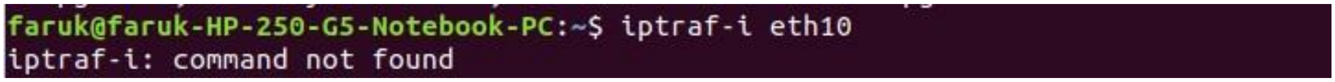
stunnel can be used to add SSL functionality to commonly used *Inetd* daemons like POP-2, POP-3, and IMAP servers, to standalone daemons like NNTP, SMTP and HTTP, and in tunneling PPP over network sockets without changes to the source code.



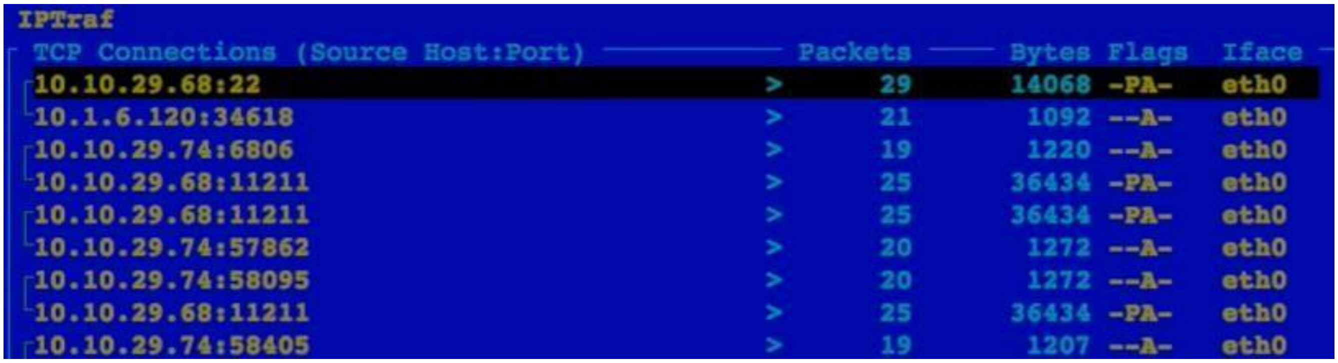
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ stunnel
[ ] Clients allowed=500
[.] stunnel 5.44 on x86_64-pc-linux-gnu platform
[.] Compiled with OpenSSL 1.1.0g  2 Nov 2017
[.] Running  with OpenSSL 1.1.1  11 Sep 2018
[.] Update OpenSSL shared libraries or rebuild stunnel
[.] Threading:PTHREAD Sockets:POLL,IPv6,SYSTEMD TLS:ENGINE,FIPS,OCSP,PSK,SNI Aut
h:LIBWRAP
[ ] errno: (*__errno_location ())
[!] Invalid configuration file name "/etc/stunnel/stunnel.conf"
[!] realpath: No such file or directory (2)
```

41. iptraf/nethogs/iftop/ntop : see what's using bandwidth . Iptraf is a very friendly console interactive tool to monitor traffic statistics including many customization options, together with vnstat it is the most complete tool shown in this tutorial.

Iptraf -i eth10 --→immediately start the IP traffic monitor on the specified interface.



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ iptraf-i eth10
iptraf-i: command not found
```



```
IPTraf
┌ TCP Connections (Source Host:Port) ──────────── Packets ──── Bytes Flags  Iface ─
┌10.10.29.68:22                              >        29     14068 -PA-   eth0
└10.1.6.120:34618                            >        21      1092 --A-   eth0
┌10.10.29.74:6806                            >        19      1220 --A-   eth0
└10.10.29.68:11211                           >        25     36434 -PA-   eth0
┌10.10.29.68:11211                           >        25     36434 -PA-   eth0
└10.10.29.74:57862                           >        20      1272 --A-   eth0
┌10.10.29.74:58095                           >        20      1272 --A-   eth0
└10.10.29.68:11211                           >        25     36434 -PA-   eth0
┌10.10.29.74:58405                           >        19      1207 --A-   eth0
```

