Experiment No. – 1			
Date of Performance:			
Date of Submission:			
Program Execution/ formation/ correction/ ethical practices (06)	Timely Submission (01) Viva (03)	Experiment Total (10)	Sign with Date

Experiment No. 1
Network reconnaissance tools

1.1 **Aim:** To use basic networking commands in Linux (ping, tracert, nslookup, netstat, ARP, RARP, ip, ifconfig, dig, route).

1.2 **Course Outcome:** Explain the need for Cyber Security and its aspects.

1.3 **Learning Objectives:** Explain the various commands involved in network reconnaissance.

1.4 **Requirement:** Kali Linux

1.5 **Related Theory:**

ifconfig command:
You can use the **ifconfig** command to assign an address to a network interface and to configure or display the current network interface configuration information. The **ifconfig** command must be used at system startup to define the network address of each interface present on a system.

Netstat command:
The netstat command displays information regarding traffic on the configured network interfaces, such as the following:

- The address of any protocol control blocks associated with the sockets and the state of all sockets

- The number of packets received, transmitted, and dropped in the communications subsystem
- Cumulative statistics per interface
- Routes and their status

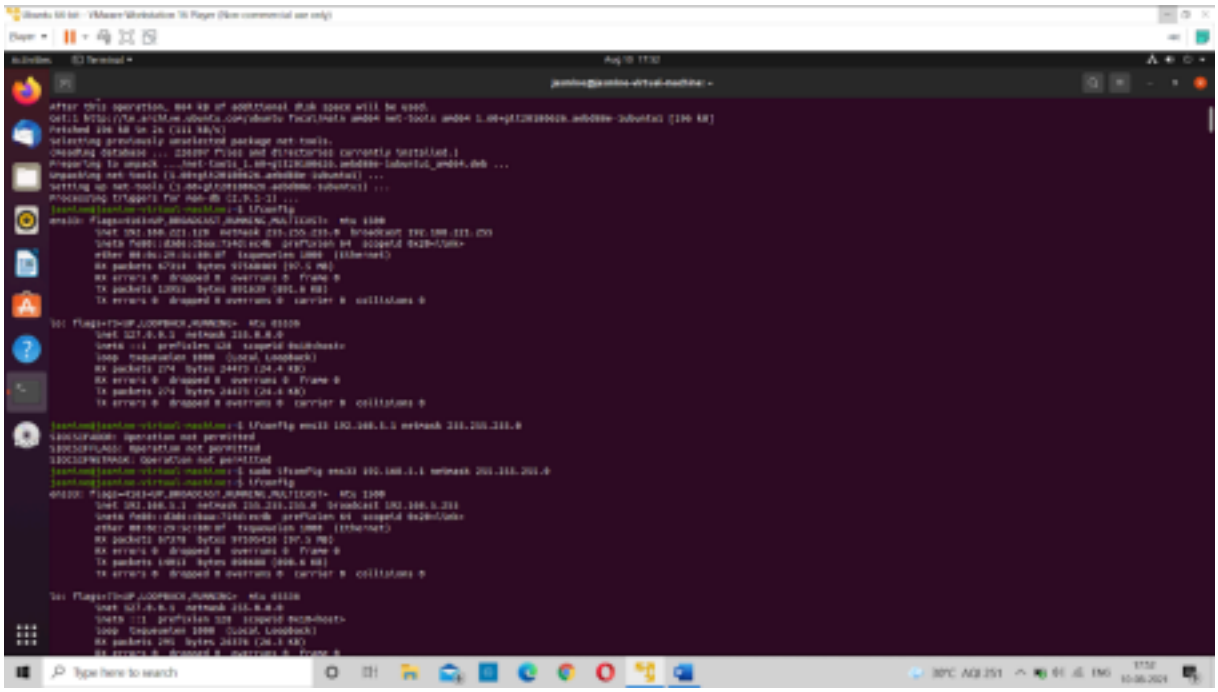


Figure 1.1 Implementation of ifconfig command

Assign the IP address and netmask

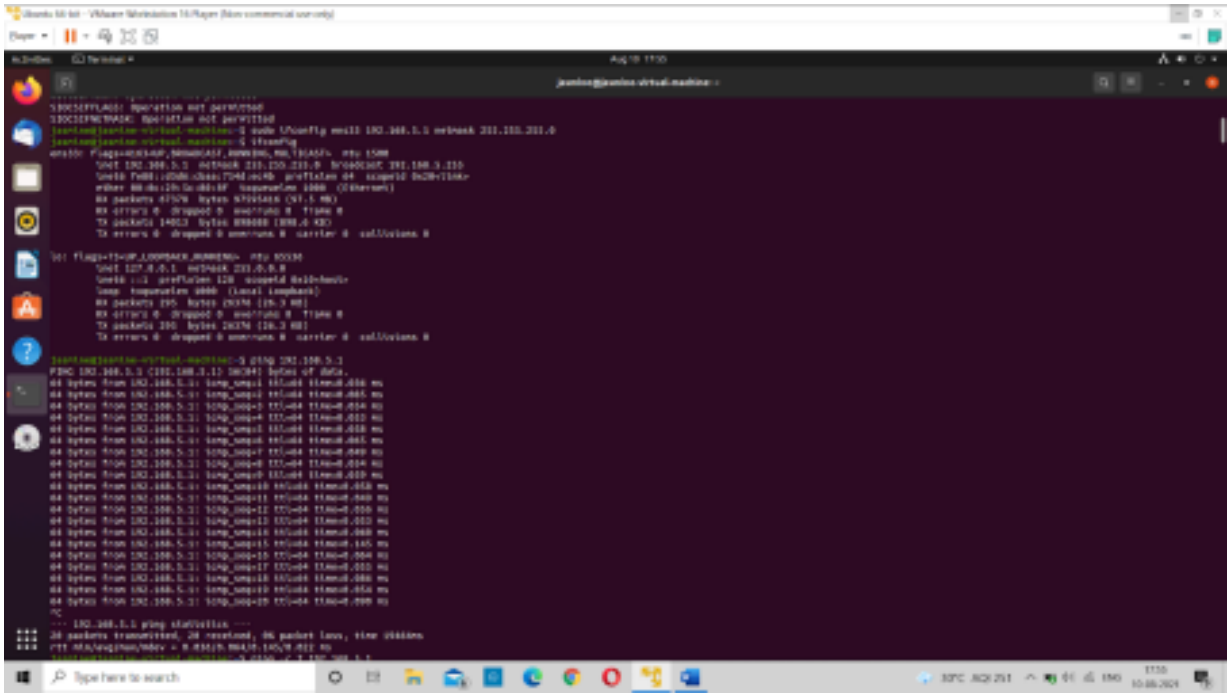


Figure 1.2 Assign the IP address and netmask

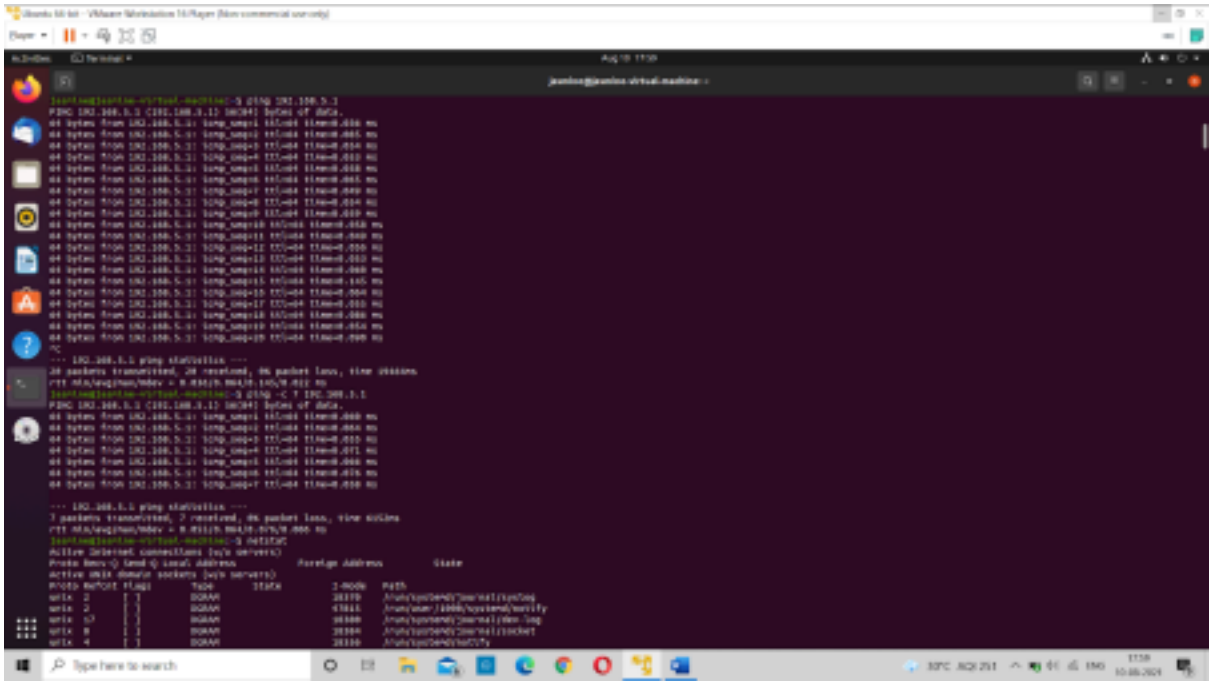


Figure 1.3 Netstat command

