Linux User and Group Management

Linux supports multiple users logging in at the same time

We can add users to group and give that groups specific access to resource

Every user has a name and a UID and a group has a name and GID

Each user requires to belong to one group and is called primary group, this group is generally the name of the user and is created automatically in red hat 7

Files created by the user belongs to the the user and their primary group and user cannot belong to one or more primary group

However, we can add users to additional groups known as supplemental groups. Groups cannot contain groups that essentially means nesting of groups is not allowed

Users must have a password to login and the name of users and groups are case sensitive

The file that stores the user account details is /etc/passwd

This file has got 7 columns

Column 1:- Has the name of the user

Column 2 :- encoded password or it would show x if we are using shadow suite which generally encrypts the password and the encrypted is password gets stored in /etc/shadow file

Column 3 :-User id # In older times the system accounts was from 0-499 and 500 being user accounts. Wherein system accounts are being referred over here are system processes. With new OS it is 0-999 for system accounts and 1000+ for user accounts

This can be modified in the /etc/login.defs

Column 4 :- Group id # is the GID of the primary group and by looking at the GID # in the /etc/group file we can find the group name

Column 5 :- Comment field

Column 6 :- It’s the home directory, by default it is /home/$username

This can be configured in /etc/defaults/useradd

Or by using useradd -D

Column 7 :- It’s the default login shell by default is /bin/bash we can change the default by going in to

/etc/default/useradd or useradd -D

In order to find the available shells we can find that out /etc/shells

With the shadow suite file all the account aging information and password hashes are stored in /etc/shadow file. This is only readable by root

/etc/shadow file has 9 columns

Column 1 :- It is the username and must match the /etc/passwd file

Column 2 :- It would indicate the password or it would show that the password has not been set, in RHEL it would either show that there is an exclamation mark stating that either the password is not set or the account is locked.

Chacracters between the dollar symbol indicates the hash type 1 Md5 5 SHA256 and 6 SHA 512

Hash type being used can be changed in the /etc/login.defs file.

Note :- In case you change the hash type you have to force fully change the password in order for the old users to take the advantage of the new hashing algorithm

Column 3 :- # of days from 1970 january when the password was changed

Column 4 :- Number of days before the password can be changed (0 indicates that it can be changed anytime)

Column 5 :- # of days the password must be changed (99999 means 274 years)

Column 6 :- # of days to warn the user before the password expires

Column 7 :- # of days after the password expires and the account is disabled

Column 8 :- # of days since 1970 January and the account is disabled

Column 9 :- future use

We can change the system wide information by /etc/login.defs file or either changing the individual user information using “chage” command

Groups information is stored in /etc/groups

It has only 4 columns

First Column :- Group name

Second column :- Password if you see x you will find the password in the /etc/gshadow file

Column 3 :- It is the group id and can be changed in the /etc/login.defs file, any group id below 100 belong to system groups

Column 4 :- More than one users they would have , between them

Note :- If the password is set for a group and we are using a shadow suite it would be stored in /etc/gshadow

The purpose of setting up the password on a group is to gain privilege of that group by using a “newgrp” command

/etc/gshadow file can only by read by the root account

It also has 4 columns

Column 1 :- Username which should match with the /etc/group file

Column 2 :- It stores the password of the group in case there is an ! mark then the user would not be able to change to this group using the newgrp command

Column 3 :- Holds the information about the group admin who can change the password and the members of the group in a comma separated format

Column 4 :- Comma separated list of users who can change to the group without a password

Note :- In case if a user is changing to the new group using a newgrp command they are essentially changing their primary group and in case once they change their primary group and they create any file that file would be owned by the user and the new primary group

In order to prevent dictionary attacks a random salt is used in front of the password and you can see this salt before that $ symbol

As we know that /etc/passwd and /etc/group files are world readable by everyone, and password stored in /etc/shadow and /etc/gshadow is only readable by root

In most of the new OS you will find shadow suite installed by default and once the shadow suite is installed you can unconv the password from “pwunconv” command and group password from “grpunconv”

Once we do “pwunconv ” command you can see the password to be transferred from /etc/gshadow file to the /etc/passwd field

Adding a new group using “groupadd”