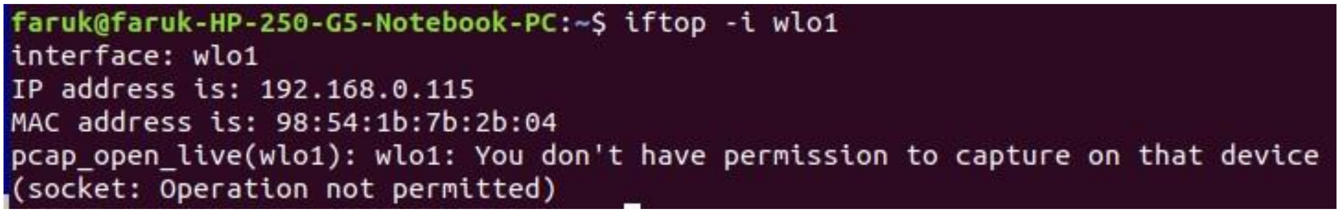
Iftop is another tool available on Debian and Ubuntu Linux distributions repositories, to install it run *apt install iftop*

To launch iftop use the -i option to specify the network interface:

iftop -i <interface>

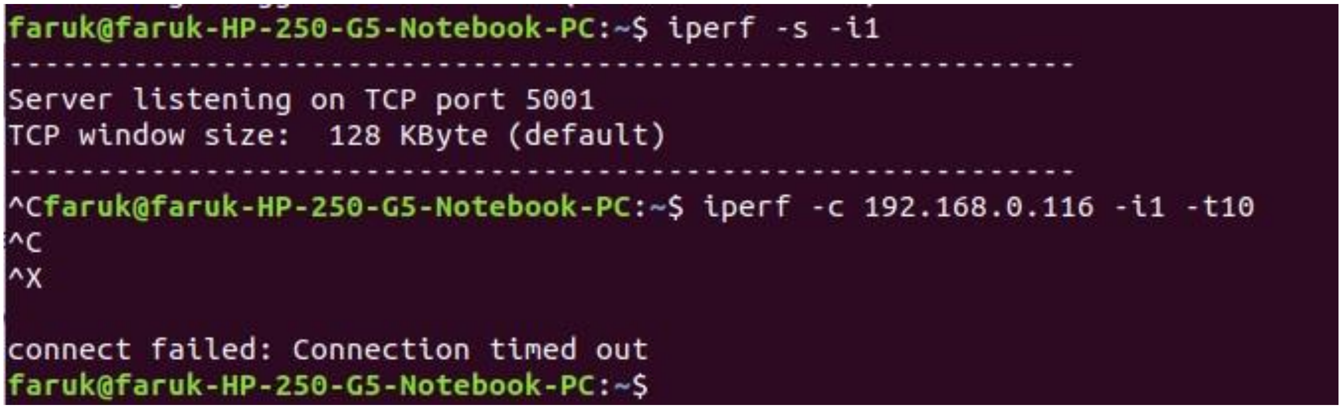
In my case:

