user@user-Veriton-M2640G:~/Documents/atr86$ ifconfig

//ethernet

enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.16.5.6 netmask 255.255.252.0 broadcast 172.16.7.255

inet6 fe80::7d80:3410:ea9d:f768 prefixlen 64 scopeid 0x20<link>

ether 94:c6:91:9e:79:63 txqueuelen 1000 (Ethernet)

RX packets 111681 bytes 96178855 (96.1 MB)

RX errors 0 dropped 4 overruns 0 frame 0

TX packets 51935 bytes 9009613 (9.0 MB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

// local network

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 668 bytes 84789 (84.7 KB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 668 bytes 84789 (84.7 KB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

// wireless

wlp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.18.8.217 netmask 255.255.240.0 broadcast 172.18.15.255

inet6 fe80::835f:9710:ecd5:1f7f prefixlen 64 scopeid 0x20<link>

ether d4:6d:6d:7e:78:18 txqueuelen 1000 (Ethernet)

RX packets 29704 bytes 4845172 (4.8 MB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 362 bytes 43481 (43.4 KB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user@user-Veriton-M2640G:~/Documents/atr86$ iwconfig

lo no wireless extensions.

enp1s0 no wireless extensions.

// wireless details

wlp4s0 IEEE 802.11 ESSID:"JUSL\_WLAN1"

Mode:Managed Frequency:5.805 GHz Access Point: 24:F2:7F:09:31:F0

Bit Rate=135 Mb/s Tx-Power=22 dBm

Retry short limit:7 RTS thr:off Fragment thr:off

Power Management:on

Link Quality=37/70 Signal level=-73 dBm

Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0

Tx excessive retries:0 Invalid misc:0 Missed beacon:0

//ip link will list all network interfaces, and you can look for the interface name to identify it.

user@user-Veriton-M2640G:~/Documents/atr86$ ip link

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

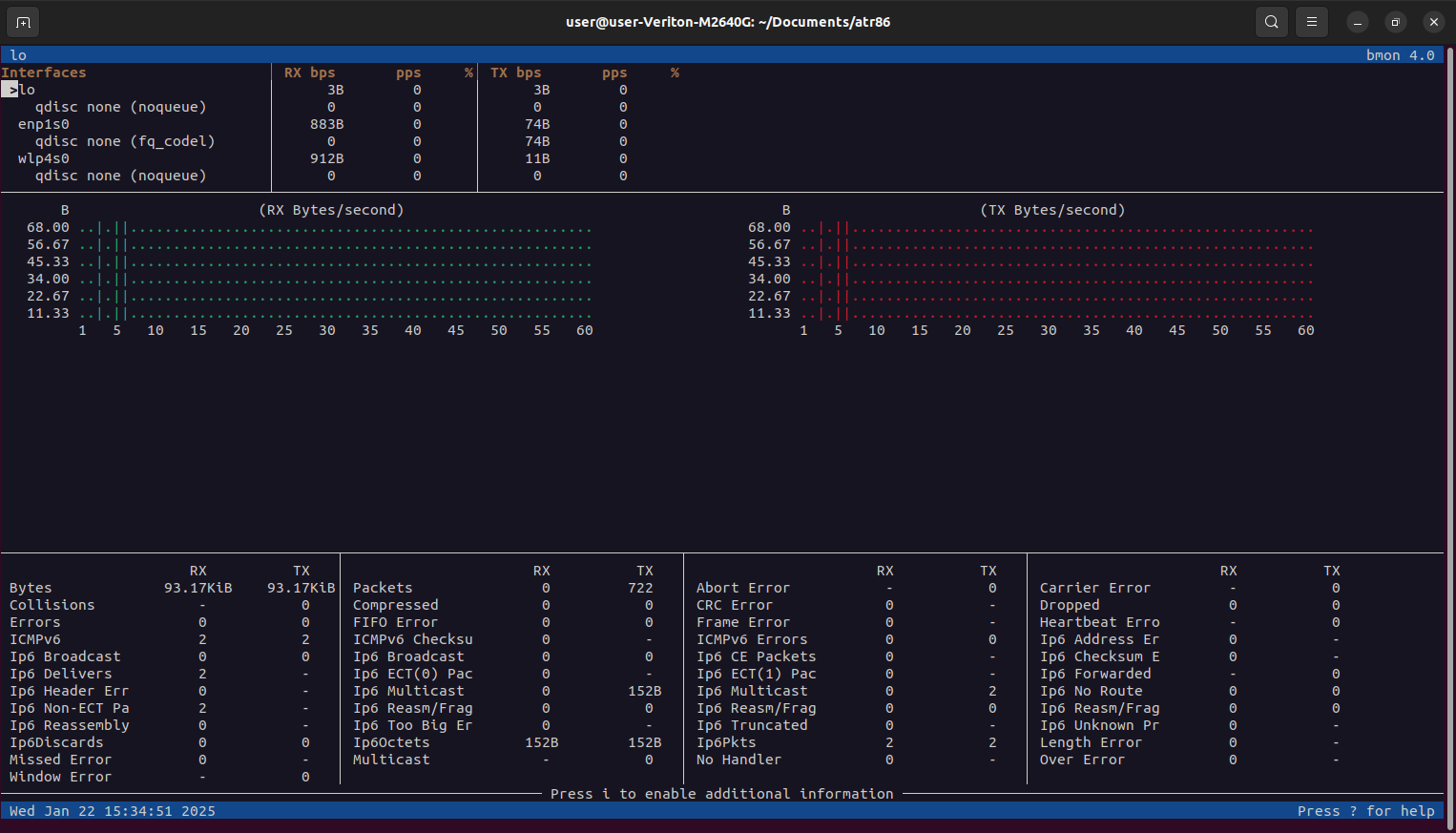
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP mode DEFAULT group default qlen 1000

link/ether 94:c6:91:9e:79:63 brd ff:ff:ff:ff:ff:ff

3: wlp4s0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP mode DORMANT group default qlen 1000

link/ether d4:6d:6d:7e:78:18 brd ff:ff:ff:ff:ff:ff

bmon



### **How Does ping Work?**

When you run the ping command, it sends small packets of data to a specific IP address or hostname (e.g., ping google.com). The target device is expected to respond with an Echo Reply message. The key information you receive from the ping results includes:

* **Round-trip time (RTT)**: The time it takes for the packet to travel to the target and back. This is often displayed in milliseconds (ms).
* **Packet loss**: If some packets don't return, you will see a loss percentage.
* **TTL (Time to Live)**: The number of hops a packet can make before being discarded. Each hop represents a router or device on the network.

user@user-Veriton-M2640G:~/Documents/atr86$ ping google.com

PING google.com (142.250.193.206) 56(84) bytes of data.

64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp\_seq=1 ttl=58 time=30.5 ms

64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp\_seq=2 ttl=58 time=30.9 ms

64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp\_seq=3 ttl=58 time=30.7 ms

64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp\_seq=4 ttl=58 time=30.8 ms

64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp\_seq=5 ttl=58 time=30.6 ms

64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp\_seq=6 ttl=58 time=30.7 ms

^C

--- google.com ping statistics ---

6 packets transmitted, 6 received, 0% packet loss, time 5008ms

rtt min/avg/max/mdev = 30.492/30.684/30.893/0.124 ms

### **What is** traceroute**?**

traceroute is a network diagnostic tool used to trace the path that packets take from your computer to a target host across a network, such as the internet. It shows the series of hops (intermediate routers or devices) that the packets pass through, along with the time it takes for each hop. This can be very useful for identifying where delays or issues occur in the network path.

### **How Does** traceroute **Work?**

When you run traceroute, it sends a series of packets with increasing **Time-to-Live (TTL)** values. TTL specifies how many hops (routers) a packet can make before it is discarded.

1. **First packet**: The TTL is set to 1, so the packet will reach the first router, which will decrement the TTL to 0 and send back an ICMP "Time Exceeded" message.
2. **Second packet**: The TTL is set to 2, so the packet will reach the second router, and the process repeats.
3. **This continues** until the packet reaches the destination, or the TTL exceeds a set maximum value (usually 30 hops).

The result is a list of all routers (hops) the packet encounters on its way to the destination, along with the time it takes for the packet to travel to each hop and back.

