BASIC COMMANDS

1.Tcpdump

Tcpdump is a command line utility that allows you to capture and analyze network traffic going through your system.

Procedure

Check if tcpdump is installed on your system

$ which tcpdump

```
/usr/sbin/tcpdump
 If tcpdump is not installed,
$ sudo apt install tcpdump
To get Supervisor Privilege
 $ su
(and password 123456)
$ sudo –i to change #
($ is changed to # and the commands can be executed in supervisor)
```

Capturing packets with tcpdump

Use the command tcpdump -D to see which interfaces are available for capture.

[root@localhost cse]# tcpdump -D

```
1.nflog (Linux netfilter log (NFLOG) interface)
 2.nfqueue (Linux netfilter queue (NFQUEUE) interface)
 3.usbmon1 (USB bus number 1)
 4.enp2s0
5.usbmon2 (USB bus number 2)
 6.any (Pseudo-device that captures on all interfaces)
7.lo [Loopback]
```

Capture all packets in any interface by running this command:

[root@localhost cse]# tcpdump -i any

```
06:03:58.258143 ARP, Request who-has 172.16.51.87 tell 172.16.22.25, length 46
06:03:58.258225 ARP, Request who-has 172.16.51.88 tell 172.16.22.25, length 46
06:03:58.260828 ARP, Request who-has 172.16.51.122 tell 172.16.22.25, length 46
```

```
06:03:58.260903 ARP, Request who-has 172.16.51.123 tell 172.16.22.25, length 46 ^C
5244 packets capture
 59636 packets received by filter
 54378 packets dropped by kernel
```

(Press ctrl+C to stop execution)

Filter packets based on the source or destination IP Address

```
[root@localhost cse]#tcpdump -i any -c5 -nn src 172.16.20.138
6:10:30.712414 ARP, Request who-has 172.16.16.16 tell 172.16.20.138, length 28
06:10:31.483765 IP 172.16.20.138.47997 > 51.158.186.98.123: NTPv4, Client, length 48
5 packets captured
5 packets received by filter
0 packets dropped by kernel
[root@localhost cse]#tcpdump -i any -c5 -nn dst 172.16.20.139
6:10:30.712414 ARP, Request who-has 172.16.16.16 tell 172.16.20.138, length 28
06:10:31.483765 IP 172.16.20.138.47997 > 51.158.186.98.123: NTPv4, Client, length 48
5 packets captured
 5 packets received by filter
 0 packets dropped by kernel
```

Filtering packets

To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using this command:

```
[root@localhost cse]# tcpdump -i any -c5 icmp
```

(tcpdump captures and displays only the ICMP-related packets.)

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
 06:15:07.800786 IP localhost.localdomain > ec2-54-204-39-132.compute-1.amazonaws.com: ICMP echo request, id 8180, seq 13, length 64
 06:15:08.063488 IP ec2-54-204-39-132.compute-1.amazonaws.com > localhost.localdomain: ICMP echo reply, id 8180, seq 13, length 64
5 packets captured
 5 packets received by filter
0 packets dropped by kernel
```

In a different terminal, try to ping another machine:

```
$ ping opensource.com
```

2. netstat

netstat (network statistics) is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc.

```
[root@localhost cse]# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
 tcp 0 0 localhost.localdo:53318 ec2-52-206-98-166:https ESTABLISHED
 tcp 0 0 localhost.localdo:36418 sg2plpkivs-v03.any:http TIME_WAIT
```

-at → list all TCP ports

-au → list all UDP ports

-l → listening ports

-lt → listening TCP

-lu → listening UDP

-s → statistics of all ports

-su →statistics of UDP

-st → statistics of TCP

3. ifconfig

It displays the details of a network interface card like IP address, MAC Address, and the status of a network interface card

```
[cse@localhost ~]$ ifconfig
enp2s0: flags=4163 mtu 1500
 inet 172.16.20.138 netmask 255.255.0.0 broadcast 172.16.255.255
inet6 fe80::d884:13bc:fd22:2d43 prefixlen 64 scopeid 0x20
ether a0:8c:fd:e7:10:86 txqueuelen 1000 (Ethernet)
RX packets 4474083 bytes 280780119 (267.7 MiB)
 RX errors 0 dropped 353 overruns 0 frame 0
 TX packets 14455 bytes 1798944 (1.7 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
 lo: flags=73 mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
 inet6 ::1 prefixlen 128 scopeid 0x10
loop txqueuelen 1000 (Local Loopback)
RX packets 4154 bytes 352264 (344.0 KiB)
```

