

Cyber Security And Ethical Hacking

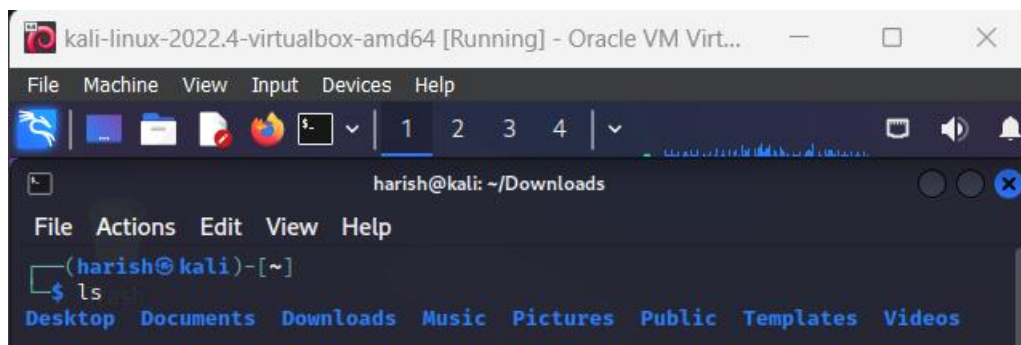
Assignment-1: Linux Command List

Name:P.Harish Chowdary

Reg.No:20BCD7052

A. File and Directory Operations:

1. **ls** (used to list all the folders and files in the current directory)



Here in the home directory of the kali linux, these are the folderes that are present and each folder contains different files.

2. **cd** (Used to enter into the folder present in the current directory to perform operations on the files present in the folder)



Here using the cd command I have navigated to the Downloads folder.

Note: Names of files and folders are case sensitive in linux and we have to enter the exact combination of words to open a file or the folder.

3. **pwd** (Displays the current directory and the folder we are currently in)



As we changed the directory to the downloads, we can see the path to it.

4. **mkdir** (Creates a new directory in the current directory)

```
(harish@kali)-[~/Downloads]
$ mkdir testdirectory

(harish@kali)-[~/Downloads]
$ ls
sqlinj.pdf  testdirectory
```

Here we have created a new directory called “testdirectory” and we can see that it is present using the “ls” command.

5. touch (Creates an empty file)

```
(harish@kali)-[~/Downloads]
$ touch testfile

(harish@kali)-[~/Downloads]
$ ls
sqlinj.pdf  testdirectory  testfile
```

Here we have made a new empty file called “testfile” and we can see that it is present using the “ls” command.

```
(harish@kali)-[~/Downloads]
$ nano testfile

(harish@kali)-[~/Downloads]
$ cat testfile
hello this is harish and currently in the "testfile" file once we use the cp
the contents of this file into another file named "testfile2"
```

Some content was added to the file using the “nano” command and is displayed using the “cat” command. So that we can do the next steps.

6. cp (Used to copy the contents of source file to destination file)

```
(harish@kali)-[~/Downloads]
$ touch testfile2

(harish@kali)-[~/Downloads]
$ cat testfile2

(harish@kali)-[~/Downloads]
$ cp testfile testfile2

(harish@kali)-[~/Downloads]
$ cat testfile2
hello this is harish and currently in the "testfile" file once we use the cp
command, we can copy
the contents of this file into another file named "testfile2"
```

For this a new empty file named “testfile2” was created and the contents of the “testfile” are copied to the “testfile2”

7. mv (This command is used to move the files or directories from the source directory to the destination directory)

```
(harish@kali)-[~/Downloads]
$ mv testfile testdirectory

(harish@kali)-[~/Downloads]
$ ls
sqlinj.pdf  testdirectory  testfile2

(harish@kali)-[~/Downloads]
$ cd testdirectory

(harish@kali)-[~/Downloads/testdirectory]
$ ls
testfile
```

Here “testfile” is moved from the Downloads directory to the “testdirectory” and the same is shown using the ‘ls’ command.

8. rm (This command is used to remove the file or directory.)

```
(harish@kali)-[~/Downloads/testdirectory]
$ rm testfile

(harish@kali)-[~/Downloads/testdirectory]
$ ls
```

We have removed the testfile from the testdirectory.