### **Key Information in the Output:**

* **Hop Number**: The sequential number of each hop.
* **IP Address/Hostname**: The IP address or domain name of the router or device at that hop.
* **Round-trip Time (RTT)**: The time it took for the packet to travel to that hop and back, usually in milliseconds (ms).
* **Asterisks (**\***)**: This can indicate a timeout or that the router didn't respond to the traceroute query. This can happen if the router is configured to ignore traceroute probes.

user@user-Veriton-M2640G:~/Documents/atr86$ traceroute google.com

traceroute to google.com (142.250.192.238), 64 hops max

1 172.16.4.1 0.209ms 0.219ms 0.215ms

2 136.232.88.1 0.592ms 0.313ms 0.275ms

3 136.232.68.189 1.325ms 1.149ms 1.327ms

4 \* \* \*

5 172.26.14.75 25.788ms 26.438ms 25.701ms

6 172.26.14.75 25.860ms 61.683ms 25.801ms

7 72.14.195.34 29.417ms 29.290ms 29.249ms

8 \* \* \*

9 209.85.252.64 31.877ms 31.924ms 31.768ms

10 142.251.54.63 25.289ms 25.433ms 25.359ms

11 142.251.255.55 28.429ms 28.045ms 28.130ms

12 \* \* \*

13 \* \* \*

14 \* \* \*

^C

user@user-Veriton-M2640G:~/Documents/atr86$ dig google.com

; <<>> DiG 9.18.18-0ubuntu0.22.04.2-Ubuntu <<>> google.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33324

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 65494

;; QUESTION SECTION:

;google.com. IN A

;; ANSWER SECTION:

google.com. 300 IN A 142.250.193.206

;; Query time: 100 msec

;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)

;; WHEN: Wed Jan 22 16:12:53 IST 2025

;; MSG SIZE rcvd: 55

user@user-Veriton-M2640G:~/Documents/atr86$

***telnet***

telnet is a command-line tool that allows you to connect to remote servers and devices over a network using the Telnet protocol. It can be useful for troubleshooting network services, checking if specific ports are open, and accessing remote systems (though SSH is typically preferred due to better security).

While telnet is an older protocol and not secure (because it transmits data, including passwords, in plain text), it can still be useful for certain tasks like testing network connectivity to a service.

### **Basic Syntax:**

telnet

user@user-Veriton-M2640G:~/Documents/atr86$ telnet 172.16.4.112

Trying 172.16.4.112...

Connected to 172.16.4.112.

Escape character is '^]'.

Kernel 3.10.0-514.21.1.el7.x86\_64 on an x86\_64

localhost login: be2386

Password:

Last login: Wed Jan 22 17:55:54 from ::ffff:172.16.5.6

[be2386@localhost ~]$ ls

ccfile.cpp cpnm dsa dsa2 oops oos q5.c skel

[be2386@localhost ~]$ cd oos

[be2386@localhost oos]$ ls

a1 Main.class q1.java q2.java q3.java q4.java Room.class RoomDemo.class Stack.class student.class

[be2386@localhost oos]$ cd ..

[be2386@localhost ~]$ ls

ccfile.cpp cpnm dsa dsa2 oops oos q5.c skel

[be2386@localhost ~]$ cd ..

[be2386@localhost yr23]$ ls

be2301 be2307 be2313 be2319 be2325 be2331 be2337 be2343 be2349 be2355 be2361 be2367 be2373 be2379 be2385 be2391 be2397 be23L04

be2302 be2308 be2314 be2320 be2326 be2332 be2338 be2344 be2350 be2356 be2362 be2368 be2374 be2380 be2386 be2392 be2398 be23L05

be2303 be2309 be2315 be2321 be2327 be2333 be2339 be2345 be2351 be2357 be2363 be2369 be2375 be2381 be2387 be2393 be2399 be23L06

be2304 be2310 be2316 be2322 be2328 be2334 be2340 be2346 be2352 be2358 be2364 be2370 be2376 be2382 be2388 be2394 be23L01 be23L07

be2305 be2311 be2317 be2323 be2329 be2335 be2341 be2347 be2353 be2359 be2365 be2371 be2377 be2383 be2389 be2395 be23L02 be23L08

be2306 be2312 be2318 be2324 be2330 be2336 be2342 be2348 be2354 be2360 be2366 be2372 be2378 be2384 be2390 be2396 be23L03 be23L09

[be2386@localhost yr23]$ cd ..

[be2386@localhost ug]$ ls

guest pr21 pr22 pr23 yr14 yr15 yr16 yr17 yr18 yr19 yr20 yr21 yr22 yr23

[be2386@localhost ug]$ cd ..

[be2386@localhost student]$ ls

pg ug

[be2386@localhost student]$ cd ..

[be2386@localhost usr]$ ls

faculty student

[be2386@localhost usr]$ exit

logout

Connection closed by foreign host.

user@user-Veriton-M2640G:~/Documents/atr86$ telnet 172.16.4.112

Trying 172.16.4.112...

Connected to 172.16.4.112.

Escape character is '^]'.

Kernel 3.10.0-514.21.1.el7.x86\_64 on an x86\_64

localhost login: be2386

Password:

Login incorrect

localhost login: be2386

Password:

Last failed login: Wed Jan 22 18:00:54 IST 2025 from ::ffff:172.16.5.6 on pts/7

There was 1 failed login attempt since the last successful login.

Last login: Wed Jan 22 17:58:13 from ::ffff:172.16.5.6

[be2386@localhost ~]$ ls

ccfile.cpp cpnm dsa dsa2 oops oos q5.c skel

[be2386@localhost ~]$ cd oos

[be2386@localhost oos]$ ls

a1 Main.class q1.java q2.java q3.java q4.java Room.class RoomDemo.class Stack.class student.class

[be2386@localhost oos]$ cd ..

[be2386@localhost ~]$ ls

ccfile.cpp cpnm dsa dsa2 oops oos q5.c skel

[be2386@localhost ~]$ cd ..

[be2386@localhost yr23]$ exit

logout

Connection closed by foreign host.

user@user-Veriton-M2640G:~/Documents/atr86$

**Route**

user@user-Veriton-M2640G:~/Documents/atr86$ route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

default \_gateway 0.0.0.0 UG 100 0 0 enp1s0

default \_gateway 0.0.0.0 UG 20600 0 0 wlp4s0

link-local 0.0.0.0 255.255.0.0 U 1000 0 0 enp1s0

172.16.4.0 0.0.0.0 255.255.252.0 U 100 0 0 enp1s0

172.18.0.0 0.0.0.0 255.255.240.0 U 600 0 0 wlp4s0

user@user-Veriton-M2640G:~/Documents/atr86$

[user@localhost ~]$ route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

default gateway 0.0.0.0 UG 100 0 0 enp4s0

default gateway 0.0.0.0 UG 600 0 0 wlp3s0

172.16.4.0 0.0.0.0 255.255.252.0 U 100 0 0 enp4s0

172.18.0.0 0.0.0.0 255.255.240.0 U 600 0 0 wlp3s0

192.168.122.0 0.0.0.0 255.255.255.0 U 0 0 0 virbr0

[user@localhost ~]$

ROUTE(8) Linux System Administrator's Manual ROUTE(8)

NAME

route - show / manipulate the IP routing table

SYNOPSIS

route [-CFvnNee] [-A family |-4|-6]

route [-v] [-A family |-4|-6] add [-net|-host] target [netmask Nm] [gw Gw] [metric N] [mss M] [window W] [irtt I] [reject] [mod] [dyn]

[reinstate] [[dev] If]

route [-v] [-A family |-4|-6] del [-net|-host] target [gw Gw] [netmask Nm] [metric M] [[dev] If]

route [-V] [--version] [-h] [--help]

NOTE

This program is obsolete. For replacement check ip route.

DESCRIPTION

Route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an inter‐

face after it has been configured with the ifconfig(8) program.

When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the

routing tables.

**host**

HOST(1) BIND 9 HOST(1)

NAME

host - DNS lookup utility

SYNOPSIS

host [-aACdlnrsTUwv] [-c class] [-N ndots] [-p port] [-R number] [-t type] [-W wait] [-m flag] [ [-4] | [-6] ] [-v] [-V] {name} [server]

DESCRIPTION

host is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa. When no arguments or options are

given, host prints a short summary of its command-line arguments and options.

name is the domain name that is to be looked up. It can also be a dotted-decimal IPv4 address or a colon-delimited IPv6 address, in which case host by de‐

fault performs a reverse lookup for that address. server is an optional argument which is either the name or IP address of the name server that host should

query instead of the server or servers listed in /etc/resolv.conf.

OPTIONS

-4 This option specifies that only IPv4 should be used for query transport. See also the -6 option.

-6 This option specifies that only IPv6 should be used for query transport. See also the -4 option.

-a The -a ("all") option is normally equivalent to -v -t ANY. It also affects the behavior of the -l list zone option.

-A The -A ("almost all") option is equivalent to -a, except that RRSIG, NSEC, and NSEC3 records are omitted from the output.

-c class

This option specifies the query class, which can be used to lookup HS (Hesiod) or CH (Chaosnet) class resource records. The default class is IN (In‐

ternet).

-C This option indicates that named should check consistency, meaning that host queries the SOA records for zone name from all the listed authoritative

name servers for that zone. The list of name servers is defined by the NS records that are found for the zone.

-d This option prints debugging traces, and is equivalent to the -v verbose option.

-l This option tells named to list the zone, meaning the host command performs a zone transfer of zone name and prints out the NS, PTR, and address

records (A/AAAA).

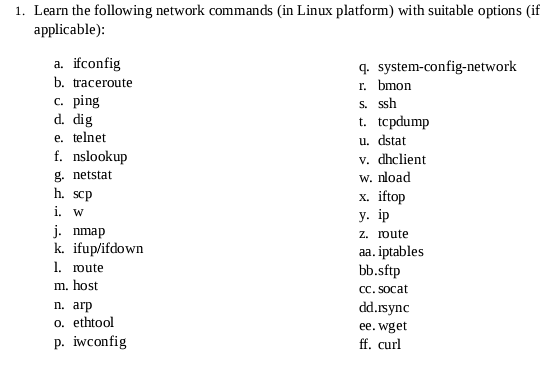
Together, the -l -a options print all records in the zone.

-N ndots

This option specifies the number of dots (ndots) that have to be in name for it to be considered absolute. The default value is that defined using

the ndots statement in /etc/resolv.conf, or 1 if no ndots statement is present. Names with fewer dots are interpreted as relative names, and are

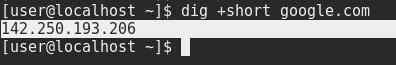
searched for in the domains listed in the search or domain directive in /etc/resolv.conf.



DAY 2

dig:-

dig +short google.com



SCP(1) BSD General Commands Manual SCP(1)

NAME

scp — secure copy (remote file copy program)

SYNOPSIS

scp [-12346BCpqrv] [-c cipher] [-F ssh\_config] [-i identity\_file] [-l limit] [-o ssh\_option] [-P port] [-S program] [[user@]host1:]file1 ...

[[user@]host2:]file2

DESCRIPTION

scp copies files between hosts on a network. It uses ssh(1) for data transfer, and uses the same authentication and provides the same secu‐

rity as ssh(1). scp will ask for passwords or passphrases if they are needed for authentication.

File names may contain a user and host specification to indicate that the file is to be copied to/from that host. Local file names can be

made explicit using absolute or relative pathnames to avoid scp treating file names containing ‘:’ as host specifiers. Copies between two

remote hosts are also permitted.

**NETSTAT**

# netstat

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

[user@localhost ~]$ netstat -s

Ip:

Forwarding: 1

149439 total packets received

11 with invalid addresses

0 forwarded

0 incoming packets discarded

122958 incoming packets delivered

63115 requests sent out

4 outgoing packets dropped

2 dropped because of missing route

103 fragments dropped after timeout

112 reassemblies required

103 packet reassemblies failed

Icmp:

46 ICMP messages received

0 input ICMP message failed

ICMP input histogram:

destination unreachable: 20

timeout in transit: 26

20 ICMP messages sent

0 ICMP messages failed

ICMP output histogram:

destination unreachable: 20

IcmpMsg:

InType3: 20

InType11: 26

OutType3: 20

Tcp:

752 active connection openings

0 passive connection openings

1 failed connection attempts

135 connection resets received

3 connections established

47534 segments received

61841 segments sent out

585 segments retransmitted

1 bad segments received

163 resets sent

Udp:

47271 packets received

20 packets to unknown port received

0 packet receive errors

657 packets sent

0 receive buffer errors

0 send buffer errors

IgnoredMulti: 28087netstat

UdpLite:

TcpExt:

530 TCP sockets finished time wait in fast timer

1416 delayed acks sent

1 delayed acks further delayed because of locked socket

Quick ack mode was activated 44 times

1 packets directly queued to recvmsg prequeue

24225 packet headers predicted

9430 acknowledgments not containing data payload received

8900 predicted acknowledgments

TCPSackRecovery: 235

Detected reordering 1 times using SACK

TCPDSACKUndo: 2

4 congestion windows recovered without slow start after partial ack

TCPLostRetransmit: 6

518 fast retransmits

2 retransmits in slow start

TCPTimeouts: 6

TCPLossProbes: 328

TCPLossProbeRecovery: 35

TCPSackRecoveryFail: 1

TCPDSACKOldSent: 44

TCPDSACKRecv: 10

11 connections reset due to unexpected data

130 connections reset due to early user close

TCPDSACKIgnoredNoUndo: 1

TCPSackShiftFallback: 1578

IPReversePathFilter: 8

TCPRcvCoalesce: 5204

TCPOFOQueue: 510

TCPChallengeACK: 1

TCPSYNChallenge: 1

TCPAutoCorking: 2948

TCPSynRetrans: 4

TCPOrigDataSent: 38485

TCPHystartTrainDetect: 1

TCPHystartTrainCwnd: 20

TCPKeepAlive: 1619

IpExt:

InNoRoutes: 1

InMcastPkts: 46965

OutMcastPkts: 211

InBcastPkts: 32390

OutBcastPkts: 18

InOctets: 78323342

OutOctets: 44705190

InMcastOctets: 8685013

OutMcastOctets: 22116

InBcastOctets: 4177304

OutBcastOctets: 1404

InNoECTPkts: 171544

[user@localhost ~]$

[user@localhost ~]$ netstat --listening

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 0.0.0.0:sunrpc 0.0.0.0:\* LISTEN

tcp 0 0 localhost.locald:domain 0.0.0.0:\* LISTEN

tcp 0 0 localhost:ipp 0.0.0.0:\* LISTEN

tcp6 0 0 [::]:sunrpc [::]:\* LISTEN

tcp6 0 0 localhost:ipp [::]:\* LISTEN

tcp6 0 0 localhost:smc-https [::]:\* LISTEN

udp 0 0 0.0.0.0:mdns 0.0.0.0:\*

udp 0 0 localhost.locald:domain 0.0.0.0:\*

udp 0 0 0.0.0.0:bootps 0.0.0.0:\*

udp 0 0 0.0.0.0:bootpc 0.0.0.0:\*

udp 0 0 0.0.0.0:47173 0.0.0.0:\*

udp 0 0 0.0.0.0:sunrpc 0.0.0.0:\*

udp 0 0 0.0.0.0:6326 0.0.0.0:\*

udp 0 0 localhost:323 0.0.0.0:\*

udp6 0 0 [::]:41625 [::]:\*

udp6 0 0 [::]:33900 [::]:\*

udp6 0 0 [::]:mdns [::]:\*

udp6 0 0 [::]:sunrpc [::]:\*

udp6 0 0 localhost:323 [::]:\*

raw6 0 0 [::]:ipv6-icmp [::]:\* 7

Active UNIX domain sockets (only servers)

Proto RefCnt Flags Type State I-Node Path

unix 2 [ ACC ] STREAM LISTENING 21504 /var/run/mcelog-client

unix 2 [ ACC ] STREAM LISTENING 30653 @/tmp/.ICE-unix/1744

unix 2 [ ACC ] STREAM LISTENING 25066 @/tmp/.ICE-unix/1069

unix 2 [ ACC ] STREAM LISTENING 30488 /run/user/1000/systemd/private

unix 2 [ ACC ] STREAM LISTENING 30494 /run/user/1000/bus

unix 2 [ ACC ] STREAM LISTENING 30525 @/tmp/.X11-unix/X0

unix 2 [ ACC ] STREAM LISTENING 30504 /run/user/1000/keyring/control

unix 2 [ ACC ] STREAM LISTENING 26666 /var/run/NetworkManager/private-dhcp

unix 2 [ ACC ] STREAM LISTENING 34862 /tmp/.esd-1000/socket

unix 2 [ ACC ] STREAM LISTENING 34865 /run/user/1000/pulse/native

unix 2 [ ACC ] STREAM LISTENING 15667 /run/systemd/private

unix 2 [ ACC ] STREAM LISTENING 22248 /var/lib/gssproxy/default.sock

unix 2 [ ACC ] STREAM LISTENING 30526 /tmp/.X11-unix/X0

unix 2 [ ACC ] STREAM LISTENING 27899 @/tmp/dbus-5wY5W7V7ko

unix 2 [ ACC ] STREAM LISTENING 15681 /run/rpcbind.sock

unix 2 [ ACC ] SEQPACKET LISTENING 15687 /run/udev/control

unix 2 [ ACC ] STREAM LISTENING 24819 @/tmp/dbus-RX9yuIsT

unix 2 [ ACC ] STREAM LISTENING 21462 @ISCSID\_UIP\_ABSTRACT\_NAMESPACE

unix 2 [ ACC ] STREAM LISTENING 43348 /tmp/.org.chromium.Chromium.woFBwz/SingletonSocket

unix 2 [ ACC ] STREAM LISTENING 23391 /run/user/42/systemd/private

unix 2 [ ACC ] STREAM LISTENING 23396 /var/run/libvirt/libvirt-sock

unix 2 [ ACC ] STREAM LISTENING 23398 /var/run/libvirt/libvirt-sock-ro

unix 2 [ ACC ] STREAM LISTENING 32875 /run/user/1000/keyring/pkcs11

unix 2 [ ACC ] STREAM LISTENING 23403 /run/user/42/bus

unix 2 [ ACC ] STREAM LISTENING 24816 @/tmp/dbus-UNVomodY

unix 2 [ ACC ] STREAM LISTENING 32877 /run/user/1000/keyring/ssh

unix 2 [ ACC ] STREAM LISTENING 26613 @/tmp/dbus-pEYrqy0a

unix 2 [ ACC ] STREAM LISTENING 24815 @/tmp/dbus-tOPyghLP

unix 2 [ ACC ] STREAM LISTENING 75383 /tmp/OSL\_PIPE\_1000\_SingleOfficeIPC\_ebf4c14b10ff5047a7c52cb16b89cb45

