Unit 10: User authentication Session 4

Work on user accounts:

- A user is an entity, in a Linux operating system, that can manipulate files and perform several other operations. Each user is assigned an ID that is unique for each user in the operating system.
- After installation of the Linux operating system, the ID 0 is assigned to the root user and the IDs 1 to 999 (both inclusive) are assigned to the system users and hence the ID for local user begins from 1000 onwards.
- Now we will discuss the important commands to manage users in Linux.

useradd:

- useradd is a command in Linux that is used to add user accounts to your system.
- Only root or users with sudo privileges can use the useradd command to create new user accounts.
- When we run the 'useradd' command in the Linux terminal, it performs the following major things:
  - ✓ It edits /etc/passwd, /etc/shadow, /etc/group and /etc/gshadow files for the newly created user accounts.
  - ✓ Creates and populates a home directory for the new user.
  - ✓ Sets permissions and ownerships to the home directory.

Syntax:

The syntax of the useradd command is:

```
# useradd [options] [user name]
```

Important options used are:

- ✓ -m : Adds a home directory (this is a default on some distributions)
- ✓ -s : Sets the user's preferred shell if it's different from /bin/bash
- ✓ -u : Specifies a particular user ID (UID)
- ✓ -g: We can create users with specific group IDs as well with the -g option.

Example:

```
[root@tecmint ~]# useradd -u 1002 navin
[root@tecmint ~]# cat /etc/passwd | grep navin
navin:x:1002:1002::/home/navin:/bin/bash
```

passwd:

- The passwd command changes passwords for user accounts.
- A normal user may only change the password for their own account, while the superuser may change the password for any account.
- passwd also changes the account or associated password validity period.

Syntax:

The syntax of the useradd command is:

```
# passwd [options] [username]
```

Important options used along with passwd:

- ✓ **-d, –delete:** This option deletes the user password and makes the account password-less.
- ✓ **-e, –expire:** This option immediately expires the account password and forces the user to change password on their next login.

Example:

```
[root@tecmint ~]# passwd tecmint
Changing password for user tecmint.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

userdel:

- ➢ userdel command in Linux system is used to delete a user account and related files.
- ➢ This command basically modifies the system account files, deleting all the entries which refer to the username LOGIN.

Syntax:

```
# userdel [options] username
```

Important Options used along with userdel:

- ✓ **-f:** This option forces the removal of the specified user account. It doesn't matter that the user is still logged in. It also forces the *userdel* to remove the user's home directory and mail spool, even if another user is using the same home directory or even if the mail spool is not owned by the specified user.
- ✓ **-r:** Files in the user's home directory will be removed with the home directory itself and the user's mail spool. Files located in other file systems have to be searched for and deleted manually.

Example:

```
gptcs@gptcs-VirtualBox:~$ cat /etc/passwd | tail -1
test_user:x:1001:1001::/home/test_user:/bin/sh
gptcs@gptcs-VirtualBox:~$ sudo userdel -f test_user
gptcs@gptcs-VirtualBox:~$ cat /etc/passwd | tail -1
openldap:x:128:136:OpenLDAP Server:/var/lib/ldap:/bin/false
```

usermod:

- ➢ usermod command is used to change the properties of a user in Linux through the command line.
- ➢ After creating a user if we need to change their attributes like password or login directory etc. In that case we use the usermod command.

Syntax:

```
# usermod [options] username
```

- ➢ Only root or users with sudo access can invoke usermod and modify a user account. On success, the command does not display any output.

Important Options used along with usermod:

- ✓ To change the home directory of a user

  usermod -d /home/newfolder existinguser

- ✓ To change the group of a user

  usermod -g newgroupname existinguser

- ✓ To change user login name

  usermod -l usernewname useroldname

- ✓ To lock a user

  usermod -L existinguser // -U for unlocking user

- ✓ To set an unencrypted password for the user

  usermod -p newpassword existinguser

Example:

```
gptcs@gptcs-VirtualBox$ cat /etc/passwd | tail -1
test_user:x:1001:1001::/home/test_user:/bin/sh
gptcs@gptcs-VirtualBox$ sudo usermod -l shivu test_user
gptcs@gptcs-VirtualBox$ cat /etc/passwd | tail -1
shivu:x:1001:1001::/home/test_user:/bin/sh
```

Look at the above snapshot, user name **test_user** is replaced by the new user name **shivu.**

Work on groups:

- Users can be listed in groups. Groups allow you to set permissions on the group level instead of having to set permissions for every individual user.
- Each group in a Linux system is uniquely identified by a group identification number or GID. All the information listing groups in a system are stored in /etc/group file. The hashed passwords for groups are stored in /etc/gshadow file.
- Every user has a primary user group and zero or more supplementary groups. On login, the group membership is set to the primary group of user.
- In Linux, there can be many users of a single system, (normal user can take uid from 1000 to 60000, and one root user (uid 0) and 999 system users (uid 1 to 999)).

groupadd:

- groupadd command is used to create a new user group.
- groupadd command creates a new group account using the values specified on the command line and the default values from the system.

Syntax:

```
# groupadd [options] group_name
```

Important options used along with groupadd command:

-f, **--force**

This option causes the command to simply exit with success status if the specified group already exists.

-g, **--gid**

The numerical value of the group's ID. This value must be unique

-r, **--system**

Create a system group.

Example:

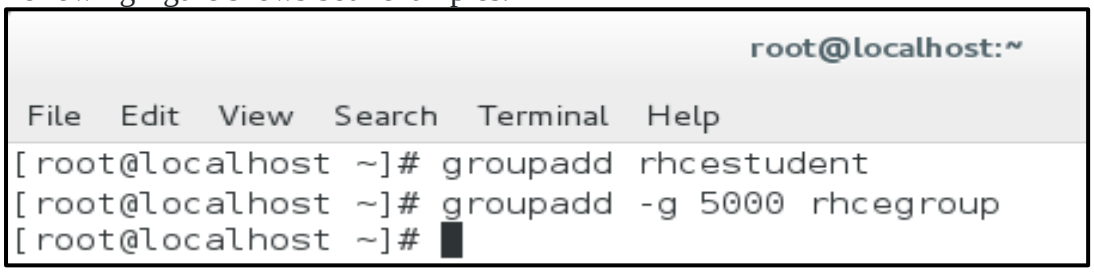
To create a new group named **rhcestudent**, use following command: -

```
#groupadd rhcestudent
```

- ✓ By default Linux picks the next available GID in sequence. For example, if currently used GID is 1000 then new group will get GID 1001.
- ✓ To use a custom GID, we have to specify it with **–g** option. For example, following command will create a new group named **rhcegroup** with GID **5000**.

```
#groupadd –g 5000 rhcegroup
```

Following figure shows both examples.



groupmod:

- groupmod command in Linux is used to modify or change the existing group on Linux system.
- It can be handled by superuser or root user.
- Basically, it modifies a group definition on the system by modifying the right entry in the database of the group.

Syntax:

```
# groupmod [option] group_name
```

Important options used along with groupmod command:

- ✓ **g, –gid GID:** The group ID of the given GROUP will be changed to GID.
- ✓ **-n, –new-name NEW_GROUP:** The name of group will change into newname.
- ✓ **-h, –help:** This option display help massage and exist.
- ✓ **-o, –non-unique:** This option used with the -g option that allow to change the group GID to a non-unique value.
- ✓ **-p, –password PASSWORD:** This gives the encrypted password.
- ✓ **-R, –root CHROOT_DIR:** Apply changes in the CHROOT_DIR directory and use the configuration files from the CHROOT_DIR directory.

Example:

Following command changes the group name **rhcegroup** to group name **rhcelab**.

```
#groupmod –n rhcelab rhcegroup
```

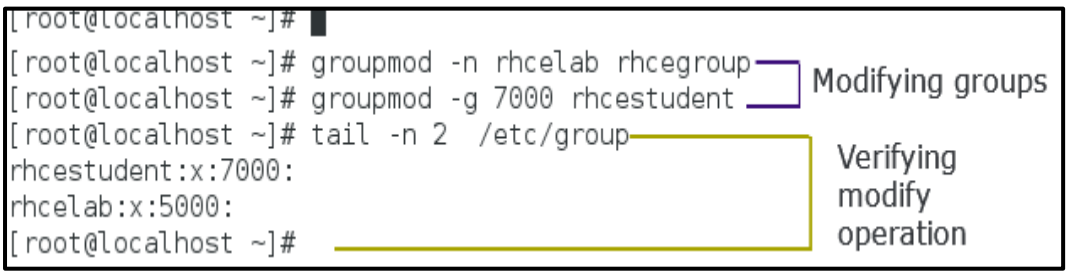
- ✓ Just like group name, we can also change GID of group. To change it, use following command: -

```
#groupmod –g [New Group GID] [Group Name]
```

- ✓ For example, following command changes GID of the group rhcegroup to 7000.

```
#groupmod –g 7000 rhcegroup
```

Following figure shows above commands with output.



gpasswd:

- *gpasswd* command is used to administer the */etc/group* and */etc/gshadow*.
- *gpasswd* command **assigns** a user to a group with some security criteria.
- *gpasswd* command is called by a group administrator with a group name only which prompts for the new password of the group.
- System administrators can use the *-A* option to define group administrator(s) and *-M* option to define members.

Syntax:

```
# gpasswd [option] group
```

Important options used along with gpasswd command:

- ✓ **-a, –add :** This option is used to add a user to the named group.
- ✓ **-d, –delete :** It is used to remove a user from the named group.
- ✓ **-r, –remove-password :** It is used to remove the password from the named group.
- ✓ **-R, –restrict :** This option will restrict the access to the named group.
- ✓ **-A, –administrators :** Set the list of administrative users.
- ✓ **-M, –members :** It set the list of group members.
- ✓ **-h, –help :** It displays the help message and exit.

Examples:

```
# gpasswd -a mozart musicians
```

Adds the user **mozart** to the group **musicians**.

```
# gpasswd -A george mathematicians
```

Give user **george** administrative rights to the group **mathematicians**.

```
# gpasswd -d hope yankees
```

Remove user **hope** from the group **yankees**.

groupdel:

- *groupdel* command is used to delete a existing group.
- It will delete all entry that refers to the group, modifies the system account files, and it is handled by superuser or root user.

Syntax:

```
groupdel –f existinggroup
```

Option used:

-f –force: It used to delete a group even if it is the primary group of a user

Example:

Following command delete group **rhcegroup**.

```
#groupdel rhcegroup
```

Following figure shows above operation.