9. find (This command is used to find the files or directories in the current working directory)

```
(harish@kali)-[~/Downloads]
$ find testfile
find: 'testfile': No such file or directory

(harish@kali)-[~/Downloads]
$ find testfile2
testfile2
```

We moved the “testfile” and then we delete so we cannot find it but we can find the “testfile2” in the current directory.

B. File viewing and editing

1. **cat** (This command is used to concatenate and view the contents of the file)

```
(harish@kali)-[~]
$ cd Downloads

(harish@kali)-[~/Downloads]
$ cat testfile2
hello this is harish and currently in the "testfile" file once we use the cp command
the contents of this file into another file named "testfile2"

(harish@kali)-[~/Downloads]
$
```

We created the testfile2 above and the contents of the file are viewed using the 'cat' command

2. **less** (view the file content with pagination)

```
(harish@kali)-[~/Downloads]
$ less testfile2

hello this is harish and currently in the "testfile" file once we use the cp comma
nd, we can copy
the contents of this file into another file named "testfile2"
testfile2 (END)
```

In this way less command is used and we can navigate through the contents of the file

3. **head** (This command is used to print the head of the file and we can even specify the number of lines we want to print from the beginning.)

```
(harish@kali)-[~/Downloads]
$ cat testfile2
This is the beggining of the file.

hello this is harish and currently in the "testfile" file once we use the cp comma
nd, we can copy
the contents of this file into another file named "testfile2"

This is the end of the file.

(harish@kali)-[~/Downloads]
$ head -n 1 testfile2
This is the beggining of the file.
```

We can see the difference using the cat and head command.

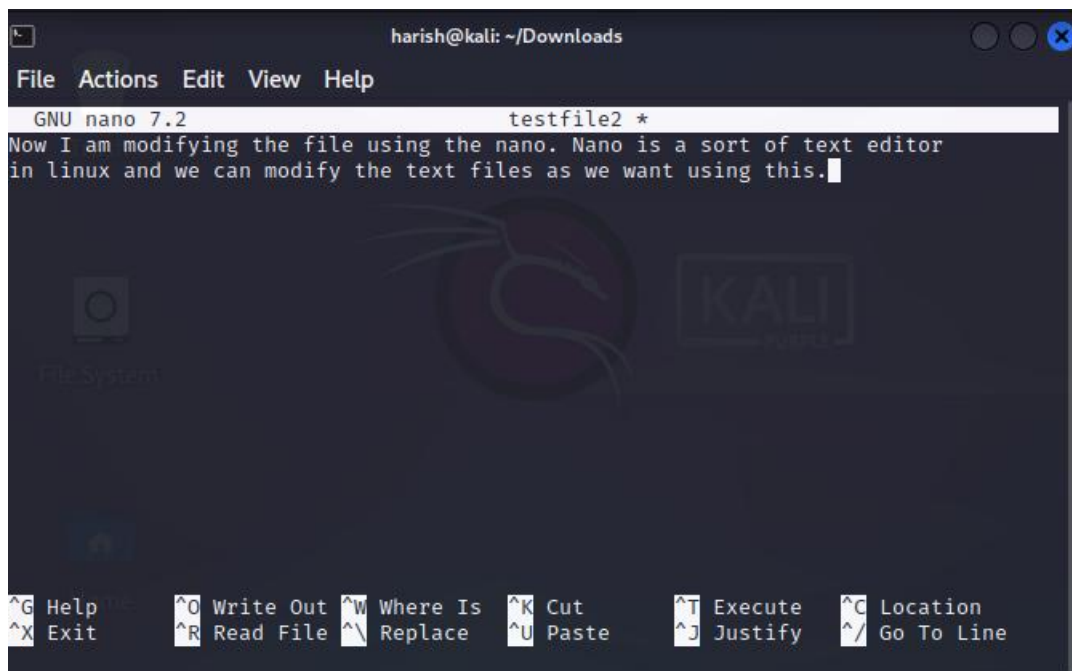
4. **tail** (Display the end of the file and we can mention how many lines we want to display from the end.)

```
(harish@kali)-[~/Downloads]
$ tail -n 1 testfile2
This is the end of the file.
```

We can see the difference from the cat command.