unix 2 [ ACC ] STREAM LISTENING 28027 /run/user/42/pulse/native

unix 2 [ ACC ] STREAM LISTENING 32924 @/tmp/dbus-F6guJvkj

unix 2 [ ACC ] STREAM LISTENING 29403 @/tmp/dbus-1g8ii7H9

unix 2 [ ACC ] STREAM LISTENING 26512 /run/user/42/wayland-0

unix 2 [ ACC ] STREAM LISTENING 26520 /var/run/ceph/ceph-mon.localhost.asok

unix 2 [ ACC ] STREAM LISTENING 32192 @/tmp/dbus-IPmG5kBDnm

unix 2 [ ACC ] STREAM LISTENING 29402 @/tmp/dbus-Kt6GBcpj

unix 2 [ ACC ] STREAM LISTENING 1468 /run/systemd/journal/stdout

unix 2 [ ACC ] STREAM LISTENING 30654 /tmp/.ICE-unix/1744

unix 2 [ ACC ] STREAM LISTENING 21459 @ISCSIADM\_ABSTRACT\_NAMESPACE

unix 2 [ ACC ] STREAM LISTENING 21450 /var/run/libvirt/virtlockd-sock

unix 2 [ ACC ] STREAM LISTENING 21453 /run/dbus/system\_bus\_socket

unix 2 [ ACC ] STREAM LISTENING 21456 /var/run/avahi-daemon/socket

unix 2 [ ACC ] STREAM LISTENING 16592 /run/lvm/lvmpolld.socket

unix 2 [ ACC ] STREAM LISTENING 26871 @/tmp/.X11-unix/X1024

unix 2 [ ACC ] STREAM LISTENING 21460 /var/run/cups/cups.sock

unix 2 [ ACC ] STREAM LISTENING 21463 /var/run/libvirt/virtlogd-sock

unix 2 [ ACC ] STREAM LISTENING 16600 /run/lvm/lvmetad.socket

unix 2 [ ACC ] SEQPACKET LISTENING 16604 /run/systemd/coredump

unix 2 [ ACC ] STREAM LISTENING 24818 @/tmp/dbus-SRtl1O8A

unix 2 [ ACC ] STREAM LISTENING 22249 /run/gssproxy.sock

unix 2 [ ACC ] STREAM LISTENING 25067 /tmp/.ICE-unix/1069

unix 2 [ ACC ] STREAM LISTENING 22262 /var/run/abrt/abrt.socket

unix 2 [ ACC ] STREAM LISTENING 26872 /tmp/.X11-unix/X1024

Active Bluetooth connections (only servers)

Proto Destination Source State PSM DCID SCID IMTU OMTU Security

Proto Destination Source State Channel

[user@localhost ~]$

Netstat:- displays all statistics about packets and the network

ss:- same like netstat, but shows more info in a faster way

**SS(8)**  System Manager's Manual SS(8)

NAME

ss - another utility to investigate sockets

SYNOPSIS

ss [options] [ FILTER ]

DESCRIPTION

ss is used to dump socket statistics. It allows showing information similar to netstat. It can display more TCP and

state information than other tools.

aroy@aroy-VirtualBox:~/Desktop$ ss --tcp // display only tcp packets also (ss -t)

State Recv-Q Send-Q Local Address:Port Peer Address:Port Process

ESTAB 0 0 10.0.2.15:48396 142.250.192.46:https

ESTAB 0 0 10.0.2.15:57044 142.250.195.234:https

ESTAB 0 0 10.0.2.15:49118 142.250.77.110:https

ESTAB 0 0 10.0.2.15:57186 142.250.183.67:

**NSLOOKUP(1)**  BIND9 NSLOOKUP(1)

NAME

nslookup - query Internet name servers interactively

SYNOPSIS

nslookup [-option] [name | -] [server]

DESCRIPTION

Nslookup is a program to query Internet domain name servers. Nslookup has two modes: interactive and non-interactive. Interactive mode

allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain.

Non-interactive mode is used to print just the name and requested information for a host or domain.

[user@localhost ~]$ nslookup google.com

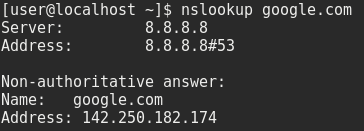
Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: google.com

Address: 142.250.182.174



[**user@localhost ~]$ iwconfig**

virbr0 no wireless extensions.

virbr0-nic no wireless extensions.

lo no wireless extensions.

enp4s0 no wireless extensions.

wlp3s0 IEEE 802.11 ESSID:"JUSL\_WLAN1"

Mode:Managed Frequency:2.462 GHz Access Point: 24:F2:7F:09:31:E0

Bit Rate=12 Mb/s Tx-Power=15 dBm

Retry short limit:7 RTS thr:off Fragment thr:off

Power Management:off

Link Quality=37/70 Signal level=-73 dBm

Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0

Tx excessive retries:0 Invalid misc:1 Missed beacon:0

[user@localhost ~]$

**System-config-network**

requires password of admin

iptables

Iptables and ip6tables are used to set up, maintain, and inspect the tables of IPv4 and IPv6 packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.

Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called

a `target', which may be a jump to a user-defined chain in the same table.

**tcpdump**

Requires sudo access

Captures 10 packets that has been communicated through tcp

oem@oem-HP-ProDesk-600-G2-MT:~$ **sudo tcpdump -c 10**

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode

listening on eno1, link-type EN10MB (Ethernet), snapshot length 262144 bytes

15:42:03.315232 ARP, Request who-has 172.16.5.13 tell \_gateway, length 46

15:42:03.321172 ARP, Request who-has 172.16.5.13 tell 172.16.6.24, length 46

15:42:03.384941 IP 172.16.4.42.netbios-ns > 172.16.7.255.netbios-ns: UDP, length 50

15:42:03.384942 IP6 fe80::b5e8:9a04:b44a:d8da.51825 > ff02::1:3.5355: UDP, length 22

15:42:03.384948 IP 172.16.4.42.51825 > 224.0.0.252.5355: UDP, length 22

15:42:03.390624 IP oem-HP-ProDesk-600-G2-MT.50065 > one.one.one.one.domain: 1386+ PTR? 13.5.16.172.in-addr.arpa. (42)

15:42:03.401713 IP 172.16.15.8.http-alt > oem-HP-ProDesk-600-G2-MT.33014: Flags [P.], seq 2367159415:2367159531, ack 2201209625, win 1432, options [nop,nop,TS val 193384638 ecr 913974201], length 116: HTTP

15:42:03.426952 IP one.one.one.one.domain > oem-HP-ProDesk-600-G2-MT.50065: 1386 NXDomain 0/0/0 (42)

15:42:03.427484 IP oem-HP-ProDesk-600-G2-MT.36252 > one.one.one.one.domain: 670+ PTR? 1.4.16.172.in-addr.arpa. (41)

15:42:03.442867 IP oem-HP-ProDesk-600-G2-MT.33014 > 172.16.15.8.http-alt: Flags [.], ack 116, win 501, options [nop,nop,TS val 913974492 ecr 193384638], length 0

**10 packets captured**

32 packets received by filter

0 packets dropped by kernel

oem@oem-HP-ProDesk-600-G2-MT:~$

**TO INTERFACE enp0s3**

sudo tcpdump -c 5 -i enp0s3

[sudo] password for aroy:

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode

listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes

00:23:47.224369 IP aroy-VirtualBox.44446 > bom12s20-in-f14.1e100.net.https: Flags [P.], seq 2974257462:2974258794, ack 14636398, win 65535, length 1332

00:23:47.224983 IP aroy-VirtualBox.44446 > bom12s20-in-f14.1e100.net.https: Flags [P.], seq 1332:1827, ack 1, win 65535, length 495

00:23:47.228651 IP bom12s20-in-f14.1e100.net.https > aroy-VirtualBox.44446: Flags [.], ack 1332, win 65535, length 0

00:23:47.228659 IP bom12s20-in-f14.1e100.net.https > aroy-VirtualBox.44446: Flags [.], ack 1827, win 65535, length 0

00:23:47.282487 IP aroy-VirtualBox.47374 > dlinkrouter.domain: 63432+ [1au] PTR? 46.42.251.142.in-addr.arpa. (55)

5 packets captured

18 packets received by filter

0 packets dropped by kernel

tcpdump -D

1.enp0s3 [Up, Running, Connected]

2.any (Pseudo-device that captures on all interfaces) [Up, Running]

3.lo [Up, Running, Loopback]

4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]

5.nflog (Linux netfilter log (NFLOG) interface) [none]

6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]

7.dbus-system (D-Bus system bus) [none]

8.dbus-session (D-Bus session bus) [none]

aroy@aroy-VirtualBox:~/Desktop$

**W(1) User Commands W(1)**

NAME

w - Show who is logged on and what they are doing.