Changing the group password is “gpasswd”

Note :- settings for hash is stored in /etc/login.defs file

In order to update the hash being used we can either manually edit the file or change it by using the authconfig command

# authconfig –passalgo=sha512 --update

Note :- In case you update the HASH this will not be updated on old user account hence in order to make sure that the HASH change applies to older account as well we need to make sure that the user change password at the next login

In order to force a user to change the password at the next login we have to make sure that their password has expired and user must change it

# chage -d 0 $username

We can check for password using “pwscore” command by piping the output to pwscore

Password policies :-

In Linux password policies work on credit system

Example

In case you have a minimum password

When we create a user without specifying any settings there are certain defaults that are already assumes

We can find these settings in the /etc/login.defs file

In case if we change anything in this file this would only affect the new users and the existing users on the system we have to do that manually

Password againg information in /etc/login.defs file is overwritten by /etc/security/pwquality.conf file

In case you don’t want the user to have a home directory while adding the user we can use #usreadd -m command while creating the user

Note :- global user account settings is stored in /etc/login.defs file

In order to make password aging change for a user we can either edit the /etc/shadow file but this is not recommended and we use chage command in order to do that

# chage -d changed the # of days since 1970 when the password was last changed

# chage -d 0 $username :- will make sure that the user changes the password at the next login

or passwd –expire $username

# chage -E sets the account expiry date

# chage -I :- sets the number of days for inactivity after password expiration and the account is locked

# chage -m :- minimum # of days before a password can be changed by the user (0) means that the user can change the password at any time

# chage -M :- Maximum # of days for which the password is valid

# chage -W :- # number of days for warning before the password needs to be changed

# chage -l :- lists the password aging information about a user

Note :- When the password expires user can still login using SSH keys and in case the account expires user cannot login

Note :- /etc/default/useradd :- also defines the skeleton directory what this does it copies the files contains in the skeleton directory to the home directory of the user’s automatically

Note :- When you have to two exclamation marks that indicates that the password is not set

In case if we do “ls /home” we will get to know the users that have a home directory also I would like to state in case we do “ls -la /home/bob” we will get to know about the files that were copied over from the skeleton directory as part of the /etc/defaults/useradd file

Skeleton directory is at /etc/skel location

In case if we don’t want the delete the home directory of a user we would use “userdel $username”

In case if we want to delete the home directory as well we would use “userdel -r $username”

Modify Users

In order to change the user account settings once they have created we would use “usremod” command

#usermod -d :- changed the user home directory. We might to do this in case we moved their home directory and we want it to reflect in the /etc/passwd field

#usermod -g/-u :- changes the primary group id and user id

#usermod -G :- Adding users to supplemental group. This is a complete list and would override their current list of groups unless we use it in conjunction with -a

#usermod -l :- allows the user to change the login name

#usermod -L/-U :- Locks and unlock the account

#usermod -m :- moves the user home directory

# usremod -s :- define the login shell for the user account

Note :- the configuration file that we have in /etc/security/pwquality.conf defines the password policies which is applied on the system.

Note :- /sbin/nologin :- is the shell that does not allows any login on the system

Common passwd options

# passwd -l/u :- Locks and unlocks the password

# passwd -e :- expires the password that means that the user has to change the password at the next login

# passwd -d :- deletes the user password, it is exactly like a new user that is created without any password

# passwd -S :- outputs the password status

Note :- groups can not be nested

In case you want to add a user to the group

# gpasswd -a $username $groupname

Alternatively we can also use

#usermod -aG $groupaname #username

We can switch users in case there is group that has some specific permissions and we want to use that we can use the “newgrp” command to switch to that group

What this essentially does is that it changed the primary group for that user, so in case if you have created any file while changing the group the new file would be owned by the new primary group

In order to get information about the user groups type “id” and that would list the primary groups and the supplementary groups

In order to revert back to the previous group just type “newgrp” without any arguments.

In case you don’t want to enter the password you can add the users to that group

#usermod -aG $groupname

#gpasswd -a $username $groupname

In order to add a user as an admin for a group

#gpasswd -A $username $groupname

To verify as we all know that we have the information about the admin for the group in /etc/gshadow file