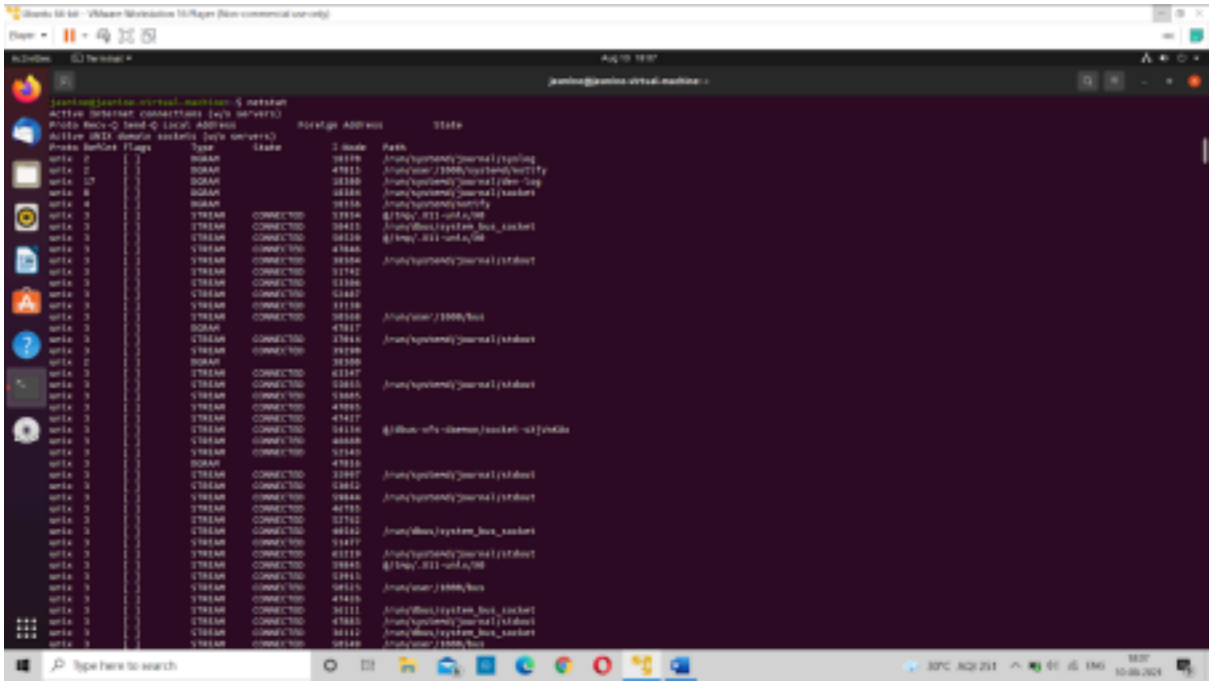


Figure 1.4 Netstat command

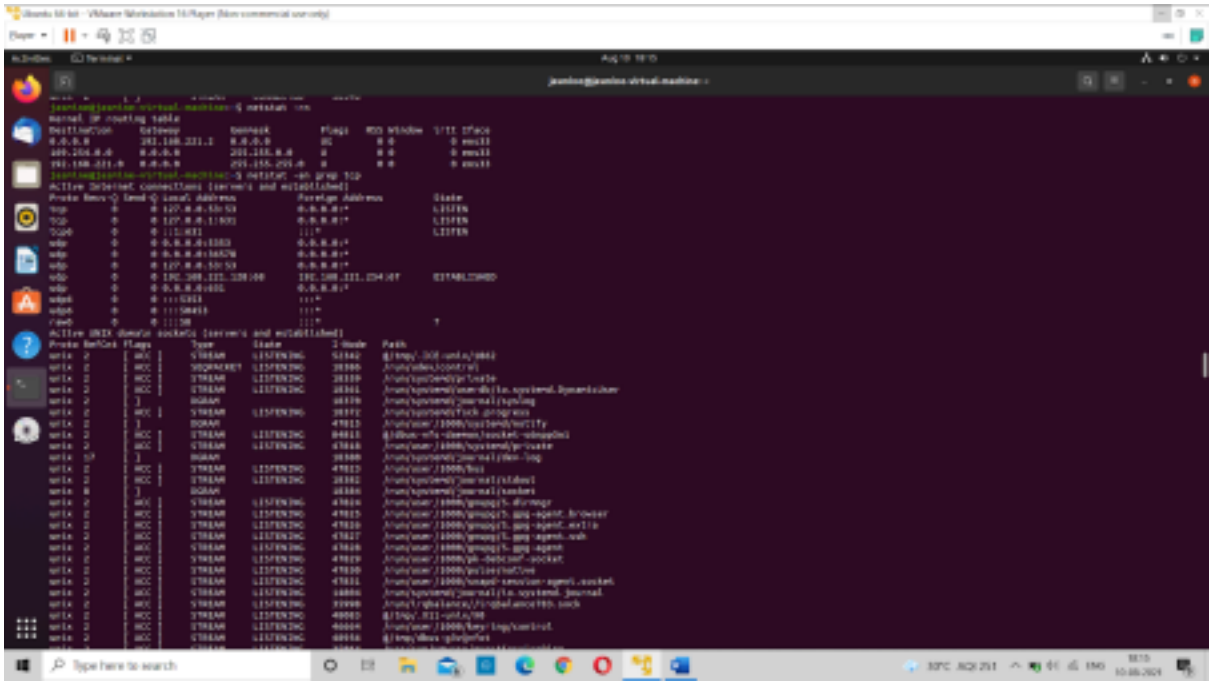


Figure 1.5 Netstat command

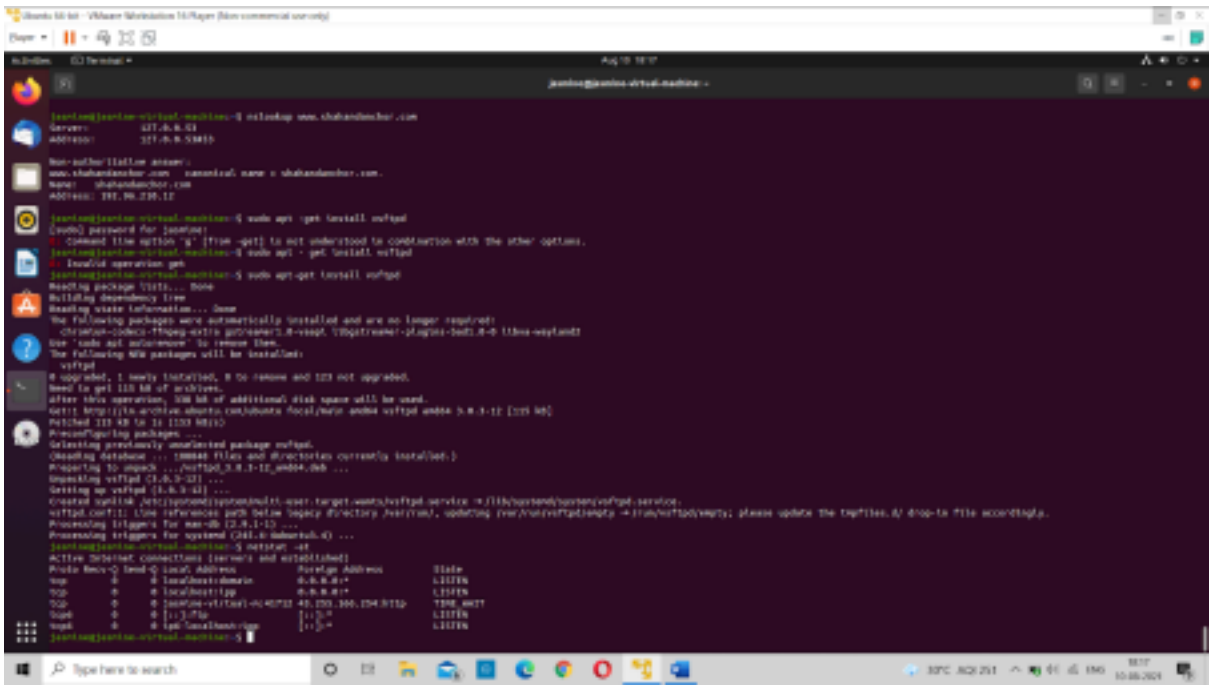


Figure 1.6 Netstat command

traceroute command

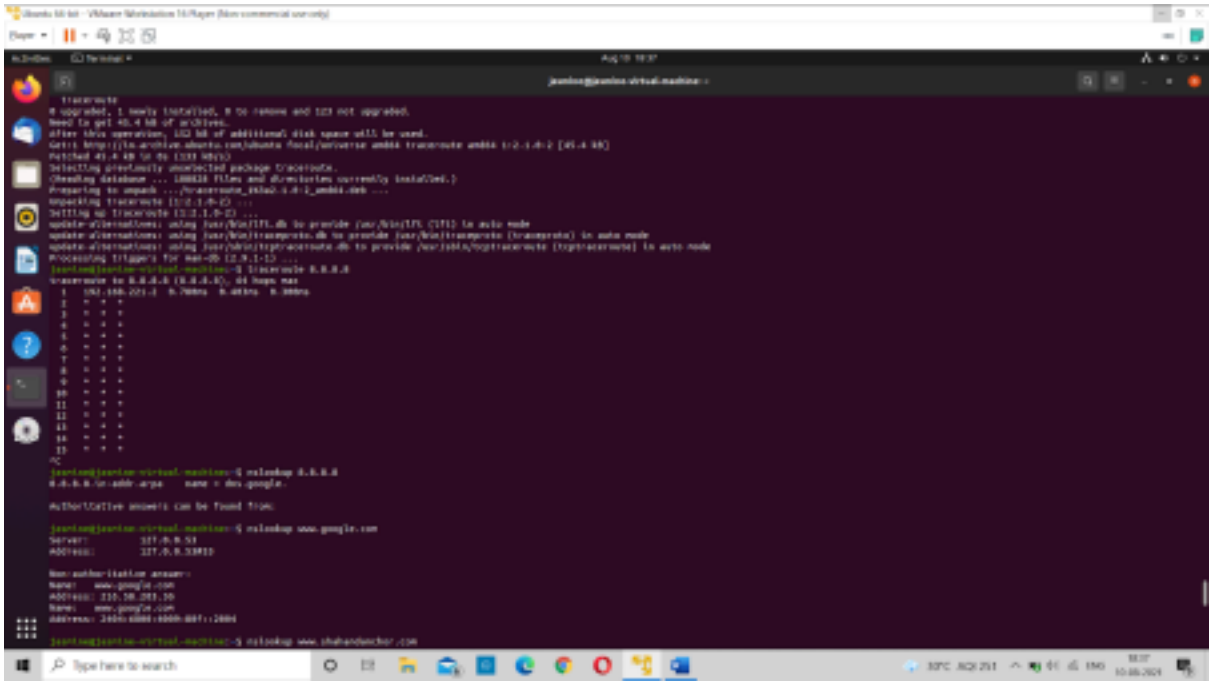


Figure 1.7 traceroute command

nslookup command:

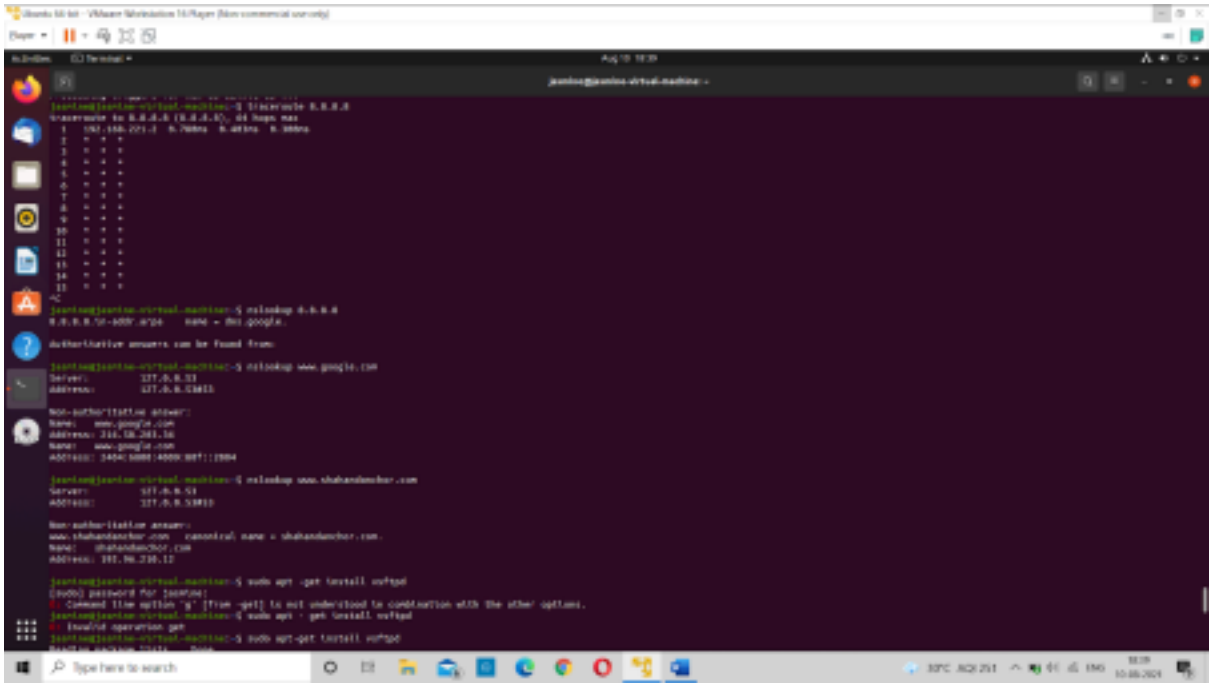


Figure 1.8 nslookup command

ARP:

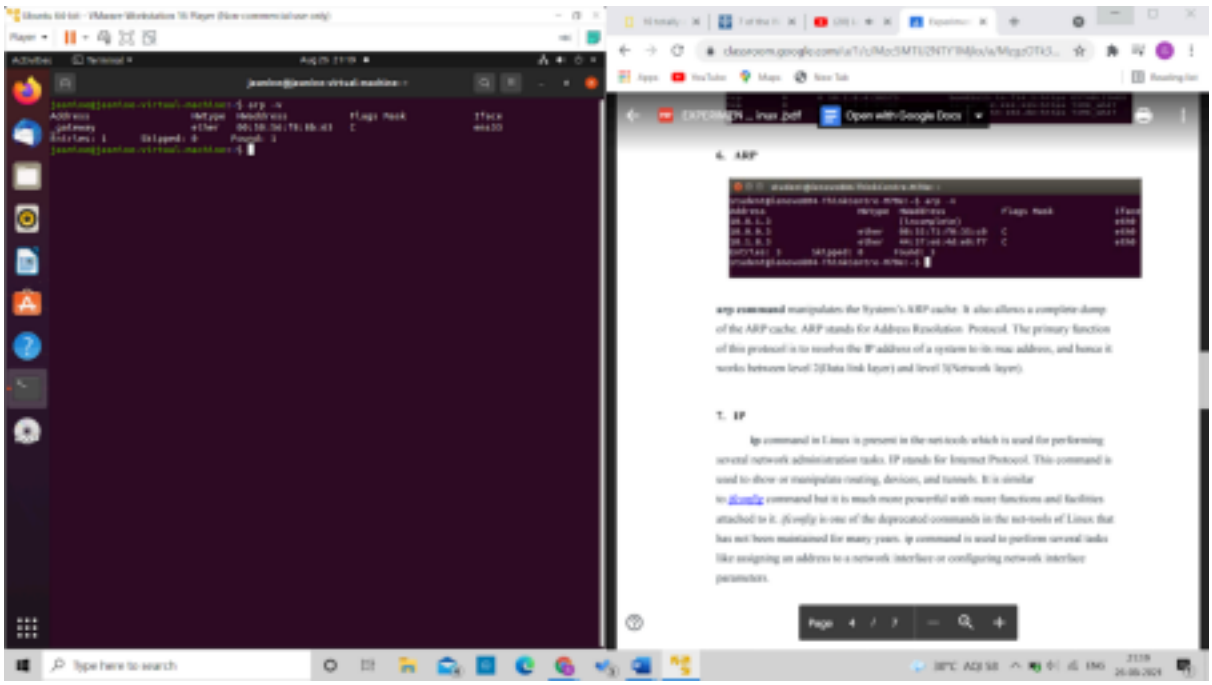


Figure 1.9 arp command

Dig:

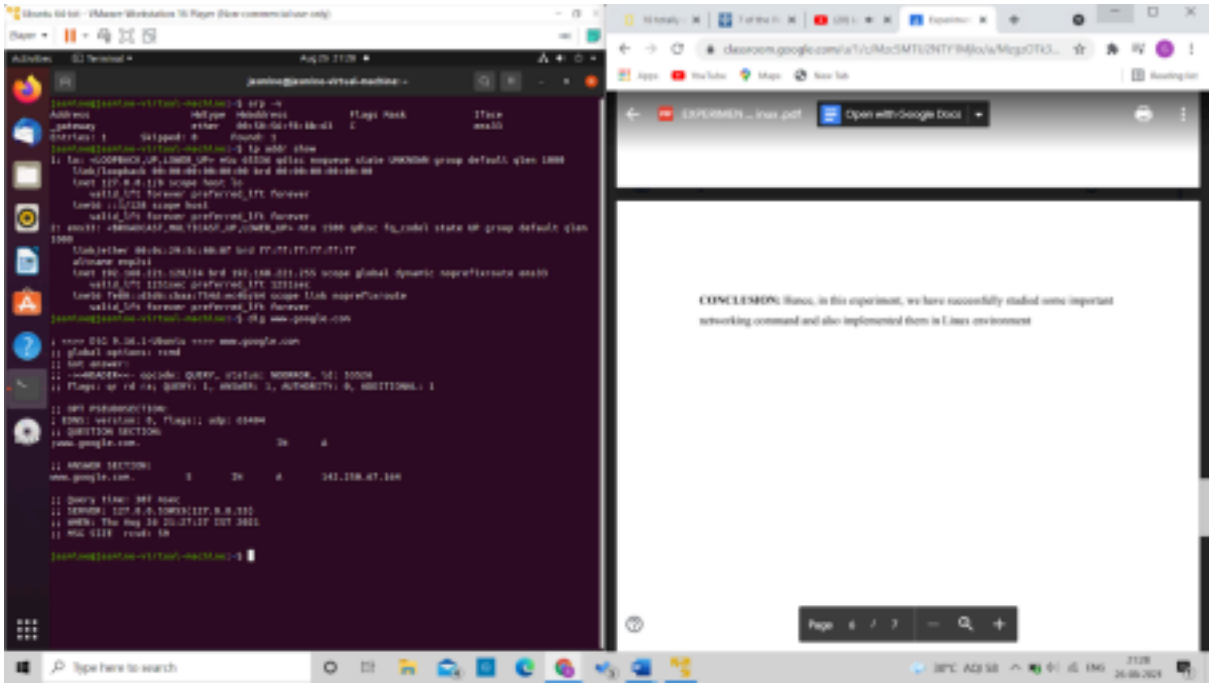


Figure 1.10 dig command

1.6 **Procedure:**

a) Execute all the commands listed above and observe the output.

b) By applying the variations in the above listed commands, note down the difference between them.

1.7 **Command and Output:**

Ifconfig:

```
┌──(root㉿kali)-[/home/kali]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.30.20  netmask 255.255.254.0  broadcast 172.16.31.255
        inet6 fe80::a00:27ff:fe81:720a  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:81:72:0a  txqueuelen 1000  (Ethernet)
        RX packets 270999  bytes 380768712 (363.1 MiB)
        RX errors 0  dropped 4  overruns 0  frame 0
        TX packets 64032  bytes 4683248 (4.4 MiB)
        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 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 74  bytes 6024 (5.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 6024 (5.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

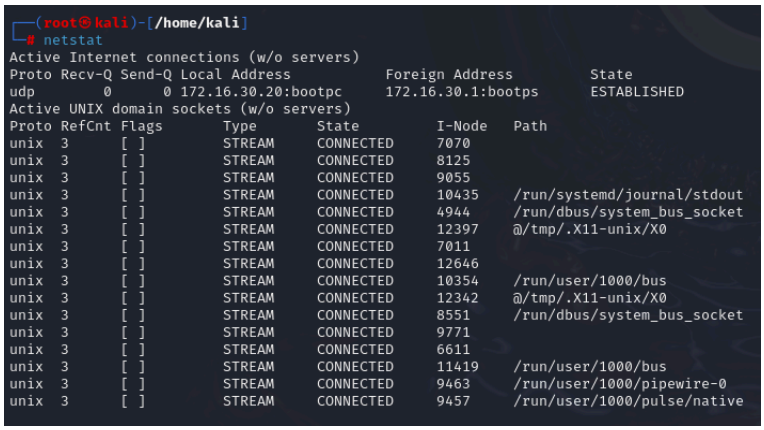
```
┌──(root㉿kali)-[/home/kali]
└─# ifconfig eth0 192.168.5.1 netmask 255.255.255.0

┌──(root㉿kali)-[/home/kali]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.5.1  netmask 255.255.255.0  broadcast 192.168.5.255
        inet6 fe80::a00:27ff:fe81:720a  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:81:72:0a  txqueuelen 1000  (Ethernet)
        RX packets 271399  bytes 380794407 (363.1 MiB)
        RX errors 0  dropped 4  overruns 0  frame 0
        TX packets 64032  bytes 4683248 (4.4 MiB)
        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 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 74  bytes 6024 (5.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 6024 (5.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