SYNOPSIS

w [options] user [...]

DESCRIPTION

w displays information about the users currently on the machine, and their processes. The header shows, in this order,

the current time, how long the system has been running, how many users are currently logged on, and the system load aver‐

ages for the past 1, 5, and 15 minutes.

The following entries are displayed for each user: login name, the tty name, the remote host, login time, idle time,

JCPU, PCPU, and the command line of their current process.

The JCPU time is the time used by all processes attached to the tty. It does not include past background jobs, but does

include currently running background jobs.

The PCPU time is the time used by the current process, named in the "what" field.

aroy@aroy-VirtualBox:~/Desktop$ w

19:37:12 up 1:47, 1 user, load average: 1.26, 1.19, 0.93

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

aroy tty2 tty2 17:29 2:08m 0.05s 0.04s /usr/libexec/gnome-session-binary --session=ubuntu

**NMAP**

Nmap (“Network Mapper”) is an open source tool for network exploration and security auditing. It was designed to rapidly

scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine

what hosts are available on the network, what services (application name and version) those hosts are offering, what

operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of

other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it

useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service

uptime.

OBTAINED IP FROM INET FIELD OF INTERFACES SHOWN IN IFCONFIG

aroy@aroy-VirtualBox:~$ nmap 127.0.0.1

Starting Nmap 7.80 ( https://nmap.org ) at 2025-02-21 00:31 IST

Nmap scan report for localhost (127.0.0.1)

Host is up (0.000060s latency).

Not shown: 999 closed ports

PORT STATE SERVICE

631/tcp open ipp

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds

aroy@aroy-VirtualBox:~$ nmap 10.0.2.15

Starting Nmap 7.80 ( https://nmap.org ) at 2025-02-21 00:31 IST

Nmap scan report for aroy-VirtualBox (10.0.2.15)

Host is up (0.000062s latency).

All 1000 scanned ports on aroy-VirtualBox (10.0.2.15) are closed

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds

aroy@aroy-VirtualBox:~$

SCANNING ALL IPs on the network —- why -Pn?

nmap 192.168.0.161

Starting Nmap 7.80 ( https://nmap.org ) at 2025-02-21 01:25 IST

**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 3.03 seconds

roy@aroy-VirtualBox:~$ nmap -Pn 192.168.0.1

Starting Nmap 7.80 ( https://nmap.org ) at 2025-02-21 01:28 IST

Nmap scan report for dlinkrouter (192.168.0.1)

Host is up (0.024s latency).

Not shown: 995 filtered ports

PORT STATE SERVICE

53/tcp open domain

80/tcp open http

443/tcp open https

8888/tcp open sun-answerbook

9999/tcp open abyss

Nmap done: 1 IP address (1 host up) scanned in 8.86 seconds

aroy@aroy-VirtualBox:~$ nmap -Pn 192.168.0.133

Starting Nmap 7.80 ( https://nmap.org ) at 2025-02-21 01:29 IST

Nmap scan report for Galaxy-M34-5G (192.168.0.133)

Host is up (0.019s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

110/tcp open pop3

Nmap done: 1 IP address (1 host up) scanned in 6.92 seconds

aroy@aroy-VirtualBox:~$ nmap -Pn 192.168.0.161

Starting Nmap 7.80 ( https://nmap.org ) at 2025-02-21 01:30 IST

Nmap scan report for DESKTOP-QAU5LNL (192.168.0.161)

Host is up (0.0071s latency).

Not shown: 996 filtered ports

PORT STATE SERVICE

110/tcp open pop3

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 4.74 seconds

aroy@aroy-VirtualBox:~$

Traceroute MORE:- ON Virtual Machine

1 \_gateway (10.0.2.2)

: This shows that the first hop is your local gateway (likely your router or modem). The IP address 10.0.2.2 is a private IP address, indicating it's within your local network. The times (1.218 ms, 1.130 ms, 1.067 ms) are the round-trip times (RTT) for the packets to reach the gateway and return.

2 \_gateway (10.0.2.2): This is interesting. You're seeing your own gateway again as the second hop. This often indicates some kind of network address translation (NAT) or internal routing happening within your local network. It might be a virtualized network environment or a more complex home network setup. It's not necessarily an error, but it's worth understanding your network configuration if you see this.

**Ethtool**

**https://www.baeldung.com/linux/using-ethtool**

**The** [***ethtool***](https://man7.org/linux/man-pages/man8/ethtool.8.html)**is a command-line tool in Linux for managing** [**network interface devices**](https://www.baeldung.com/linux/network-interface-configure)**. It allows us to modify the parameters of the devices and query the information of those devices.**

**To get the general properties of a network interface device, we simply run *ethtool* followed by its name**:

aroy@aroy-VirtualBox:~/Desktop$ ethtool enp0s3

Settings for enp0s3:

Supported ports: [ TP ]

Supported link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100baseT/Full

1000baseT/Full

Supported pause frame use: No

Supports auto-negotiation: Yes

Supported FEC modes: Not reported

Advertised link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100baseT/Full

1000baseT/Full

Advertised pause frame use: No

Advertised auto-negotiation: Yes

Advertised FEC modes: Not reported

Speed: 1000Mb/s

Duplex: Full

Auto-negotiation: on

Port: Twisted Pair

PHYAD: 0

Transceiver: internal

MDI-X: off (auto)

netlink error: Operation not permitted

Current message level: 0x00000007 (7)

drv probe link

Link detected: yes

**roy@aroy-VirtualBox:~/Desktop$ sudo ethtool --driver enp0s3**

driver: e1000

version: 6.8.0-52-generic

firmware-version:

expansion-rom-version:

bus-info: 0000:00:03.0

supports-statistics: yes

supports-test: yes

supports-eeprom-access: yes

supports-register-dump: yes

supports-priv-flags: no

### Obtaining Device Statistics

sudo ethtool --statistics enp0s3

aroy@aroy-VirtualBox:~/Desktop$ sudo ethtool -S enp0s3

NIC statistics:

rx\_packets: 195588

tx\_packets: 110352

rx\_bytes: 162309924

tx\_bytes: 38385195

rx\_broadcast: 0

tx\_broadcast: 5

rx\_multicast: 0

tx\_multicast: 74

rx\_errors: 0

tx\_errors: 0

tx\_dropped: 0

multicast: 0

collisions: 0

rx\_length\_errors: 0

rx\_over\_errors: 0

rx\_crc\_errors: 0

rx\_frame\_errors: 0

rx\_no\_buffer\_count: 0

rx\_missed\_errors: 0

tx\_aborted\_errors: 0

tx\_carrier\_errors: 0

tx\_fifo\_errors: 0

tx\_heartbeat\_errors: 0

tx\_window\_errors: 0

tx\_abort\_late\_coll: 0

tx\_deferred\_ok: 0

tx\_single\_coll\_ok: 0

tx\_multi\_coll\_ok: 0

tx\_timeout\_count: 0

tx\_restart\_queue: 0

rx\_long\_length\_errors: 0

rx\_short\_length\_errors: 0

rx\_align\_errors: 0

tx\_tcp\_seg\_good: 5198

tx\_tcp\_seg\_failed: 0

rx\_flow\_control\_xon: 0

rx\_flow\_control\_xoff: 0

tx\_flow\_control\_xon: 0

tx\_flow\_control\_xoff: 0

rx\_long\_byte\_count: 162309924

rx\_csum\_offload\_good: 0

rx\_csum\_offload\_errors: 0

alloc\_rx\_buff\_failed: 0

tx\_smbus: 0

rx\_smbus: 0

dropped\_smbus: 0

aroy@aroy-VirtualBox:~/Desktop$

In Ethernet, the [PAUSE frame mechanism](https://www.juniper.net/documentation/us/en/software/junos/traffic-mgmt-qfx/topics/concept/cos-qfx-series-congestion-notification-understanding.html) is a way to relieve traffic congestion during transfer. When one end of the data link cannot catch up, it can send a pause frame to the other end to slow down the transmission rate. **The *ethtool* command offers multiple options that we can use to query and configure the parameters associated with the PAUSE frame mechanism**.

/Desktop$ sudo ethtool --show-pause enp0s3

Pause parameters for enp0s3:

Autonegotiate: on

RX: on

TX: off

IPTABLES:-

n the modern world, a large amount of data is exchanged between machines. In most cases, the exchange happens between two untrusted machines. For example, any data that flows over HTTP is agnostic of the machine on which the application runs on.

With a specific focus on privacy and data protection, a machine must limit its network to a trusted list of clients. So with this in mind, we usually protect a network behind a firewall.

