Share:

Linux is the **kernel of an operating system**. It was developed by Linus Torvalds completely from scratch. However, it is a **UNIX clone**. It is developed as open-source software which is inherently very robust and secure. Anybody can install Linux and can make changes to it, creating their own Linux version. These versions of Linux are officially called "**distros**" or "**distributions**". Some of the popular Linux versions are:

> *Want to Enrich your Career Potential as a Linux Developer then Enrol into our* ***Linux Certification*** Course

- Ubuntu Linux
- Red Hat Enterprise Linux
- Linux Mint
- Debian
- Fedora

The unmatched security and stability of the Linux kernel, coupled with the open source nature of it, makes it one of the extensively used software.

Linux is used both in software development and in servers. Most of the devices and embedded systems in the world implement one or the other distribution of Linux. About 90% of the internet servers across the world are Linux servers. And, about **80% of the smartphones** in the world run Linux kernel.

Every system in the world is now connected through a network. Networking amongst systems is essential for the exchange of information. Networking in computers speaks about networking both within the network and across the internet. A network can be as small and simple as a home network or as complex as a network for a space station.

Networking includes network configuration and troubleshooting. In this module, we are learning about the Linux networking commands and network troubleshooting.

Linux Networking Commands

Linux networking commands are used extensively to inspect, analyze, maintain and troubleshoot the network/s connected to the system.

Let us first know the list of the basic networking commands used in Linux followed by a detailed explanation of each.

1. ifconfig
2. ip
3. traceroute
4. tracepath
5. ping
6. netstat
7. ss
8. dig
9. nslookup
10. route
11. host
12. arp
13. iwconfig
14. hostname
15. curl or wget
16. mtr
17. whois
18. ifplugstatus
19. iftop
20. tcpdump

Details of Linux Networking Commands

ifconfig

Linux ifconfig stands for interface configurator. It is one of the most basic commands used in network inspection.

ifconfig is used to initialize an interface, configure it with an IP address and enable or disable it. It is also used to display the route and the network interface.

Basic information displayed upon using ifconfig are:

1. IP address
2. MAC address
3. MTU(Maximum Transmission Unit)

- To get all the details using ifconfig

Syntax:

```
Ifconfig
```

Output:

This picture shows the IP address of 3 networks, Ethernet, local network and WLAN.

- To get details of specific interface

Using this command, you can get details of a specific interface. This is shown below.

Commands:

```
ifconfig eth0
ifconfig lo
ifconfig wlan0
```

- To assign an IP address and Gateway to an interface

This command can also be used to assign IP address and Gateway to an interface. However, these details will be reset after the system reboot.

Important Linux Commands

Syntax:

```
ifconfig eth0 <address> netmask <address>
```

- To enable or disable an interface

ifconfig can be used to enable or disable an interface.

To enable an interface

Syntax:

```
ifup eth0
```

- To disable an interface

Syntax:

```
ifdown eth0
```

- To set the size of MTU

By default, MTU has a size of 1500. This can be however set externally by the user using ifconfig.

Syntax:

```
Ifconfig eth0 mtu xxxx
```

xxxx can be replaced by the size of your choice.

ip

This is the latest and updated version of ifconfig command.

Syntax:

1. ip a
2. ip addr

This command gives the details of all networks like ifconfig.

This command can also be used to get the details of a specific interface.

Commands to get details are:

Syntax:

```
ip a show eth0
ip a show lo
ip a show wlan0
```

traceroute

Linux traceroute is one of the most useful commands in networking. It is used to troubleshoot the network. It detects the delay and determines the pathway to your target. It basically helps in the following ways:

1. It provides the names and identifies every device on the path.
2. It follows the route to the destination
3. It determines where the network latency comes from and reports it.

```
traceroute <destination>
```

If you don't have the traceroute service installed in your system, you can install it using the following command:

```
sudo apt-get install inetutils-traceroute
```

Example:

Command:

```
$ traceroute google.com
```

The output provides the following information:

1. The specified hostname
2. Size of the packets
3. The maximum number of the hops required.
4. The IP address.

- To avoid the reverse DNS lookup, add -n in the command syntax.

Command:

```
$ traceroute -n google.com
```

The output indicates the network delays. The asterisks shown in the output indicates a potential problem in reaching that host. They indicate the packet loss during communication to the network.

Generally, the traceroute command sends UDP packets. It can as well send TCP or ICMP packets.

- To specifically send in ICMP, use this,

Command:

```
$ sudo traceroute -I google.com
```

To send a variant of TCP, use this,

Command:

```
$ sudo traceroute -T google.com
```

tracepath

Linux tracepath is similar to traceroute command. It is used to detect network delay. However, it doesn't require root privileges.

It is installed in Ubuntu by default.

It traces the route to the specified destination and identifies each hop in it. If your network is weak, it recognizes the point where the network is weak.