iftop -i wlp3s0



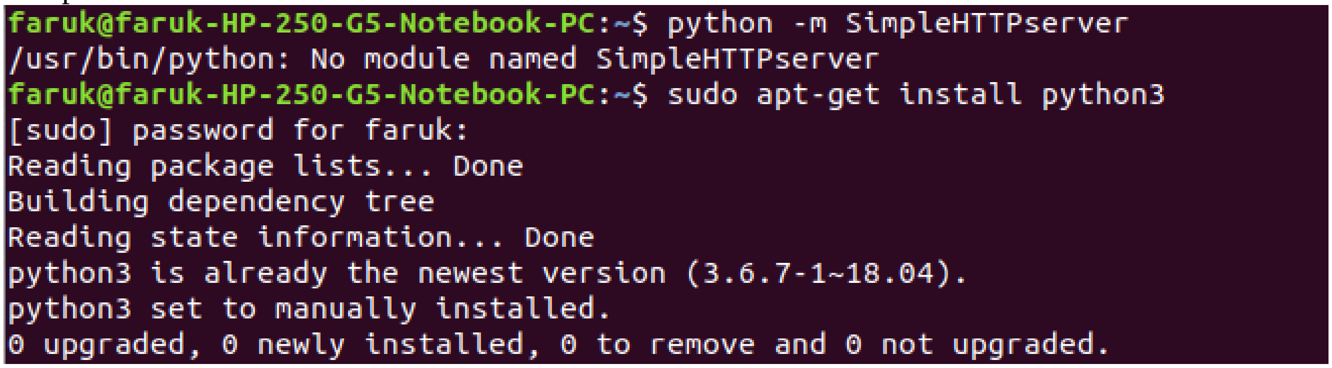
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ iftop -i wlo1
interface: wlo1
IP address is: 192.168.0.115
MAC address is: 98:54:1b:7b:2b:04
pcap_open_live(wlo1): wlo1: You don't have permission to capture on that device
(socket: Operation not permitted)
```

<u>42.</u> **<u>ab/nload/iperf :</u>** benchmarking tools . Iperf is an open source networking tool used to measure throughput or performance of a network. It can be used to test TCP and UDP.



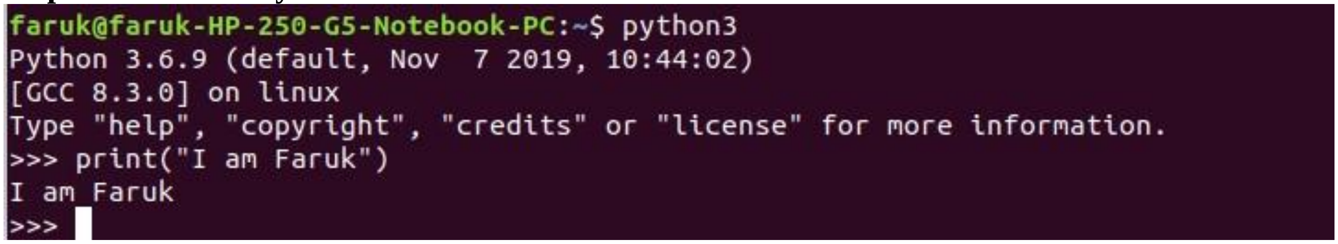
```
faruk@faruk-HP-250-G5-Notebook-PC:~$ iperf -s -i1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
^Cfaruk@faruk-HP-250-G5-Notebook-PC:~$ iperf -c 192.168.0.116 -i1 -t10
^C
^X

connect failed: Connection timed out
faruk@faruk-HP-250-G5-Notebook-PC:~$
```

<u>43.</u> **<u>python -m SimpleHttpserver :</u>** serve files from a directory. SimpleHTTPServer is a python module which allows you to instantly create a web server or serve your files in a snap. Main advantage of python’s SimpleHTTPServer is you don’t need to install Apache or Nginx, since you have python interpreter installed.

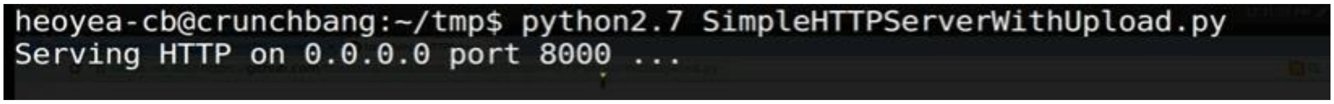


```
faruk@faruk-HP-250-G5-Notebook-PC:~$ python -m SimpleHTTPserver
/usr/bin/python: No module named SimpleHTTPserver
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo apt-get install python3
[sudo] password for faruk:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.6.7-1~18.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Its possible to run Python code on terminal