**In this tutorial, we’ll discuss iptables, which is a user-space firewall for Linux machines. It filters connections based on user-defined rules**. In the following sections, we’ll understand these rules and their behaviors in detail.

**Socat:-**

In this tutorial, we’ll take a look at the [*socat*](https://linux.die.net/man/1/socat) command in Linux. *Socat* is a flexible, multi-purpose relay tool. **Its purpose is to establish a relationship between two data sources**, where each data source can be a file, a Unix socket, UDP, TCP, or standard input.

*Socat* is useful for connecting applications inside separate boxes. Imagine we have Box A and Box B, and inside Box A, there’s a database server application running. Furthermore, Box A is closed to the public, but Box B is open. Our network will allow a connection from Box B to Box A.

**sudo apt-get install -y socat**

Using nc:- Install from user packages if not present

Let’s connect *nc* with the Transmission Control Protocol (TCP) and stream data from both directions. We’ll need two console terminals to conduct this experiment.

On the first terminal, let’s run *nc* in listening mode:

Terminal 1**:-nc -l localhost 1234**

The *-l* flag indicates *nc* is in the listening mode. It listens on *localhost* port *1234*.

On the second terminal, let’s run *socat* to connect [*STDIO*](https://www.baeldung.com/linux/pipes-redirection) to *localhost* with port *1234* using the TCP protocol:

Terminal 2:-**socat STDIO TCP4:localhost:1234**

n the command above, the first argument is the standard input, represented with the keyword *STDIO*. The second argument is a string with a special syntax. As we can see, the string is divided into three parts with a colon delimiter. The first part is the address format, *TCP4*. The second part is the server or the IP address, *localhost*. The last part is the port, *1234*.

The *socat* application connects the stream from the first argument *(STDIO)* to the one mentioned in the second argument *(TCP4:localhost:1234).* In this case, we can switch the order of the arguments and it doesn’t matter because it’s bidirectional.

TYPE IN 1 OR 2:- APPEARS IN ANOTHER!

ip

p - show / manipulate routing, network devices, interfaces and tunnels

ip address

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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000

link/ether 08:00:27:e8:29:5a brd ff:ff:ff:ff:ff:ff

inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3

valid\_lft 80422sec preferred\_lft 80422sec

inet6 fe80::cff9:6f1:7bbc:c016/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

aroy@aroy-VirtualBox:~/Desktop$

SSH and related commands:-

sudo ssh aroy@aroy-VirtualBox@172.0.0.1

[sudo] password for aroy:

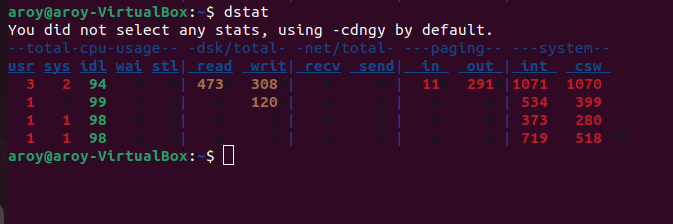
CONNECTION REFUSED

WSL/VM or VM/VM not allowed

**DSTAT**

**Use dstat if you need a versatile, real-time snapshot of multiple system metrics, including network, CPU, disk, and memory.**

**Compared to other stat options like top, vmstat, netstat, iostat, and sar.**



**Combining Network and Other System Stats** You can combine various stats to get a more comprehensive view of your system, including both CPU and network activity. For example:

**$ dstat -c -n**

**--total-cpu-usage-- -net/total-**

**usr sys idl wai stl| recv send**

**Monitor Network Traffic by Protocol You can even monitor traffic by protocol (TCP/UDP, etc.) with:**

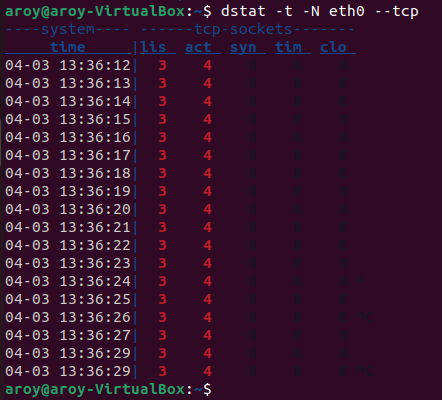
**dstat -t -N eth0 --tcp**

**t for time stamp**

**-N for more readable format**

**eth0 is for which network interface**

**-- tcp is which protocol/udp also works**



HOST

The host command in computer networking is a simple utility used to perform DNS (Domain Name System) lookups. It resolves domain names into IP addresses and vice versa, and is widely used for querying DNS records, troubleshooting DNS issues, and gathering information about domain names and IP addresses.

host [options] <domain> [<server>]

### Common Uses of the host Command:

1. **Basic Domain Name to IP Address Lookup**: To find the IP address associated with a domain name, you can run the following command:

host example.com

This will return the **A record** (IPv4 address) for example.com.

**Reverse Lookup (IP Address to Domain Name)**: To perform a reverse DNS lookup, i.e., find the domain name associated with an IP address, you can use:

host 192.168.1.1

DESCRIPTION

host is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa.

When no arguments or options are given, host prints a short summary of its command-line arguments and options.

aroy@aroy-VirtualBox:~$ man host

aroy@aroy-VirtualBox:~$ host google.com

google.com has address 142.250.207.78

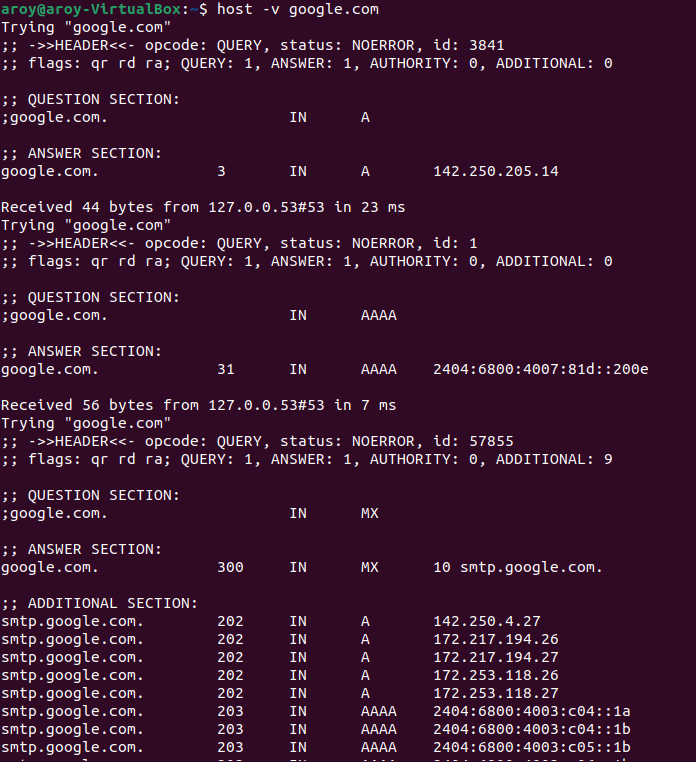
google.com has IPv6 address 2404:6800:4007:81d::200e

google.com mail is handled by 10 smtp.google.com.

aroy@aroy-VirtualBox:~$ host 142.250.207.78

78.207.250.142.in-addr.arpa domain name pointer hkg12s32-in-f14.1e100.net.

78.207.250.142.in-addr.arpa domain name pointer pnmaaa-bd-in-f14.1e100.net.