Syntax:

```
tracepath <destination>
```

Example:

```
tracepath mindmajix.com
```

Output:

ping

Linux ping is one of the most used network troubleshooting commands. It basically checks for the network connectivity between two nodes.

ping stands for Packet INternet Groper.

The ping command sends the ICMP echo request to check the network connectivity.

It keeps executing until it is interrupted.

Use **Ctrl+C Key** to interrupt the execution.

Syntax:

```
ping <destination>
```

Example:

Command:

```
$ ping google.com
```

The ping shows a successful connection to google.com

You can also use the IP address to ping directly.

You can limit the number of packets by including "**-c**" in the ping command.

Syntax:

```
ping -c <number> <destination>
```

You can specify the c count and limit the response packets to that.

Functions:

The command is used to measure the average response. If there is no response for ping command, you can assume one of the following issues with the network:

- There is a physical issue causing network loss.
- The destination address might be dysfunctional or incorrect.
- The ping request is blocked due to a target.
- There might be a problem with the routing table.

Note: The response rate of the ping command will be affected by the connection at your system and also the location of the server you are pinging to. So expect a delay in the response if the connection at your point is weak.

netstat

Linux netstat command refers to the network statistics.

It provides statistical figures about different interfaces which includes open sockets, routing tables and connection information.

Syntax:

```
netstat
```

Output:

Observe the output displaying all the open sockets.

Variations in netstat command

Below are few variations of the netstat command used.

1) To display the programs

Syntax:

```
netstat -p
```

This displays the programs associated with the open socket.

2) To get the details of the ports

Syntax:

```
netstat -s
```

This gives detailed statistics of all the ports.

3) To get the information of the routing table

Syntax:

```
netstat -r
```

This gives information related to the routing table.

ss

Linux ss command is the replacement for netstat command. It is regarded as a much faster and more informative command than netstat.

The faster response of ss is possible as it fetches all the information from within the kernel userspace.

Syntax:

```

```

This command gives information about all TCP, UDP, and UNIX socket connections.

You can use -t , -u, -x in the command respectively to show TCP/UDP or UNIX sockets. You can combine each of this with "a" to show the connected and listening sockets.

Syntax:

```
ss -ta
ss -ua
ss -xa
```

If you want to see only the listening sockets of TCP/UDP or UNIX sockets, combine it with "l"

Syntax:

```
ss -lt
ss -lu
ss -lx
```

- To get a list of all the established sockets of TCP for IPV4,

Command:

```
$ ss -t4 state established
```

- To get a list of all closed TCP sockets,

Command:

```
$ ss -t4 state closed
```

- To get a list of all connected ports for a specific IP address:

Command:

```
$ ss dst XXX.XXX.XXX.XXX
```

dig

Linux dig command stands for Domain Information Groper. This command is used in DNS lookup to query DNS name server. It is also used to troubleshoot DNS related issues.

It is mainly used to verify DNS mappings, MX Records, host addresses and all other DNS records for a better understanding of the DNS topography.

This command is an improvised version of nslookup command.

Syntax:

```
dig <domainName>
```

Example:

```
$ dig google.com
```

Output:

dig command outputs the A records by default. If you want to specifically search for MX or NS type, use the syntax below.

Command:

```
$ dig google.com MX
```

- To get all types of records at once, use the keyword ANY ass below:

Command:

```
$ dig google.com ANY
```

The dig command does the query on the servers listed in /etc/resolv.conf.

nslookup

Linux nslookup is also a command used for DNS related queries. It is the older version of dig.

Syntax:

```
nslookup <domainName>
```

Example:

```
nslookup mindmajix.com
```

Output:

As we see in the output above, it displays the record information relating to

```
mindmajix.com
```

route

Linux route command displays and manipulates the routing table existing for your system.

A router is basically used to find the best way to send the packets across to a destination.

Syntax:

```
route
```

Output:

The above output displays all the existing routing table entries for the system. It says that if the destination address is within the network range of 10.0.0.0 to 10.0.0.255, then the gateway is *, which 0.0.0.0. This is a special address which indicates a non-existent destination.

The packets which lie outside this network range will be forwarded to the default gateway, which is further routed.

- Displaying numerical IP address

You can use -n in the option in the syntax to display the output in complete numerical form.

Syntax:

```
route -n
```

- To add a gateway

The packets that are not within the range are forwarded to the specific gateway. You can specify the gateway address using the following command.

Syntax:

```
route add default gw <IP address>
```

- To get routing information

The kernel maintains all the routing cache information in a table for faster routing. To list the routing cache information, use the following command,

Syntax:

```
route -Cn
```

host

Linux host command displays the domain name for a given IP address and IP address for a given hostname. It is also used to fetch DNS lookup for DNS related query.

Example:

```
host mindmajix.com
host 149.77.21.18
```

You can combine the host command with -t, and get DNS resource records like SOA, NS, A, PTR, CNAME, MX, SRV.

