Linux User Managements

- A user's activity is determined by:
  - the user's role
  - the user's group

- When an user is created on the system, the following information is stored in **/etc/passwd**
  - A typical entry looks like this:

```
user1:x:1003:1003:User Name,,,,:/home/dir:/bin/bash
```

`/etc/passwd` File

- **Field #1**: ***login name***
  - user name the user needs to type in to log into the system
- **Field #2**: ***password field***
  - The `x` character indicates that encrypted password is stored in a separate file: `/etc/shadow`
- **Field #3**: ***UID***
  - User ID associated to login name

`/etc/passwd` File

- **Field #4**: ***GID***
  - main Group ID associated to login name
- **Field #5**: ***other info or comment***
  - other information about the user, such as real name, office #, telephone number, etc
- **Field #6**: ***default home directory for user***
  - set by administrator, directory is owned and managed by user
- **Field #7**: ***default shell for user***
  - shell that user will start in when login into the system

`/etc/passwd` File

`/etc/passwd` file

- It generally requires ***root*** access for modifications
- It's content can be viewed by anyone
- Users can modify content related to their own account information using the appropriate commands only (such as **`passwd`** to change password)
- All user passwords will be stored in **`/etc/shadow`**, accessible only by root or root processes

/etc/shadow File

- Each entry in **/etc/shadow** contains the user's login, their encrypted password, and a number of fields relating to password expiration. A typical entry looks like this:

```
user1:/3GJllg1o4152:11009:0:99999:7:::
```

1. Username: up to 8 characters. Case-sensitive, usually all lowercase. A direct match to the username in the /etc/passwd file.

2. Encrypted password: A **!** or ***** is used in this field.

File Permission

File permissions for **/etc/passwd** and **/etc/shadow**

```
-rw-r--r-- 1 root root   2080  Feb 2 10:48 /etc/passwd

-rw-r----- 1 root shadow 1539  Feb 5 15:24 /etc/shadow
```

/etc/shadow File

3. Last password change (last changed): Days since Jan 1, 1970 that password was last changed.

4. Minimum: The minimum number of days required between password changes i.e., the number of days left before the user is allowed to change his/her password.

5. Maximum: The maximum number of days the password is valid (after that user is forced to change his/her password).

6. Warn : The number of days before password is to expire that user is warned that his/her password must be changed.

/etc/shadow File

7. Inactive : The number of days after password expires that account is disabled.

8. Expire : Days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used.

9. Last **x** number of passwords not allowed.

Exercise

This is a good time to review the `cut` utility.

Using root privileges, try

```
cut -d: -f1-3 /etc/shadow
```

Question

If two users have the same password, will the encrypted password be the same?

`useradd` Command

- **`useradd`** ***[options] username***
  - Used to create a new user account. Linux will also create a group with the same name by default.
  - Useful options
    - **`-d`** Define home directory
    - **`-g`** Initial group name, the group name must exist
    - **`-G`** A comma-separated list of supplementary groups which the user should belong to

`useradd` Command

`-c` Any text string: add comments or other information: such as user's full name

`-N` Do not create a group with the same name as username, but add the user to an existing group: users (GID=100)

`-e` Account expiration date: **`YYYY-MM-DD`**

`-s` Login shell

`-m` Create home directory if does not exist, and copy initial files contained in **`/etc/skel`** (skeleton)

`-D` Display and change the default values

```
useradd -D
```

`useradd` Command

- When invoked without the **`-D`** option, the **`useradd`** command creates a new user account using the values specified on the command line and the default values from the system.
- Example:

```
useradd -c "Norman Han" -d /home/hann -m
-g faculty -G computer,staff -e 2017-01-01
-s /bin/bash hann
```

Add a group

Use `groupadd`

to add a groups `computer,staff`

```
groupadd computer
groupadd staff
groupadd faculty
```

Check using `cat /etc/group`

`userdel` Command

- **`userdel`** ***[options] username***
  - Remove an user from the system
  - Options
    - **`-r`**
      - Remove the user's home directory and files contained in it

`userdel`

Example:

- **`userdel user2`**
- Deletes the entries of `user2` from **`/etc/passwd`** and **`/etc/shadow`**, but it does not delete the user's home directory and files in it.

`usermod` Command

`usermod [options] username`

- Allows for modifying most of the information stored in **`/etc/passwd`** associated with an user account
- several options available to modify almost all of the information associated with any account, assuming the user has the right to modify the information
  - **`-c`**
    - add comments or other information
  - **`-d`**
    - change home directory if **`-m`** option is given, the contents of the user's home directory will be moved to the new home directory, which will be created if it doesn't exist

`usermod` Command

`-g` change initial group

`-G` change supplementary groups which user is also a member of, if the user currently is a member of a group which is not listed, the user will be removed from the group

`-s` change the login shell

`-e` The date on which the user account will be disabled: **`YYYY-MM-DD`**

`usermod` Command

`-l` Change login name

`-L` Disable/lock user's password, which will place a **`!`** before encrypted password in **`/etc/shadow`**

`-U` Enable/Unlock user's password, which will remove the "!' from encrypted password in **`/etc/shadow`**

usermod

Example:

```
usermod -c "Sherpa Tenzing, WT130, ext 4529, 613-727-4723" user4
```

Caution: Do not cut and paste this command in bash, double quotes do not translate correctly.