IFTOP - Chumu command like bmon somewhat a little

iftop - display bandwidth usage on an interface by host

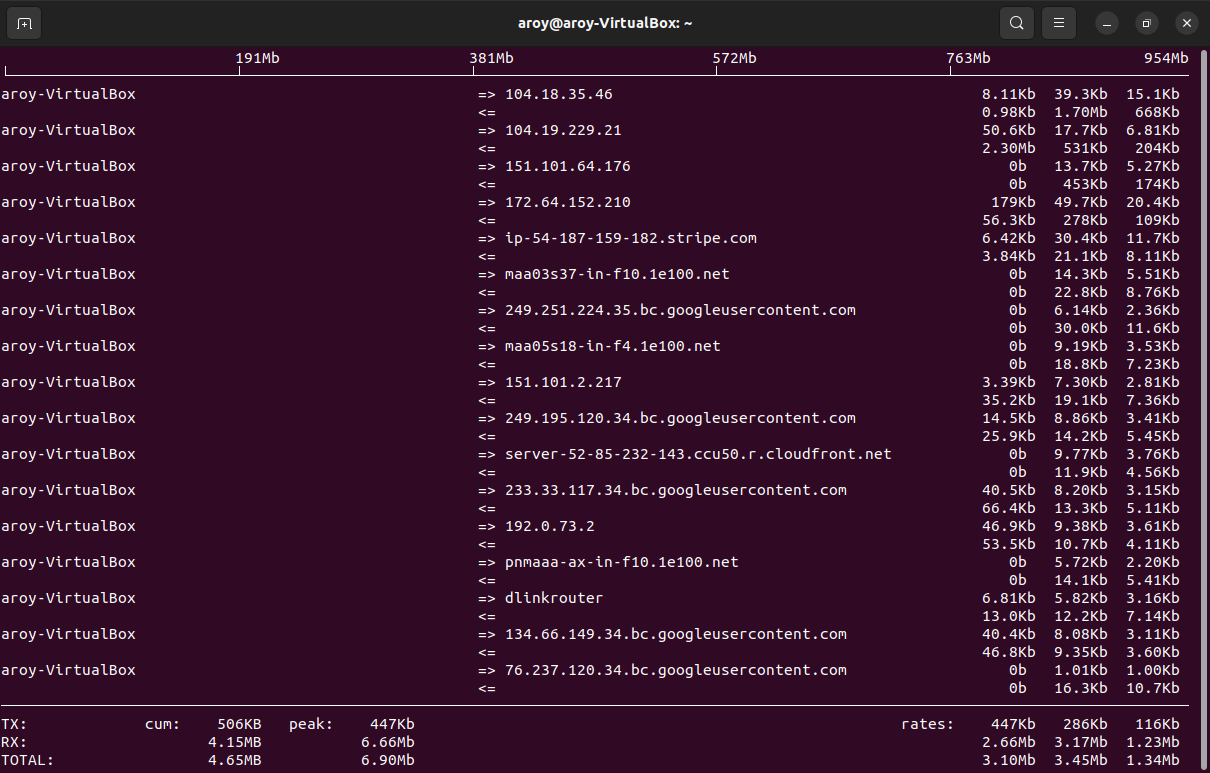
DESCRIPTION

iftop listens to network traffic on a named interface, or on the first interface it can find which looks like an external

interface if none is specified, and displays a table of current bandwidth usage by pairs of hosts. iftop must be run

with sufficient permissions to monitor all network traffic on the interface; see pcap(3) for more information, but on

most systems this means that it must be run as root.



aroy@aroy-VirtualBox:~$ sudo iftop

[sudo] password for aroy:

interface: enp0s3

IP address is: 10.0.2.15

MAC address is: 08:00:27:e8:29:5a

aroy@aroy-VirtualBox:~$

**MAC address and IP address**

### What is a MAC Address?

A **MAC (Media Access Control) address** is a unique identifier assigned to the network interface card (NIC) of a device that connects to a network. It is used to identify the device at the data link layer (Layer 2) of the OSI model.

* **Format**: A MAC address is typically a 48-bit (6-byte) address written in hexadecimal format, often displayed as six pairs of characters separated by colons or hyphens (e.g., 00:1A:2B:3C:4D:5E).
* **Purpose**: The MAC address is used to identify devices on the local network. It ensures that data is delivered to the correct physical device on a network, particularly within Ethernet networks or Wi-Fi networks. MAC addresses are **hardware addresses** that are burned into the network interface hardware (like a NIC or Wi-Fi adapter) at the time of manufacture.

### What is an IP Address?

An **IP (Internet Protocol) address** is a numerical label assigned to each device connected to a network that uses the Internet Protocol for communication. It operates at the **network layer (Layer 3)** of the OSI model.

* **Types**: There are two versions of IP addresses:
  + **IPv4**: A 32-bit address, usually represented as four decimal numbers separated by dots (e.g., 192.168.1.1).
  + **IPv6**: A 128-bit address, usually represented as eight groups of four hexadecimal digits separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
* **Purpose**: The IP address is used to identify devices on a network and to route traffic across different networks. Unlike a MAC address, which is used for local communication, the IP address allows for devices to communicate over larger, more complex networks, such as the Internet.

Differences b/w both:-

* It is a hardware address, specific to the network interface card (NIC) or adapter.
* It operates on the local network level (Layer 2).
* It is **permanent** and hardcoded in the hardware (but can be spoofed).

**IP Address**:

* It is a logical address, used for identifying devices on a network or across networks (Layer 3).
* It can change over time (e.g., when you reconnect to a network, or a device gets a new IP through DHCP).
* It works at a higher level and is used for routing data across the network, including the internet.

**ARP:-**

The arp command in Linux is used to display and manipulate the ARP (Address Resolution Protocol) cache. ARP is a protocol used to map IP addresses to MAC (Media Access Control) addresses on a local network.

Following command shows numerical addresses instead of trying to determine symbolic host, port or user names.

aroy@aroy-VirtualBox:~$ arp -n

Address HWtype HWaddress Flags Mask Iface

10.0.2.2 ether 52:54:00:12:35:02 C enp0s3

To view the current ARP cache, use the following command:

aroy@aroy-VirtualBox:~$ arp -a

\_gateway (10.0.2.2) at 52:54:00:12:35:02 [ether] on enp0s3

aroy@aroy-VirtualBox:~$ arp -av —--more verbose info

\_gateway (10.0.2.2) at 52:54:00:12:35:02 [ether] on enp0s3

Entries: 1 Skipped: 0 Found: 1

**DHCLIENT:-**

dhclient is a command-line utility used to obtain an IP address and other network configuration details from a DHCP (Dynamic Host Configuration Protocol) server. It’s commonly used on Linux and Unix-based systems to manage IP addresses automatically for network interfaces.

< ACCESS IS NOT OBTAINED, EVEN using sudo>

aroy@aroy-VirtualBox:~$ sudo dhclient

Error: ipv4: Address already assigned.

aroy@aroy-VirtualBox:~$

**nload:-**

**nload is a command-line tool used for monitoring network traffic in real-time. It provides a visual representation of incoming and outgoing traffic, showing bandwidth usage on a network interface.**

**Here’s how to use nload effectively:**

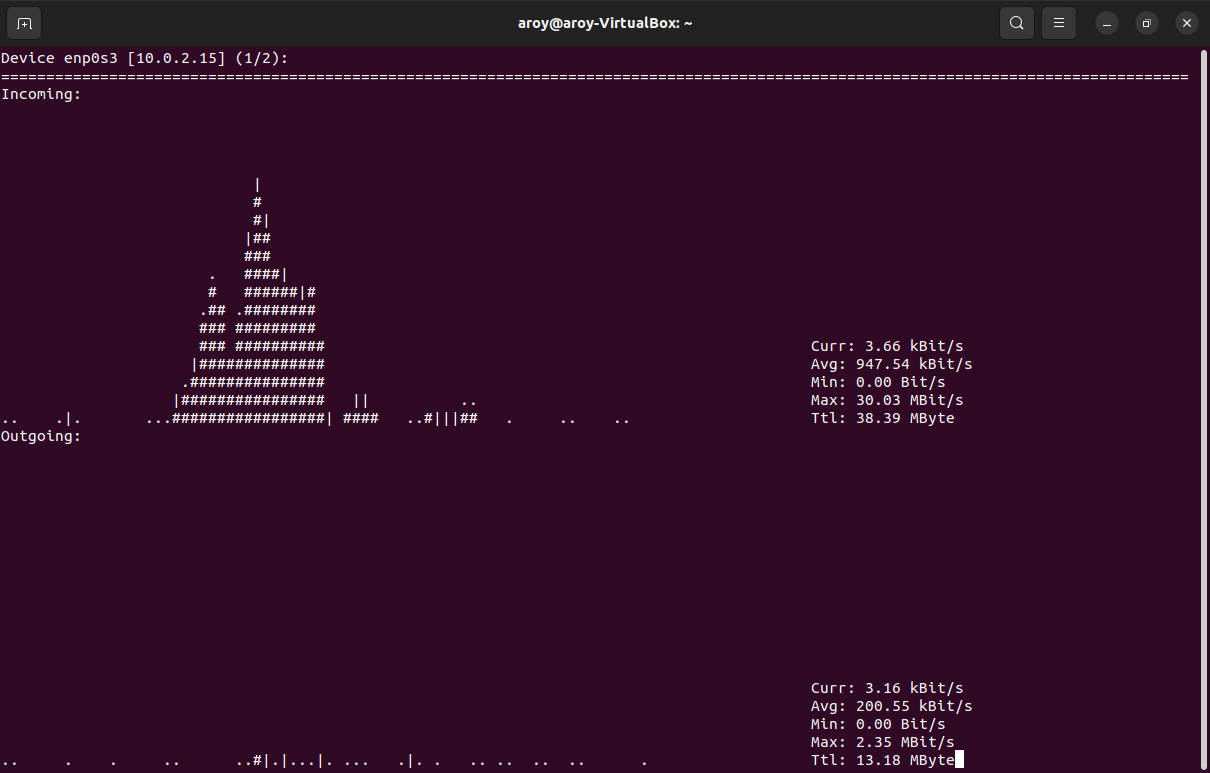
**nload**

**Optional:- specify interface name**

**v for verbose**

**m to mute graph and display all interface details in a single page**

### 



A sample of total usage:-

aroy@aroy-VirtualBox:~$ nload -t 200 -i 1024 -o 128 -U M

[Screencast from 03-04-2025 06:53:17 PM.webm](https://drive.google.com/file/d/1xuKmZhNcSSjbSiwRNdLTcWOEqHJz0tUM/view?usp=sharing)