Syntax:

```
host -t <resourceName>
```

arp

Linux arp command stands for Address Resolution Protocol. It is used to view and add content to kernel's ARP table.

Syntax:

```
arp
```

All the systems maintain a table of IP addresses and their corresponding MAC addresses. This table is called the ARP Lookup table. When a destination is requested to connect through IP address, your router will check for the MAC address in this table. If it is cached, the table will not be used.

By default, arp displays the hostnames. You can get the IP addresses, by using :

Command:

```
$ arp -n
```

You can also delete the entries from the arp table, as shown below.

Command:

```
$ arp -d HWADDR
```

Frequently asked Linux Interview Questions

iwconfig

Linux iwconfig is used to configure the wireless network interface. It is used to set and view the basic WI-FI details like SSID and encryption. To know more about this command, refer to the man page.

Syntax:

```
iwconfig
```

Output:

hostname

Linux hostname is the simple command used to view and set the hostname of a system.

Syntax:

```
hostname
```

Output:

- To set the hostname

Use the syntax below to set the hostname.

Syntax:

```
sudo hostname <newName>
```

The hostname set through this command is not permanent. It will be reset to the name in hostname file back when the system reboots.

In order to permanently set a hostname, you have to re-write the hostname in hostname file, present on the server. Once set, you have to reboot the box.

In Ubuntu, /etc/hostname file is used.

In RHEL, /etc/sysconfig/network is used.

curl & wget

Linux curl and wget commands are used in downloading files from the internet through CLI. The curl command has to be used with an option "O" to fetch the file, while the wget command is used directly.

Below are the syntax and the example for the two commands.

a) curl

Syntax:

```
curl -O <fileLink>
```

Example:

```
curl -O google.com/doodles/childrens-day-2014-multiple-countries
```

b) wget

Syntax:

```
wget <fileLink>
```

Example:

```
wget google.com/doodles/new-years-day-2012
```

Output:

mtr

Linux mtr command is a combination of ping and the traceroute command. It continuously displays information regarding the packets sent with the ping time of each hop. It is also used to view the network issues.

Syntax:

```
mtr <path>
```

Example:

```
$ mtr google.com
```

Output:

You can use mtr with –report option. It sends 10 packets to each hop that is found on the way.

Syntax:

```
$ mtr --report <path>
```

whois

Linux whois command is used to fetch all the information related to a website. You can get all the information about a website including the registration and the owner information.

Syntax:

```
whois <websiteName>
```

Example:

```
whois mindmajix.com
```

Output:

ifplugstatus

Linux ifplugstatus command is used to check if a cable is plugged into the network interface. This command is not directly available on Ubuntu. You can install this using the command below:

Command:

```
sudo apt-get install ifplugd
```

Syntax:

```
ifplugstatus
```

Output:

In the output above, "link beat detected" means that the cable is plugged in.

iftop

Linux iftop command is used in traffic monitoring.

Use the following command to download iftop on your system.

Command:

```
$ wget http://www.ex-parrot.com/pdw/iftop/download/iftop-0.17.tar.gz
```

This will give a zip file. To extract it, use the following command,

Command:

```
$  tar zxvf iftop-0.17.tar.gz
```

You can compile this using,

Commands:

```
$ cd iftop-0.17
$  ./configure
$ make
$ make install
```

Now, run the tool as a root user,

```
$ sudo iftop -I <interface>
```

Output:

You can view the ports using the -P option in command like this,

Command:

```
$ sudo iftop -P
```

You can use -B command to get the data in bytes, instead of bits (which is shown by default).

Command:

```
$ iftop -B
```

tcpdump

Linux tcpdump command is the most used command in network analysis among other Linux network commands. It captures the traffic that is passing through the network interface and displays it.

This kind of access to the packet will be crucial when troubleshooting the network.

Syntax:

```
$ tcpdump -i <network_device>
```

Output:

You can also specify the protocol (TCP, UDP, ICMP, and others) in the command like this,

Command:

```
$ tcpdump -i <network_device> tcp
```

To specify the port, use the command,

Command:

```
$ tcpdump -i <network_device> port 80
```

tcpdump command keeps executing and sending packets unless canceled. Hence you can specify the number of events to be captured to control the continuous execution.

Command:

```
$ tcpdump -c 20 -i <network_device>
```

You can also specify the IP you are capturing from, using the tag src or dst.

Command:

```
$ tcpdump -c 20 -i <network_device> src XXX.XXX.XXX.XXX
```

You can save the network traffic captured at an instant, into a file and use it later. This can be done using the command below,

a) Save into a file

Command:

```
$ tcpdump -w /path/ -i <network_device>
```

b) Read from the file

Command:

```
$ tcpdump -r /path
```

These were the most essential network commands in Linux that are used frequently for network analysis and troubleshooting.