5. **nano** (Text editor for creating and editing files.)

```
(harish@kali)-[~/Downloads]
$ nano testfile2
```



```
(harish@kali)-[~/Downloads]
$ cat testfile2
Now I am modifying the file using the nano. Nano is a sort of text editor
in linux and we can modify the text files as we want using this.
```

In this way we can use nano command to view and modify the contents of the file and we can see that the contents of the testfile2 have been changed.

C. File permissions.

1. **chmod** (this command is used to change the permissions of file or directory so that only a particular group or a particular person will have the permissions to view or edit or execute the file.)

```
(harish@kali)-[~]  
$ cd Downloads  
  
(harish@kali)-[~/Downloads]  
$ ls -l  
total 4472  
-rw-r--r-- 1 harish harish 4569662 Feb  8 17:00 sqlinj.pdf  
drwxr-xr-x 2 harish harish  4096 May 30 11:31 testdirectory  
-rw-r--r-- 1 harish harish    139 May 30 13:10 testfile2
```

```
(harish@kali)-[~/Downloads]  
$ chmod 644 testdirectory  
  
(harish@kali)-[~/Downloads]  
$ ls -l  
total 4472  
-rw-r--r-- 1 harish harish 4569662 Feb  8 17:00 sqlinj.pdf  
drw-r--r-- 2 harish harish  4096 May 30 11:31 testdirectory  
-rw-r--r-- 1 harish harish    139 May 30 13:10 testfile2
```

Ear

each write, read, and execute permissions have the following number value

r (read) = 4

w (write) = 2

x (execute) = 1

no permissions = 0

To find out the file's permissions in numeric mode simply calculate the totals for all users classes. For example, to give read, write and execute permission to the file's owner, read and execute permissions to the file's group and only read permissions to all other users you would do the following:

Owner: $rw\!x=4+2+1=7$

Group: $r\!-x=4+0+1=5$

Others: $r\!-x=4+0+0=4$

So in our example we have given 644 to the testdirectory and we can see that the execution of the file permission to the harish(owner) is lost and only viewing permission is given to other groups and all other users.

2. chown (this command is used to change the owner of the file)

```
(harish@kali)~/Downloads
$ ls -l
total 4472
-rw-r--r-- 1 harish harish 4569662 Feb  8 17:00 sqlinj.pdf
drw-r--r-- 2 harish harish  4096 May 30 11:31 testdirectory
-rw-r--r-- 1 harish harish    139 May 30 13:10 testfile2

(harish@kali)~/Downloads
$ sudo chown kali testfile2

(harish@kali)~/Downloads
$ ls -l
total 4472
-rw-r--r-- 1 harish harish 4569662 Feb  8 17:00 sqlinj.pdf
drw-r--r-- 2 harish harish  4096 May 30 11:31 testdirectory
-rw-r--r-- 1 kali  harish    139 May 30 13:10 testfile2
```

We can see that the ownership of the file testfile2 has changed from harish to kali.

3. chgrp (This command is used to change the group.)

```
(harish@kali)~/Downloads
$ ls -l
total 4472
-rw-r--r-- 1 harish harish 4569662 Feb  8 17:00 sqlinj.pdf
drw-r--r-- 2 harish harish  4096 May 30 11:31 testdirectory
-rw-r--r-- 1 kali  harish    139 May 30 13:10 testfile2

(harish@kali)~/Downloads
$ sudo chgrp kali testfile2

(harish@kali)~/Downloads
$ ls -l
total 4472
-rw-r--r-- 1 harish harish 4569662 Feb  8 17:00 sqlinj.pdf
drw-r--r-- 2 harish harish  4096 May 30 11:31 testdirectory
-rw-r--r-- 1 kali  kali    139 May 30 13:10 testfile2
```

We can see the group of testfile2 has changed from harish to kali.