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ python3
Python 3.6.9 (default, Nov  7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("I am Faruk")
I am Faruk
>>>
```



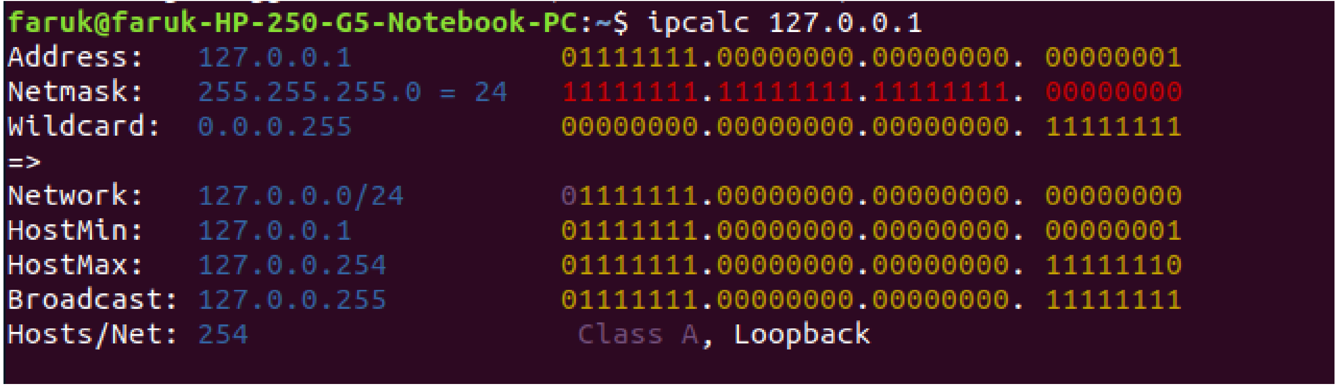
```
heoyea-cb@crunchbang:~/tmp$ python2.7 SimpleHTTPServerWithUpload.py
Serving HTTP on 0.0.0.0 port 8000 ...
```

<u>44.</u> <u>ipcalc :</u> easily see what 13.21.2.3/25 means . **ipcalc** provides a simple way to calculate IP information for a host.

Directory listing for /

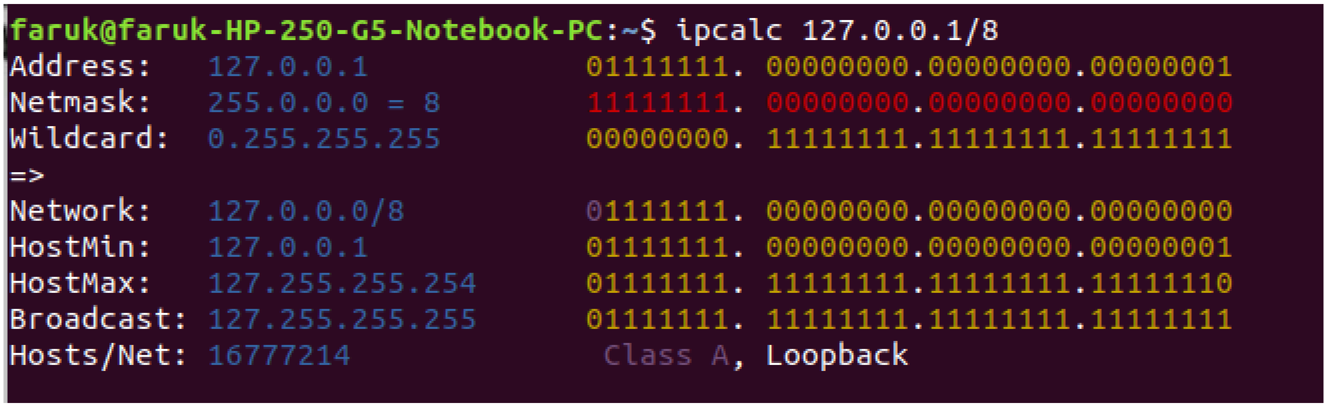
- 1st_hokage.jpg

Run **ipcalc** with your IP address to see everything you need to know:



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ ipcalc 127.0.0.1
Address:   127.0.0.1            01111111.00000000.00000000. 00000001
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   127.0.0.0/24         01111111.00000000.00000000. 00000000
HostMin:   127.0.0.1            01111111.00000000.00000000. 00000001
HostMax:   127.0.0.254          01111111.00000000.00000000. 11111110
Broadcast: 127.0.0.255          01111111.00000000.00000000. 11111111
Hosts/Net: 254                   Class A, Loopback
```

