Users and Groups

- Most beautiful feature of Linux OS is it allows multiple users accessing the system at the same time.
- Security to protect each user from other permissions

Types of user

TYPE	EXAMPLE	USER ID (ID)	GROUP ID (GID)	HOME DIR	SHELL
ROOT	root	0	0	/root	/bin/bash
REGULAR	imran, vagrant	1000 to 60000	1000 to 60000	/home/username	/bin/bash
SERVICE	ftp, ssh, apache	1 to 999	1 to 999	/var/ftp etc	/sbin/nologin

3 types of users in linux

1. Superuser or root user

- Super user or the root user is the most powerful user. administrator user

2. System user

- System users are created by the software or application, e.g when we install apache, it creates user apache.

3. Normal user

- Normal users are the users created by root user, only root has permission to create or remove the user.

Whenever a user is created in linux.

- A home directory is created(/home/username)
- A mail box is created(/var/spool/mail)

- unique UID & GID are given to user

Passwd file

1. /etc/passwd

```
root@ubuntu-focal:/home/vagrant# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

- root = name
- x = link to password file i.e /etc/shadow
- 0 or 1 = UID (user id)
- 0 or 1 = GID (group id)
- root or bin = comment (brief informaiton about user)
- /root or /bin = home directory of the user
- /bin/bash or /sbin/nologin = shell

Group file

2. /etc/group

The file /etc/group stores group information. Each line stores one group entry.

Group name, group password, GID, group members

```
root@ubuntu-focal:/home/vagrant# cat /etc/group
root:x:0:
daemon:x:1:
```

3. /etc/shadow file

This file stores users' password and password related information. Just like /etc/passwd file, this file also uses an individual line for each entry.

1. Username

2. Encrypted password

3. Number of days when password was last changed

4. Number of days before password can be changed

5. Number of days after password must be changed

6. Number of days before password expiry date to display the warning message

7. Number of days to disable the account after the password expiry

8. Number of days since the account is disabled

9. Reserved field

root@ubuntu-focal:/home/vagrant# cat /etc/shadow

root:*:19229:0:99999:7:::

daemon:*:19229:0:99999:7:::

bin:*:19229:0:99999:7:::

sys:*:19229:0:99999:7:::

sync:*:19229:0:99999:7:::

Add user

adduser {username}

→ For ubuntu

Or

Useradd with non-default parameters.

• Syntax:[useradd <option> <username>]

Options : -c //comment

-d //home directory

-m //create home directory

-e //account expiry date

-g //primary group

-G //secondary group -s //shell

-u //user id

E.g

#useradd -m -c “Test User” -d /home/testuser -s /bin/sh testuser

#passwd testuser

To check account expire information

Sudo chase -l {username}

To check details

Id {username}

Groupadd Groupmod Groupdel

Syntax:[groupadd <option> <group_name>]

Option: -g

#groupadd -g 4000 student

#groupmod -n <new_name> <group_name>

#groupmod -n myfriends student

#groupdel myfriends

Usermod

Syntax:[usermod <option> <username>]

Options: -c

-d

-e

-L //lock user -U //unlock user

#usermod -c “Test User2” -d /home/testuser2 -s /bin/bash -g student testuser2

Group change can be done by editing /etc/group file as well.

Userdel

Syntax:[userdel <option> <username>]

Options: -r

#userdel testuser

#userdel -r testuser2

Changing the Default useradd Values

useradd -D

—> Prints the default values

Config file

/etc/default/useradd

Last

→ Gives user who logged into the system

Who

→ Currently logged in user

Lsof

→ Gives all the open files by the user

Permissions

Read: Allows to view the contents of a file and allows to list contents of a directory.

Write: Allows to modify contents of a file and allows create and remove, editing contents of a directory.

Execute: Allows to run a file or execute a program/ script and allows you to change directory and make it your working directory, allows long listing.

-: No permission

Octal Value	Read	Write	Execute
7	r	w	x
6	r	w	-
5	r	-	x
4	r	-	-
3	-	w	x
2	-	w	-
1	-	-	x
0	-	-	-

Managing file ownership

Changing file/ directory ownership with chown

• chown //change ownership

• Syntax chown [Username] [File/Directories]

• # chown srtimsina testfile1 //only superuser can change the ownership

• # chown -R srtimsina /home/srtimsina/testdir1

Managing file group ownership

Changing file/ directory group ownership with chgrp

• chgrp //change group ownership

• Syntax [Group name] [File/Directories]

• # chgrp testgroup1 testfile1

//only superuser can change the ownership

• # chgrp -R testgroup1 testdir1

Managing file and group ownership

Changing file and group ownership simultaneously.

• # chown apache:apache /data/index.php

//chown command can change user-owner and group-owner of a file simultaneously.

• # chown -R apache:apache /data/srtimsina.com.np

• # chown apache.apache /data/srtimina.com.np

// a dot (.) can be used instead of colon (:).

Archiving and Compressing

- Tar
- Gzip
- Zip2
- zip

Creating Archive

tar -cvf [new_file_name].tar [target_directory]
• Extracting Archive
tar -xvf [target_file_name].tar

Compressing

• tar -zcvf testdir1.tar.gz testdir1
• tar -zxvf tarfile.tar.gz
• zip -r [new_filename].zip [target_dir]
• zip -r testdir2.zip testdir2
• unzip testdir2.zip

sudo

→ sudo gives the power to a normal user to execute commands which are owned by the root user.

sudo -i
→ changes from normal to root user.

visudo
username ALL=(ALL) ALL

NOPASSWD
Username ALL=(ALL) NOPASSWD: ALL

Safer way is adding entry in dir
/etc/sudoers.d/

For groups
%groupName ALL=(ALL) NOPASSWD:ALL

Services

apt-get install apache2

Command Syntax [service] [service_name] [command]
///etc/init.d
or
[systemctl] [command] [service_name]
///lib/systemd
#service apache2 status
#systemctl status apache2
Commands can be, start, restart, reload, status, stop

systemctl is-active apache2
systemctl is-enabled apache2
systemctl enable apache2

Configuration file
systemctl works based on the configuration file.
cat /etc/systemd/system/multi-user.target.wants/apache2.service

Processes

top
→ shows all the dynamic processes based on their memory & cpu consumption. It is similar to the task manager in windows.

ps aux
→ displays processes on the screen and quits.
→ process in [ ] are called kernel threads

ps -ef
→ displays not the utilization but also the parent process id
ps -ef | grep apache2

kill (pid)
Kill -9 (pid)
→ kill forcefully

→ ps -ef | grep apache2 | grep -v 'grep' | awk "{print $2}" | xargs kill -9

Zombie:
Processes whose operations are done but their entry is still in the process table.

Orphan:
A child process that remains running even after it;s parent process is terminated or completed without waiting for the child process execution.