`chsh` Command

`chsh` ***[options] username***

- Changes login shell associated with a user account
- if a shell isn't specified on the command line, it will prompt for the shell to use

- **Options**

`-s`: specify the shell to associate to the account

-example: **`chsh -s /bin/bash user1`**

Linux Groups

- Groups are a simple mechanism for allowing a "*group*" of users with common access requirements to have access capabilities which, individually, they do not have.

  As such, users are generally associated to group(s) based on an identified need to access information they cannot access as a user alone. This is generally done by administrators.

Groups

All group information is maintained in the **`/etc/group`**
and
`/etc/gshadow` on the system.

Format of this file is different than **`/etc/passwd`** file.

Groups

Sample entry in
/etc/group

```
user1:x:500:user2,user3
```

Linux Groups

Where,

1. **group_name**: It is the name of group. If you run `ls -l` command, you will see this name printed in the group field.
2. **Password**: Generally, password is not used, hence it is empty/blank. It can store encrypted password. This is useful to implement privileged groups.
3. **Group ID (GID):** Each user must be assigned a group ID. You can see this number in your /etc/passwd file.
4. **Group List**: It is a list of user names of users who are members of the group. The user names, must be separated by comma

Linux Groups

/etc/gshadow contains lines with the following colon-separated fields:

```
group1:$1$VF61Ap3s$c7J6tC:user1:user1,user2
```

1. group name
2. encrypted password
3. comma-separated list of group administrators
4. comma-separated list of group members

Linux Groups

- Users belong to a group which is called

initial (primary) group

which is generally set-up by the administrators when they created the account.

- The user need not do anything to belong to this group, it is automatically associated with the user when they log into the system.

- The initial group's GID is listed in file **`/etc/passwd`**.

- A user can also belong to several groups (supplementary groups) at the same time and switch between them for access purposes.

Group managements

- **`groupadd`** Create a new group
- **`groupdel`** Remove a group
- **`groupmod`** Modify a group
  - Options
    - `-g` ***gid***
    - `-n` ***groupname***

Group managements

`groups` `username`

- Displays which groups the user currently belongs to

`newgrp` `groupname`

- Change group ID (effective group) to a new group they already belong to in during the login session
- Prompt for group password if it is set to change to a group which the user is not a member of

`gpasswd` Command

- `gpasswd` is used to administer the **`/etc/group`** and `/etc/gshadow` file. Every group can have administrators, members and a password. If a group password is set, non-member can supply a password to use **`newgrp`** command to change initial group.

`-A` Define group administrator, group administrator can add or delete members with `-a` and `-d` options.

`-r` Remove group password, only group members will be allowed to use `newgrp` to join the group.

`id` Command

- `id [options] username`
  - prints UID, GID and groups information for username
  - if no user is specified, then current user is assumed

`su` Command

- `su [-]` ***username***
  - allows for switching from one account to another
  - opens up a subshell as the new user
  - only possible if the user has the password for the other account
  - the dash **`-`** *which must be preceded and followed by a space*, indicates that the user wishes to use a login shell.

    Without the dash **`-`** the **`su`** command will switch personalities but not associated environment settings

`su` Exercise

1. type `su`, then observe the prompt and check directory with `pwd`

2. `exit`

3. type `su -`, again observe the prompt and check directory with `pwd`

Summary

- `useradd` - To add an user account
- `userdel` - To remove an user account
- `passwd` - To set a password
- `usermod` - To modify user account information
- `groups` - To display all groups of an user
- `groupadd` - To create a new group
- `groupdel` - To delete a group
- `newgrp` - To change primary group during the login session
- `gpasswd` - To manage groups: to designate a group administrator, to add group members, to delete group members
- `id` – To display user ID and group ID

`grep` command

- **`grep`** command selects and prints lines from a file (or a bunch of files) or the standard output that match a pattern (a regular expression).
- Options:
  - **`-i`** Ignore uppercase and lowercase when comparing.
  - **`-v`** Print only lines that do not match the pattern.
  - **`-c`** Print only a count of the matching lines.
  - **`-n`** Display the line number before each matching line.

grep

Examples:

- `cat /etc/fstab | grep "ext3"`
- `grep -v "ext3" /etc/fstab`
- `cat /etc/passwd | grep root`
- `grep -c root /etc/group`

Wildcard Characters

*	Matches 0 or more characters in a filename
?	Matches 1 character in a filename
[aed]	Matches 1 character in a filename-provided it is either a,e, or d.
[a-e]	Matches 1 character in a filename-provided it is a,b,c,d, or e
[!a-e]	Matches 1 character in a filename-provided it is not a,b,c,d, or e

Using Brace Expansion:{}

- Using **{ }** to create a list of files or directories
  - `touch ~/dir1/{f1,f2,f3}`
  - `mkdir ~/dir2/{a1,a2,a3}`
  - `mkdir -p ~/backup/{old,new}/{labs{1,2,3},lecture{1,2,3}}`
    - How many directories have been created?
      - Use `tree` command to view
        - » Option `-L`: used to define the MAX display depth of the tree structure
        - » Example: `tree / -L 1`

Define Aliases

Define your own commands

```
alias myls='ls -la'
alias
```

To display all alias that have been defined

Remove Alias

```
unalias myls
```

An alias lasts only for the duration of the login session