D. File compression and Archiving

1. **tar** (This command is used to archive the files present in the directory.)

```
(harish@kali)-[~/Downloads]
$ tar -cf testfile2.tar testfile2

(harish@kali)-[~/Downloads]
$ ls
sqlinj.pdf  testdirectory  testfile2  testfile2.tar

(harish@kali)-[~/Downloads]
$
```

We have created an archived folder using the following command where testfile2 have been archived.

2. **gzip** (This command is used to compress the file size.)

```
(harish@kali)-[~/Downloads]
$ gzip -v testfile2
testfile2: 29.5% -- replaced with testfile2.gz

(harish@kali)-[~/Downloads]
$ ls
sqlinj.pdf  testdirectory  testfile2.gz  testfile2.tar

(harish@kali)-[~/Downloads]
$
```

We can see that testfile2 have been compressed by 29.5 percent and have been replaced by testfile2.gz.

3. **unzip** (This command is used to unzip the files from the zip archive.)

We have to download an zip file and then we can unzip the file contents using the unzip command it works similar to winrar software in the windows. And there are a lot of options available.

E. Process Management

1. **ps** (this command is used to list all the running processes.)

```
(harish@kali)-[~/Downloads]
$ ps
  PID TTY          TIME CMD
 15794 pts/0    00:00:00 bash
 36461 pts/0    00:00:00 ps
```

2. **top** (display realtime system information and processes.)

```
top - 13:59:05 up 1:18, 1 user, load average: 0.23, 0.24, 0.19
Tasks: 167 total, 1 running, 166 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 2.5 sy, 0.0 ni, 95.3 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
MiB Mem : 3093.5 total, 1390.8 free, 1200.1 used, 696.4 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used, 1893.3 avail Mem

   PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
    615 root        20   0  444148 136524 84536 S   5.0   4.3   1:26.16 Xorg
   1095 harish     20   0 1014860 112620 77828 S   1.0   3.6   0:27.57 xfwm4
   1156 harish     20   0  469304  42860 34408 S   1.0   1.4   0:05.30 panel-16-+
   1032 harish     20   0  217956   2388  2040 S   0.7   0.1   0:15.66 VBoxClient
   1153 harish     20   0  425808  36032 21820 S   0.7   1.1   0:20.90 panel-13-+
  15791 harish     20   0  440304 103608 83836 S   0.7   3.3   0:06.09 qterminal
 35606 harish     20   0 2976832 315096 159208 S   0.7   9.9   0:13.56 firefox-e+
    15 root        20   0         0         0      0 I   0.3   0.0   0:04.76 rcu_preem+
   177 root       -51   0         0         0      0 S   0.3   0.0   0:03.32 irq/18-vm+
   1023 harish     20   0  217440   2364  2016 S   0.3   0.1   0:05.97 VBoxClient
   1134 harish     20   0  550876  46284 35628 S   0.3   1.5   0:02.13 xfce4-pan+
   1146 harish     20   0  744180 126224 50280 S   0.3   4.0   0:04.88 xfdesktop
   1155 harish     20   0  423620  30232 20884 S   0.3   1.0   0:15.96 panel-15-+
 22579 root        20   0         0         0      0 I   0.3   0.0   0:03.92 kworker/1+
```

With this we have listed all the processes that are running in the system.

3. **kill** (This command is used to kill a processes that is running.)

```
(harish@kali)-[~/Downloads]
$ ps
  PID TTY          TIME CMD
 15794 pts/0    00:00:00 bash
 40326 pts/0    00:00:00 ps

(harish@kali)-[~/Downloads]
$ kill 15794

(harish@kali)-[~/Downloads]
$
```