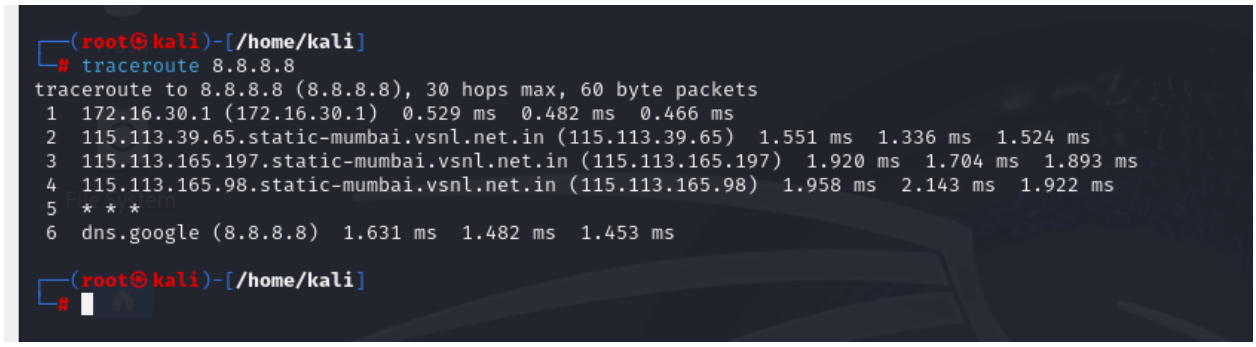
Assign the IP address and netmask

NetStat:



```
┌──(root㉿kali)-[/home/kali]
└─# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 172.16.30.20:bootpc     172.16.30.1:bootps      ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     7070
unix  3      [ ]         STREAM     CONNECTED     8125
unix  3      [ ]         STREAM     CONNECTED     9055
unix  3      [ ]         STREAM     CONNECTED     10435    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     4944     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     12397    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     7011
unix  3      [ ]         STREAM     CONNECTED     12646
unix  3      [ ]         STREAM     CONNECTED     10354    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     12342    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     8551     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     9771
unix  3      [ ]         STREAM     CONNECTED     6611
unix  3      [ ]         STREAM     CONNECTED     11419    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     9463     /run/user/1000/pipewire-0
unix  3      [ ]         STREAM     CONNECTED     9457     /run/user/1000/pulse/native
```

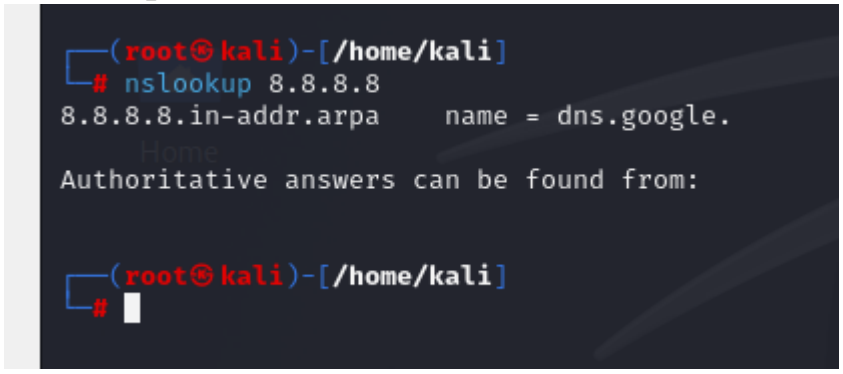
Traceroute:



```
┌──(root㉿kali)-[/home/kali]
└─# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  172.16.30.1 (172.16.30.1)  0.529 ms  0.482 ms  0.466 ms
 2  115.113.39.65.static-mumbai.vsnl.net.in (115.113.39.65)  1.551 ms  1.336 ms  1.524 ms
 3  115.113.165.197.static-mumbai.vsnl.net.in (115.113.165.197)  1.920 ms  1.704 ms  1.893 ms
 4  115.113.165.98.static-mumbai.vsnl.net.in (115.113.165.98)  1.958 ms  2.143 ms  1.922 ms
 5  * * *
 6  dns.google (8.8.8.8)  1.631 ms  1.482 ms  1.453 ms

┌──(root㉿kali)-[/home/kali]
└─#
```

NsLookup:



```
┌──(root㉿kali)-[/home/kali]
└─# nslookup 8.8.8.8
8.8.8.8.in-addr.arpa    name = dns.google.

Authoritative answers can be found from:


┌──(root㉿kali)-[/home/kali]
└─#
```

```
┌──(root㉿kali)-[/home/kali]
└─# nslookup www.google.com
Server:         172.16.100.2
Address:        172.16.100.2#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.183.164
Name:   www.google.com
Address: 2404:6800:4009:825::2004


┌──(root㉿kali)-[/home/kali]
└─#
```

ARP:

```
┌──(root㉿kali)-[/home/kali]
└─# arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
172.16.30.1              ether   d4:76:a0:01:3a:48   C                     eth0
Entries: 1      Skipped: 0      Found: 1

┌──(root㉿kali)-[/home/kali]
└─#
```

Dig:

```
┌──(root㉿kali)-[/home/kali]
└─# dig www.google.com

; <<>> DiG 9.19.21-1+b1-Debian <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27748
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         211     IN      A       142.250.183.164

;; Query time: 4 msec
;; SERVER: 172.16.100.2#53(172.16.100.2) (UDP)
;; WHEN: Fri Jul 26 11:42:37 IST 2024
;; MSG SIZE  rcvd: 59
```

1.8 **Conclusion:**

Hence we learned about Network reconnaissance tools and how to perform them in kali linux

1.9 **Questions:**

1. The **IP** command can show address information, manipulate routing, plus display network various devices, interfaces, and tunnels.

2. The **Tcpdump** command is designed for capturing and displaying packets.

3. The **Netstat** tool is used for printing network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

4. The **nslookup** utility is used to query Internet name servers interactively.

5. **Ping** is a tool that verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages.