User and Group Administrations

USER ADMINISTRATION

In Linux/Unix:

- One or more than one users in Linux at a time.
- Users on a system are identified by a username and a userid.
- The username is something that users would normally refer to, but as far as the operating system is concerned this is referred to using the user id (or uid).
- He user id numbers should be unique (one number per user).
- If you had two usernames with the same user id, effectively there permissions would be the same and the files that they create would appear to have been created by the same user.
- This should not be allowed and the useradd command will not allow usernames to share the same userid.

Some Important Points related to Users

- Users and groups are used to control access to **files and resources**
- Users login to the system by supplying their username and password
- **Every file on the system is owned by a user and associated with a group**
- Every process has an owner and group affiliation, and can only access the resources its owner or group can access.
- Every user of the system is assigned a unique user ID number (the UID)
- Users name and UID are stored in **/etc/passwd**
- User's password is stored in **/etc/shadow** in encrypted form.
- Users are assigned a **home directory** and a program that is run when they login (**Usually a shell**)
- **Users cannot read, write or execute each other's files without permission**

Types of users:

TYPE	EXAMPLE	USER ID (UID)	GROUP ID (GID)	HOME DIRECTORY	SHELL
Super User	root	0	0	/root	/bin/bash
System User	jenkins,mail, apache, nobody	1 to 499	1 to 499	/var/ftp , etc	/sbin/nologin
Normal User	maha,raj, etc	500 to 60000	500 to 60000	/home/user name	/bin/bash

1) **Super user or root user**

Super user or the root user is the **most powerful user**. He can do anything as admin user.

2. System user

System users are the users **created by the softwares or applications**. For example if we install Apache it will create a user apache. These kinds of users are known as system users.

3)Normal user

Normal users are the users **created by root user**. They are normal users like maha, raj etc. Only the root user has the permission to create or remove a user. We can make normal user as sudo user by root,

Whenever a user is created in Linux things created by default:-

- A home directory is created(/home/username)
- A mail box is created(/var/spool/mail)
- unique UID & GID are given to user

Linux uses UPG (User Private Group) scheme

- It means that whenever a user is created is has its own private group
- For Example if a user is created with the name **maha**, then a primary group for that user will be **maha** only

Important files for user

1. /etc/passwd : in this file maintain username informations

2. /etc/shadow : in this file maintain password informations

vi /etc/passwd

```
tcpdump:x:72:72::/:/sbin/nologin

ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash

Maha:x:1001:1001:myuser:/home/maha:/bin/bash
```

The above fields are

maha =name
x= link to password file i.e. /etc/shadow
1001= UID (user id)
1001=GID (group id)
myuser= comment (brief information about the user)
/home/maha = home directory of the user
/bin/bash = shell

vi /etc/shadow

```
maha:$6$iPmA:18166:0:99999:7:::
```

The fields are as follows,

1. maha = User name
2. **6iPmA** = Encrypted password
3. **18166** = Days since that password was last changed.
4. 0 = Days after which password must be changed.
5. 99999 = Days before password is to expire that user is warned.
6. 7 = Days after the password is expires that the user is disabled.
7. A reserved field.

Creating a user

 # useradd <option> <username>

options are

- -u user id
- -G Secondary group id
- -g primary group id
- -d home directory
- -c comment
- -s shell

Let's create a user with default attributes:

adduser maha

Let's create a user with our own attributes:

#adduser -d /home/maha -c myuser -s /bin/bash -u 505 raj

Assigning password to the user:

#passwd

Modifying the user's attribute

usermod <options> <username

- -l to change login name
- -L to LOCK account
- -U to UNLOCK account

ex. # usermod -l newname oldname (changing the name of the user)

ex. # usermod -L username to lock the user account

ex. # usermod -U username to unlock the user account

Note: - when an account is locked it will show ! **(Exclamation mark)** in /etc/shadow file

The password parameters

For any user we can set the parameters for the password, like min and max password age, password expiration warnings and a/c expiration date etc

#chage -l < username>

#chage -l maha

- **Last password change**: When the password was change last time.
- **Password expires**: Password expiry date
- **Password inactive**: After password expiry grace period before the account gets locked.
- **Account expires**: Date on which the account expires.
- **Minimum number of days b/w password change**: once the password is changed, it cannot be changed until a min period of specified date. [0] means never.
- **Max number of days b/w password change**: After changing the password how long it will be valid for.
- **Number of days of warning before password expires**: start of warnings to change the password, no. of days before the password expires

Changing the password parameters

Changing of the password parameters can be done by two ways.

#chage <user name >

#chage <option> <value> <username>

The options which can be used are as follows

- -m for Min password age
- -M for Max password age
- -d for last time the password is changed.
- -W Password expiration warnings
- -I Password inactive [-1 means inactive].
- -E A/C expiration date

Deleting a User:

To delete a user the syntax used is:

#userdel <username> : without home dir

#userdel –r < user name > : with home dir

GROUP ADMINISTRATION

GROUPS

- Users are assigned to groups with unique group ID numbers (the GID)
- The group name and GID are stored in **/etc/group**
- Each user is given their **own private group**
- They can also be added to their groups to gain additional access
- All users in a group can share files that belong to the group

Important files for groups

1. **/etc/group** : in this file maintain group's informations, like group name,GID

2. **/etc/gshadow** : in this file maintain group's password related informations

- A user's primary group is defined in the /etc/passwd file
- A user's Secondary groups are defined in the /etc/group file.
- The primary group is important because files created by this user will inherit that group affiliation(other groups).

- Each user is a member of at least one group, called a primary group.
- In addition, a user can be a member of an unlimited number of secondary groups.
- Group membership can be used to control the files that a user can read and edit.

For example,

If two users are working on the same project you might put them in the same group so they can edit a particular file that other users cannot access

Creating a Group with default options

To create a group the syntax is

#groupadd <name for the group>

#groupadd mahagrp

Creating a group with user specified group id (GID)

#groupadd <option> <name for the group>

#groupadd -g 595 mahagroup

Adding single or multiple users in to the group with various attributes

#gpasswd < option> <arguments> <group name>

Options:

- -M For Adding Multiple users to a group
- -A for Adding a group Administrator
- -a for Adding a single user to a group
- -d removing a user from a group

Ex: #gpasswd –M <user>,<user>,<user> <group>

Ex: #gpasswd –M sai,raj mahagrp

Adding a single user using gpasswd

Ex: #gpasswd –a jai mahagrp

Making a user as a administrator

#gpasswd –A jai mahagrp

Removing a user from the group

#gpasswd –d sai mahagrp

`

Modifying the properties of the group

#groupmod <option> <arguments> <group name>

The options are

- -g to change the group id
- -o to override the previous assigned id, if it matches with the new one.
- -n to change the group name

Ex: #groupmod –g 600 mahagroup

#groupmod –n <new name > < existing name >

Ex: #groupmod -n mahanewgrp mahagroup