4. bg and fg (used to run the processes in background and foreground)

```
(harish@kali)-[~]
$ ping google.com
PING google.com(maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e)) 56 data bytes
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=1 ttl=58 time=17.1 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=2 ttl=58 time=27.2 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=3 ttl=58 time=12.8 ms
^Z
[1]+  Stopped                  ping google.com

(harish@kali)-[~]
$ jobs
[1]+  Stopped                  ping google.com

(harish@kali)-[~]
$ ps -T
  PID     SPID TTY          TIME CMD
 15794   15794 pts/0        00:00:00 bash
  45785   45785 pts/0        00:00:00 ping
  45850   45850 pts/0        00:00:00 ps

(harish@kali)-[~]
$ bg 1
[1]+ ping google.com &

(harish@kali)-[~]
$ 64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=4 ttl=58 time=20.2 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=5 ttl=58 time=26.0 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=6 ttl=58 time=16.7 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=7 ttl=58 time=24.9 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=8 ttl=58 time=13.1 ms
^C

(harish@kali)-[~]
$ 64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=9 ttl=58 time=13.2 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=10 ttl=58 time=13.6 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=11 ttl=58 time=17.6 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=12 ttl=58 time=17.7 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=13 ttl=58 time=20.5 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=14 ttl=58 time=58.8 ms
g
ping google.com
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=15 ttl=58 time=13.5 ms
64 bytes from maa05s16-in-x0e.1e100.net (2404:6800:4007:817::200e): icmp_seq=16 ttl=58 time=16.9 ms
^C
  google.com ping statistics ---
 16 packets transmitted, 16 received, 0% packet loss, time 28430ms
 rtt min/avg/max/mdev = 12.780/20.611/58.753/10.824 ms

(harish@kali)-[~]
$
```

We have created a new job ping google.com

Then we have taken the job number which is in [] and then we started running the program in the background using the bg

We can see the results of the ping and even we terminated the processes using ctrl + c it didn't stop

Then we used fg to bring the processes to foreground and then we have terminated the processes.

F. System Information

1. uname (used to display the operating system info.)

```
(harish@kali)-[~]  
$ uname  
Linux
```

2. df (Display the disk space usage)

```
(harish@kali)-[~]  
$ df  
Filesystem      1K-blocks    Used Available Use% Mounted on  
udev             1542732         0   1542732   0% /dev  
tmpfs             316776      1008    315768   1% /run  
/dev/sda1        82083148 17694732 60172868 23% /  
tmpfs            1583864         0   1583864   0% /dev/shm  
tmpfs              5120         0     5120   0% /run/lock  
tmpfs            316772         84    316688   1% /run/user/1001
```

3. free (used to display the memory usage.)

```
(harish@kali)-[~]  
$ free  
              total        used        free      shared  buff/cache   available  
Mem:           3167728      1461232      1160332         54744        760508      1706496  
Swap:          1048572           0       1048572
```

4. Uptime (used to display the uptime of the system.)

```
(harish@kali)-[~]  
$ uptime  
14:22:01 up 1:41, 1 user, load average: 0.26, 0.32, 0.33
```

5. who (Displays who is logged in to the system.)

```
(harish@kali)-[~]  
$ who  
harish    tty7          2023-05-30 12:41 (:0)
```

6. w (Displays the logged in users and their activities.)

```
(harish@kali)-[~]  
$ w  
14:22:08 up 1:41, 1 user, load average: 0.32, 0.33, 0.34  
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT  
harish    tty7          :0           12:41    2:35m  2:34   0.45s xfce4-session
```

G. Networking

1. ifconfig (used to display all the network adapters and the assigned ip addresses of the system and to configure the network interfaces of the system.

```
(harish@kali)-[~]
$ ifconfig
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.15 netmask 255.255.255.0 broadcast 10.0.3.255
    inet6 fe80::2e1:b22d:51a3:c6bc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:06:98:75 txqueuelen 1000 (Ethernet)
    RX packets 601 bytes 186512 (182.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 571 bytes 66308 (64.7 KiB)
    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 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Here we can see that the eth0 adapter is on and then the assigned ip of the adapter is 10.0.3.15

2. ping (this command is used to send the ICMP echo requests the specified network host.)

```
(harish@kali)-[~]
$ ping google.com
PING google.com (142.250.195.78) 56(84) bytes of data:
64 bytes from maa03s38-in-f14.1e100.net (142.250.195.78): icmp_seq=1 ttl=116 time=15.0 ms
64 bytes from maa03s38-in-f14.1e100.net (142.250.195.78): icmp_seq=2 ttl=116 time=17.5 ms
64 bytes from maa03s38-in-f14.1e100.net (142.250.195.78): icmp_seq=3 ttl=116 time=17.7 ms
64 bytes from maa03s38-in-f14.1e100.net (142.250.195.78): icmp_seq=4 ttl=116 time=17.2 ms
64 bytes from maa03s38-in-f14.1e100.net (142.250.195.78): icmp_seq=5 ttl=116 time=17.8 ms
64 bytes from maa03s38-in-f14.1e100.net (142.250.195.78): icmp_seq=6 ttl=116 time=17.9 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5025ms
rtt min/avg/max/mdev = 14.972/17.196/17.942/1.023 ms

(harish@kali)-[~]
$
```

Here we have pinged google.com and then the stats of the eco requests have been displayed on the screen.

3. ssh - this command is used to connect to the remote linux servers in the most secure way. To demnstrate we need 2 linux system where one is running as the openssh server and the other is running openssh client software.

4. scp - this command is used to securely copy files between the systems where the ssh connection is established.

5. wget - this command is used to download the files from the web. Now we don't have an executable binaries as present in windows to install the software in linux distributions and to install any software from web we have to use the wget command and then download the binaries into the system.

H. System Administration

1. sudo - Execute commands with the superuser privileges ie root permissions

```
(harish@kali)-[~]
$ sudo su
[sudo] password for harish:
(root@kali)-[/home/harish]
#
```

Here we can run the commands with the root privileges.

2. apt-get - package management for Debian-based distributions

```
(root@kali)-[/home/harish]
# apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.2 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [44.6 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [172 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [217 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [928 kB]
Fetched 65.4 MB in 18s (3,595 kB/s)
Reading package lists ... Done
(root@kali)-[/home/harish]
#
```

Using this command we have updated the package list and we can install the required packages from it using 'sudo apt-get install package name.'

3. yum - it is a package management for the Red Hat based distributions.

4. Systemctl - manage the system services such as setting time, turning off the system etc.

```
(harish@kali)-[~]
$ systemctl
UNIT                                LOAD    ACTIVE SUB    >
proc-sys-fs-binfmt-misc.automount  loaded active running >
sys-devices-pci0000:00-0000:00:01.1-ata2-host2-target2:0:0-2:0:0:0-block-sr0.device loaded active plugged >
sys-devices-pci0000:00-0000:00:03.0-net-eth0.device loaded active plugged >
sys-devices-pci0000:00-0000:00:05.0-sound-card0-controlC0.device loaded active plugged >
sys-devices-pci0000:00-0000:00:08.0-net-eth1.device loaded active plugged >
sys-devices-pci0000:00-0000:00:0d.0-ata3-host1-target1:0:0-1:0:0:0-block-sda-sda1.device loaded active plugged >
sys-devices-pci0000:00-0000:00:0d.0-ata3-host1-target1:0:0-1:0:0:0-block-sda.device loaded active plugged >
sys-devices-platform-serial8250-tty-ttyS0.device loaded active plugged >
sys-devices-platform-serial8250-tty-ttyS1.device loaded active plugged >
sys-devices-platform-serial8250-tty-ttyS2.device loaded active plugged >
sys-devices-platform-serial8250-tty-ttyS3.device loaded active plugged >
sys-devices-virtual-misc-rfkill.device loaded active plugged >
sys-module-configfs.device loaded active plugged >
sys-module-fuse.device loaded active plugged >
sys-subsystem-net-devices-eth0.device loaded active plugged >
sys-subsystem-net-devices-eth1.device loaded active plugged >
```

5. Crontab - this is used to schedule the recurring tasks.

6. useradd - this command is used to add a new user to the system.

```
(harish@kali)-[~]
$ sudo useradd test

(harish@kali)-[~]
$ compgen -u
root
daemon
bin
kali
harish
test
```

We can see the added user in the list above using the following command.

7. passwd - used to change the password of a particular user.

```
(harish@kali)-[~]
$ sudo passwd test
New password:
Retype new password:
passwd: password updated successfully

(harish@kali)-[~]
$
```

With this command the password for the newly created user have been set.