### Breakdown of the Options:

1. **-t 200**This option sets the **refresh interval** for the graphs (in milliseconds).
   * -t 200 means the display will update every **200 milliseconds**.
   * A smaller value (like 200) means more frequent updates, providing a more real-time view of network activity.
2. **-i 1024**This option specifies the **incoming traffic graph scaling factor**.
   * -i 1024 means that the incoming traffic graph will be scaled by a factor of **1024**.
   * This can be used to adjust the graph's visual scale to make incoming traffic values easier to read, particularly if you're monitoring high traffic volumes.
3. **-o 128**This option sets the **outgoing traffic graph scaling factor**.
   * -o 128 means that the outgoing traffic graph will be scaled by a factor of **128**.
   * Like -i 1024, this adjusts how outgoing traffic is represented in the graph. You can use it to scale down large numbers for outgoing traffic.
4. **-U M**This sets the **unit of measurement** for displaying traffic in the graph.
   * -U M means the unit will be **Mbit/s** (megabits per second).
   * You can use other units like K (kilobits), M (megabits), G (gigabits), k (kilobytes), M (megabytes), etc. In this case, M shows the traffic in megabits per second.

### Putting It All Together:

* **-t 200**: Updates the graphs every 200 milliseconds.
* **-i 1024**: Scales the incoming traffic graph by 1024, likely to make larger values easier to read.
* **-o 128**: Scales the outgoing traffic graph by 128.
* **-U M**: Displays network usage in **Mbit/s** (megabits per second).

### Example Use Case:

This command would be useful if you're monitoring a network with high traffic volumes and want frequent updates (every 200 milliseconds), with customized scaling for incoming and outgoing traffic to make the graphs more readable, and you prefer to view the data in **megabits per second**.

Ifup/ Ifdown

The ifup and ifdown commands in Linux are used to bring a network interface **up** (activate it) or **down** (deactivate it) respectively. These commands are part of the traditional **ifupdown** network management toolset that is still in use on many older or non-systemd-based systems.

### Basic Usage:

1. **ifup Command**:
   * **Purpose**: To bring a network interface up (activate it).
   * This command enables the network interface by configuring it according to the settings in the /etc/network/interfaces file (on Debian-based systems) or equivalent configuration files on other distributions.

**ifup**: Used to enable (activate) a network interface.

**ifdown**: Used to disable (deactivate) a network interface.

route:-

DESCRIPTION

Route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or net‐

works via an interface after it has been configured with the ifconfig(8) program.

When the add or del options are used, route modifies the routing tables. Without these options, route displays the cur‐

rent contents of the routing tables.

aroy@aroy-VirtualBox:~$ route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

default \_gateway 0.0.0.0 UG 0 0 0 enp0s3

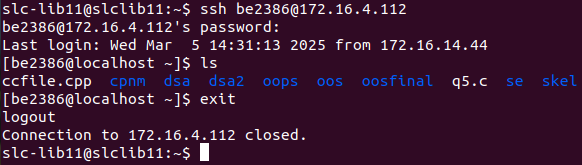
default \_gateway 0.0.0.0 UG 100 0 0 enp0s3

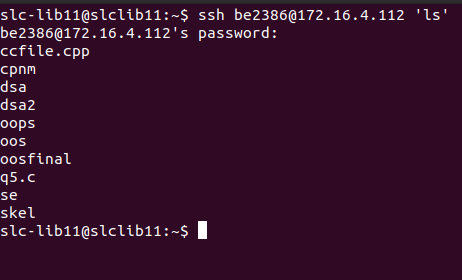
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s3

link-local 0.0.0.0 255.255.0.0 U 1000 0 0 enp0s3

SSH and its Family of commands

ssh:-Using SSH (Secure Shell) in Linux allows you to remotely access another computer securely over a network. Here's how you can use SSH on Linux:-



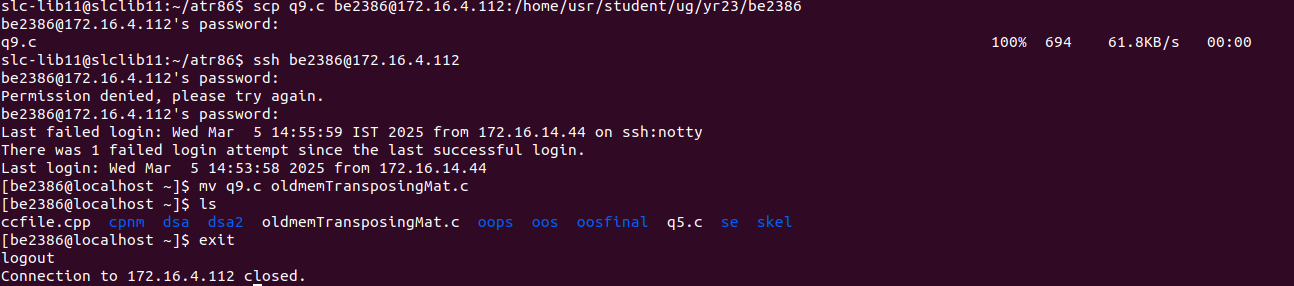


SCP (Secure Copy) is a command-line utility used for securely transferring files between a local machine and a remote machine, or between two remote machines, over SSH. It ensures that your files are transferred securely, leveraging the same SSH encryption.

scp local\_file username@remote\_host:/remote/directory

* **local\_file**: The file you want to copy from your local machine.
* **username**: The username of the remote machine.
* **remote\_host**: The IP address or hostname of the remote machine.
* **/remote/directory**: The directory path where the file will be copied on the remote machine.

scp myfile.txt john@192.168.1.10:/home/john/Documents

  
Copying from remote to local

scp username@remote\_host:/remote/file /local/directory

scp john@192.168.1.10:/home/john/myfile.txt /home/localuser/Desktop

This command will copy myfile.txt from the remote machine to the Desktop of the local machine.

slc-lib11@slclib11:~/atr86$ scp be2386@172.16.4.112:/home/usr/student/ug/yr23/be2386/oldmemTransposingMat.c /home/slc-lib11/atr86

be2386@172.16.4.112's password:

oldmemTransposingMat.c 100% 694 410.7KB/s 00:00

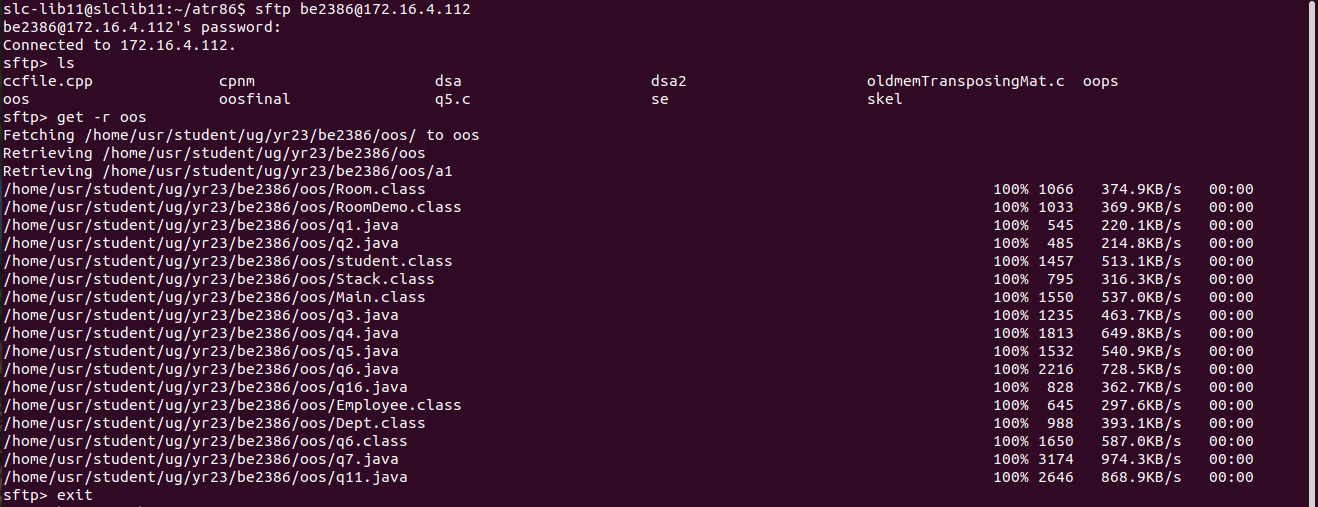
slc-lib11@slclib11:~/atr86$

Copying a Directory

scp -r myfolder john@192.168.1.10:/home/john/Documents

This will copy the entire myfolder from your local machine to the Documents directory on the remote machine.

sftp:- Download files from putty using terminal !!!



SFTP