```
 RX errors 0 dropped 0 overruns 0 frame 0
TX packets 4154 bytes 352264 (344.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

4. nslookup

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.

```
[cse@localhost ~]$ nslookup annauniv.edu
Server: 8.8.8.8
 Address: 8.8.8.8#53
 Non-authoritative answer:
 Name: annauniv.edu
Address: 103.70.60.38
[cse@localhost ~]$ nslookup 172.217.26.206
Server: 8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:
 206.26.217.172.in-addr.arpa name = maa03s23-in-f14.1e100.net.
206.26.217.172.in-addr.arpa name = maa03s23-in-f14.1e100.net.
 206.26.217.172.in-addr.arpa name = maa03s23-in-f206.1e100.net.
 206.26.217.172.in-addr.arpa name = maa03s23-in-f206.1e100.net.
Authoritative traceroute answers can be found from:
```

Lookup for any record

```
 [cse@localhost ~]$ nslookup -type=any annauniv.edu
Server: 8.8.8.8
 Address: 8.8.8.8#53
Non-authoritative answer:
 Name: annauniv.edu
Address: 103.70.60.38
annauniv.edu text = "v=spf1 ip4:103.70.60.40 -all"
 annauniv.edu mail exchanger = 0 sonic.annauniv.edu.
annauniv.edu
origin = ns.annauniv.edu
```

```
mail addr = root.annauniv.edu
 serial = 20170907
 refresh = 300
retry = 900
expire = 604800
 minimum = 86400
annauniv.edu nameserver = ns.annauniv.edu
. Authoritative answers can be found from:
```

Lookup for an ns record

```
 [cse@localhost ~]$ nslookup -type=ns annauniv.edu
Server: 8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:
annauniv.edu nameserver = ns.annauniv.edu.
Authoritative answers can be found from
```

5. traceroute

The traceroute command is used in **Linux t**o map the journey that a packet of information undertakes from its source to its destination.

```
[cse@localhost ~]$ traceroute
Usage: traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl
] [ -N squeries ] [ - p port ] [ -t tos ] [ -l flow_label ] [ -w waittime ] [ -q nqueries ] [ -s
src_addr ] [ -z sendwait ] [ -- fwmark=num ] host [ packetlen ]
 Options:
-4 Use IPv4
-6 Use IPv6
-d --debug Enable socket level debugging
 -F --dont-fragment Do not fragment packets
[cse@localhost ~]$ traceroute annauniv.edu
traceroute to annauniv.edu (103.70.60.38), 30 hops max, 60 byte packets
1 117.193.124.33 (117.193.124.33) 1.389 ms 1.216 ms 1.072 ms
2 172.16.199.74 (172.16.199.74) 1.902 ms 1.834 ms 1.761 ms
3 218.248.235.161 (218.248.235.161) 27.212 ms * *
 4 * * *
 5 218.248.178.42 (218.248.178.42) 15.521 ms * *
```

```
6 * * *
7 madurai-eg-175.232.249.45.powergrid.in (45.249.232.175) 16.007 ms 15.345 ms 15.867 ms
[cse@localhost ~]$ traceroute 172.16.20.139
traceroute to 172.16.20.139 (172.16.20.139), 30 hops max, 60 byte packets
1 localhost.localdomain (172.16.20.138) 3004.348 ms !H 3004.215 ms !H 3004.104 ms !H
```

Capture ping and traceroute PDUs using a network protocol analyzer and examine

Network protocol analyzer – wireshark

Wireshark is free & Open source network packet analyzer that is used for network analysis, troubleshooting, etc.

Wireshark is quite similar to tcpdump, the major difference between the two is that Wireshark has a graphical interface with built-in filtering options, which make it easy to use.

Installation commands on Wireshark

```
# sudo apt install wireshark
```

To Open Wireshark

Open directly or use the following commands

```
# sudo wireshark
```

In wireshark filter icmp packets

In a konsole execute

```
# ping www.sudo.com
# traceroute www.google.com
```