<u>Calculate a subnet for 192.168.1.0/24 :</u>



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ ipcalc 127.0.0.1/8
Address:   127.0.0.1            01111111. 00000000.00000000.00000001
Netmask:   255.0.0.0 = 8        11111111. 00000000.00000000.00000000
Wildcard:  0.255.255.255        00000000. 11111111.11111111.11111111
=>
Network:   127.0.0.0/8          01111111. 00000000.00000000.00000000
HostMin:   127.0.0.1            01111111. 00000000.00000000.00000001
HostMax:   127.255.255.254      01111111. 11111111.11111111.11111110
Broadcast: 127.255.255.255      01111111. 11111111.11111111.11111111
Hosts/Net: 16777214              Class A, Loopback
```

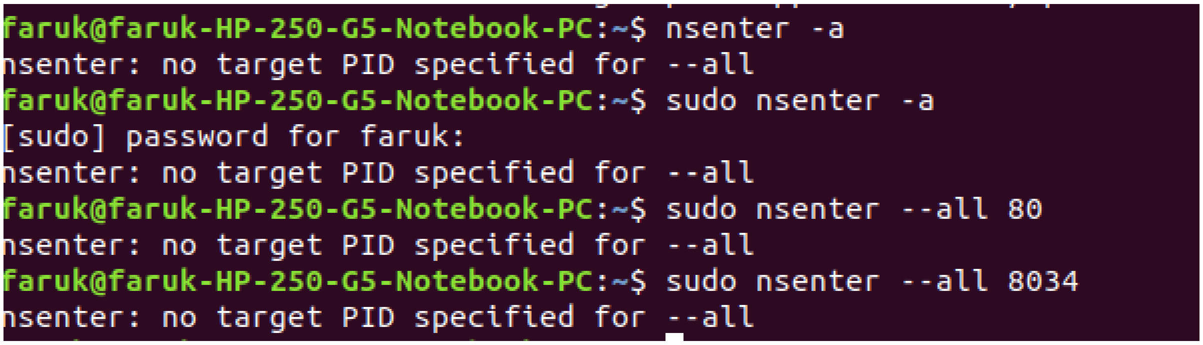
<u>45.</u> <u>nsenter:</u> enter a container process ‘s network namespace . run program with namespaces of other processes.

-a, --all

Enter all namespaces of the target process by the default

/proc/[pid]/ns/* namespace paths.

Syntax : nsenter [options] [program [arguments]]



```
faruk@faruk-HP-250-G5-Notebook-PC:~$ nsenter -a
nsenter: no target PID specified for --all
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo nsenter -a
[sudo] password for faruk:
nsenter: no target PID specified for --all
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo nsenter --all 80
nsenter: no target PID specified for --all
faruk@faruk-HP-250-G5-Notebook-PC:~$ sudo nsenter --all 8034
nsenter: no target PID specified for --all
```

Conclusion :

From doing this assignment, I learn all networking tools like ping,ipconfig, tracert,nslookup etc. When I do this assignment , l become very cheerful. Beacasue This is my first networking assignment. My Course teacher Nazrul Islam sir helps me a lot to do this assignment.