Managing Linux Users and Groups

-User: A person or service that requires access to system files or resources
-User Account: Method of providing or restricting access to system resources

UID and Privileges

-UID: User ID. Unique ID used to organize users

User ID's **0-99** are reserved for Administrative users and are added to the operating system during the installation process. Admin users are created by the OS and can't be created by an application.

ROOT has the **UID** of **0**. This UID is a privileged linux account and the Admin of the system.

The config file **/etc/login.defs** is the file that defines new user defaults, such as the location of the user's mailbox, their user ID, and their PW aging defaults. UID ranges are specified by the variables **UID_MIN & UID_MAX** and the ranges for a User Account (which is non-privileged) are between **1000 & 6000.** 1000 being the default minimum and 6000 being the default maximum.

A **linux service** is an application that is running in the background. These services may be assigned a ***user account*** as well. The User account ID for a service is in the range of **0-99** or set by the variables **SYS_MIN & SYS_MAX** (also in the config file mentioned above)**.** Service or system accounts are non privileged accounts created by a sys admin executing the command → "useradd –system <system_account_name>"

Where Linux User Account info is stored

/etc/passwd - Contains user account information
/etc/shadow - Contains user password and password aging information
/etc/group - Contains a list of groups and their members

User database file /etc/passwd

The file /etc/passwd is an example of a flat-file database. Each line of the file contains a unique user record which contains 7 fields with a ":" being used as the delimiter. The format follows:
username:password:UID:GID:comment:home_directory:default_shell

Protected user password file /etc/shadow

Linux developers built the shadow utilities and moved the passwords from /etc/passwd to /etc/shadow so that it could only be visible by root instead of being unencrypted and viewable by anyone in /etc/passwd. Encrypted password reside inside of /etc/shadow. Each record in /etc/passwd should have a corresponding record in /etc/shadow. The format is as follows:
username:password:last_modified:min_days:max_days:warn_days:inactive:Expire

- Last_modified: the field displays the number of days since jan 1 1970 that the password was last changed
- Min_days: this field displays the minimum number of days required before a password can be changed. The default value is specified in /etc/login.defs

- Warn_days: This field displays the number of days prior to password expiration the user will be warned of the pending expiration. The default value is specified in /etc/login.defs
- Inactive: Displays the number of days after password expiration the user account will be disabled. This is to prevent open accounts that aren't being used.
- Expire: Displays the number of days since 1/1/1970 after which the account will be disabled.

Group database file /etc/group

Groups allow you to give a group of users the necessary permissions and access to a certain resource or resources

Group info is stored in /etc/group and the /etc/gshadow config files

The /etc/group flat-file database contains four fields:
group:password:GID:userlist

- Group: specifies name of the group
- Password: specifies the group password
- Gid: specifies the group ID
- Userlist: list members who are secondary members of the group

Commands for displaying UIDs & GIDs

Id - displays the current UID and GID

Id <username> - does the same but for the specified user

W - displays the UID of all logged-on users, what processes they are executing, and what devices the processes are executing from.

Who - similar to the w command but does not display the processes that are executing.

Su <username> - changes the current user's UID, primary group ID, and home directory.

Su - <username> - does the same, but also reads the new user's profile (that is, reads all of the users config files). In essence the "su dash" allows you to become exactly like the new user.
(when any type of su command is executed your UID & GID changes to that of the user switched to. Non root users need to know the passwd of the user being switched to.)

Provisioning new users with useradd

Useradd adds new users (self explanatory). It obtains defaults initially from /etc/login.efs and next from /etc/default/useradd. Entries from /etc/skel are used to populate the new user's home directory.

Options for useradd:

- c: comment field
- e: specifies the date when the account will be disabled
- g: specifies the user's primary group
- G: specifies user's secondary groups. May enter a comma-deliminated list of group names or group id's
- d: defines the location of the home directory
- m: creates(makes) the home directory (not necessary if the variable CREATE_HOME in /etc/login.defs is yes)
- r: specifies that the user being created is a system user. The system user ID will be in the range specified by SYS_UID and SYS_UID_MAX in /etc/login.defs

- s: specifies the absolute path to the default shell for the user
- u: allows an admin to manually specify UID

Setting defaults in /etc/default/useradd

/etc/default/useradd contains default variables used by the command useradd unless overridden by settings in /etc/login.defs. To view the value set in /etc/default/useradd, execute:
Useradd -D

/etc/login.defs also specifies where the user's mail will be stored. The value of the variable CREATE_MAIL_SPOOL in /etc/default/useradd determines whether the file will be created when the user is added or not.

Building user consistency with the /etc/skel directory

/etc/skel is the default directory that contains files and directories copied to a new users home directory. The files in this directory can be modified if you want all new users to have specific files or settings

You can also create a skeleton directory for users with similar needs.

For example: If you have a specific group that needs specific settings in /home/<username>/.bash_profile and you have certain scripts available. You can create a directory called /etc/skel_<group_name> and copy the files from /etc/skel into this new directory you just created. Then you edit the appropriate config files and add any additional files into the new directory.
*When adding a new user for that group use the option "-k <skel_directory>"
Eg: "useradd -k /etc/skel_<group_name> <user_name>"

Using passwd to set a password

The **passwd** utility allows you or root to change your password and allows a system admin to manage password aging.

Root user can change a user's password by executing the command "passwd <username>" passwd options include:

- -l: locks the user account but does not remove the current password. The encrypted password of an account locked using passwd -l will have two exclamation points (!!) preceding the password within /etc/shadow.
- -u: unlocks the user account
- -n: sets the minimum number of days (MIN_DAYS) required before a password can be changed
- -x: sets the maximum number of days (MAX_DAYS) before a password must be changed
- -w: Sets the number of days prior to password expiration (WARN_DAYS) when the user will be warned of the pending expiration
- -i: sets the number of inactive days to wait after password has expired before disabling the account
- -s: Displays password aging information. Password aging information may also be displayed by executing the chage -l <username> command, as discussed in the next section.

If a user account has an asterisk (*) in the password field of /etc/shadow, the account is disabled. User accounts in the ID range of 0-99 don't require a password and will display an asterisk in this field.

You can force a user to change their password at their next logon by expiring their account using the command "passwd -e <username>" or

setting the last change date to 0 using the command “chage -d 0 <username>”

Changing password aging with the chage command

The chage command allows you to view or change the user’s password aging information.
“Chage -l” lists current users aging info
“Chage -l <username>” lists current aging info for a specific user
“Chage <username>” opens a text user interface with prompts for changing user aging info

(see other command options)

Modifying User settings with *usermod*

usermod is used to modify an existing user account with options similar to **useradd.**

- -G <Groupname>: Removes all of the users secondary groups and replaces them with a new secondary group or comma delimited list of secondary groups.
- -aG <group_name>: Adds a new secondary group.
- -l: Changes the username (logon name).
- -d: Changes the location of the user’s home directory
- -m: moves (renames) the current user’s home directory to the new user’s name. The following cmd renames user student2 to user2: “usermod -l user2 -d /home/user2 -m student2”

Deprovisioning users using *userdel*

Userdel is used to remove a user account. “userdel <username>” removes a user record from /etc/password & shadow but their home directory and files remain to be reassigned and backed up.
*to delete the home directory and mail as well use “userdel -r <username>”
*none of these commands remove any groups.

Provisioning workgroups with *groupadd*

groupadd is used to add groups like so: groupadd <group_name>

Options:

- -g: specified a GID for the new group. As with users its not necessary to specify a GID, as the system will automatically assign one
- -r: Specifies that the group being created is a system group

Setting group passwords with *gpasswd*

gpasswd is used to manage the files /etc/group and /etc/gshadow. Can be executed by system and group admins. To assign a group admin execute “gpasswd -A <username>”.

gpasswd options:

- -a <username>: adds a user to the group
- -d <username>: deletes a user from the group
- -r: removes the group password

Changing Workgroup Settings with *groupmod*

groupmod command is used to modify group information using the following options:

- -g: changes the groups GID number
- -n: Changes the group name

Deprovisioning the workgroup with *groupdel*

- To delete an existing group use the groupdel command like so: “groupdel <group_name>”. Before deleting a group, make sure that the users’ access to the files and directories is not compromised.