IT2654

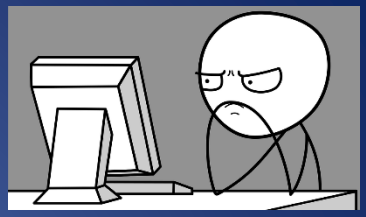
SYSTEMS ADMINISTRATION & SECURITY

Linux Topic 2: User Account & Group Management

Objectives

- Managing user account
- Managing group
- Password policy

What is a User?



- Every process on the system runs as a particular user
- Every file is owned by a particular user
- Access to files and directories are restricted by user
- View user and associated processes – ps aux
- View user associate with file or directory – ls –l
- Linux users defined in databases - /etc/passwd

User Processes

- $ ps aux
- See ps command details – man ps

```
[student@server ~]$ ps ux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
student    6175  0.0  0.1 108332  1792 pts/0    S    00:22   0:00 -bash
student    6248  0.0  0.1 108332  1788 pts/0    S    00:26   0:00 -bash
student    6277  0.0  0.1 110232  1140 pts/0    R+   00:28   0:00 ps ux
[student@server ~]$
```

User Files & Directories

- `$ ls -l`

```
[student@server ~]$ ls -l
total 40
drwxrwxr-x. 2 student student 4096 Jun  6 00:29 apple
drwxr-xr-x. 2 student student 4096 Jun  5 18:00 Desktop
drwxr-xr-x. 2 student student 4096 May 11  2015 Documents
drwxr-xr-x. 2 student student 4096 May 11  2015 Downloads
-rw-rw-r--. 1 student student   12 Jun  7 06:26 file
-rw-rw-r--. 1 student student    0 Jun  6 00:28 file two.txt
drwxr-xr-x. 2 student student 4096 May 11  2015 Music
-rw-rw-r--. 1 student student    0 Jun  6 00:27 one.txt
drwxr-xr-x. 2 student student 4096 May 11  2015 Pictures
drwxr-xr-x. 2 student student 4096 May 11  2015 Public
drwxr-xr-x. 2 student student 4096 May 11  2015 Templates
drwxr-xr-x. 2 student student 4096 May 11  2015 Videos
[student@server ~]$
```

Unix File Attributes

```
$ ls -l
 -rw-rw-r-- 1 jason users 10400 Sep 27 08:52 sales.data
```

Permissions	-rw-rw-r--
Number of links	1
Owner name	jason
Group name	users
Number of bytes in the file	10400
Last modification time	Sep 27 08:52
File name	sales.data

Adding a New User Account

- **useradd** [options] username
- Set account password using **passwd**

  The command-line utility **useradd** provides a simple method for adding new users to the system

  :

  - [root@stationX ~] # **useradd anthony**
  - [root@stationX ~] # **passwd anthony**

  When adding an account with **useradd,** you should consider using the **–m** option to create the user's home folder.

Adding a New User Account

- When you need to add several users, you can use the newusers command with a simple script file .
  - [root@stationX ~] # cat myusers
  - Alice:password:500:500:Alice:/home/alice:/bin/bash
  - Bob:dooow0rd:501:501:Bob:/ home/bob:/bin/bash
  - [root@stationX ~] # **newusers myusers**

- To configure whether or not use SHA or MD5 to generate password:
  - / etc /login.defs needs an entry like this:
  - **ENCRYPT_METHOD** SHA512

Add User

- Create user – useradd

Example: # useradd prince

- No valid password by default – cannot log in

Example: # passwd prince

- sets password

- Creates home directory - /home/prince

Delete User

- **userdel** deletes a user from the system. It may be prudent to lock the user's account first with **usermod –L**, and to delay deletion of the user's account until you are sure that none of the files in the user's home directory are still needed.

- When deleting an account with **userdel,** you should consider using the **–r** option to remove home directory.

Example: userdel bob

- Removes user from /etc/passwd file but home directory is intact

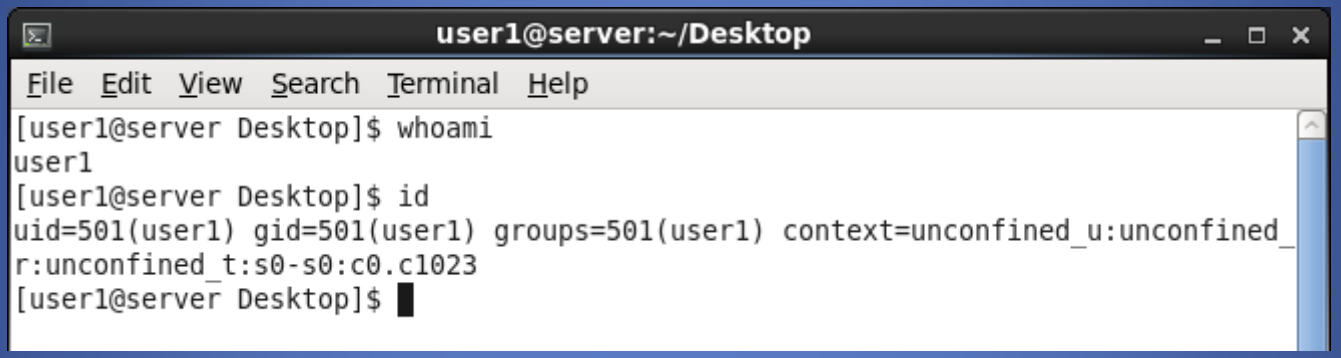
Example: userdel –r prince

- Removes user and home directory

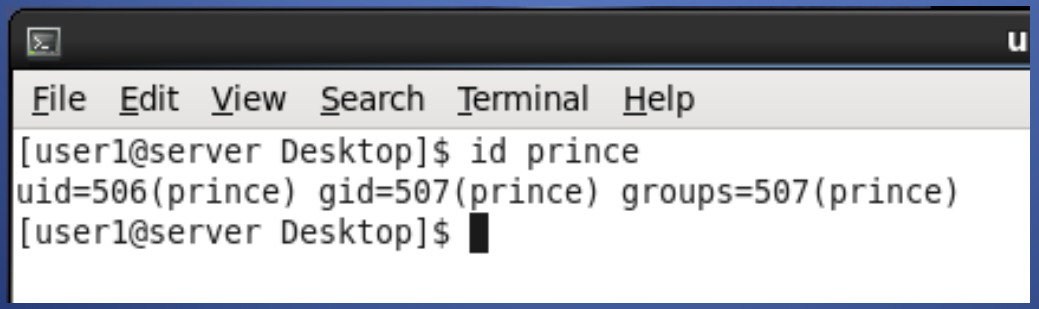
id

- Displays user information

Example: id

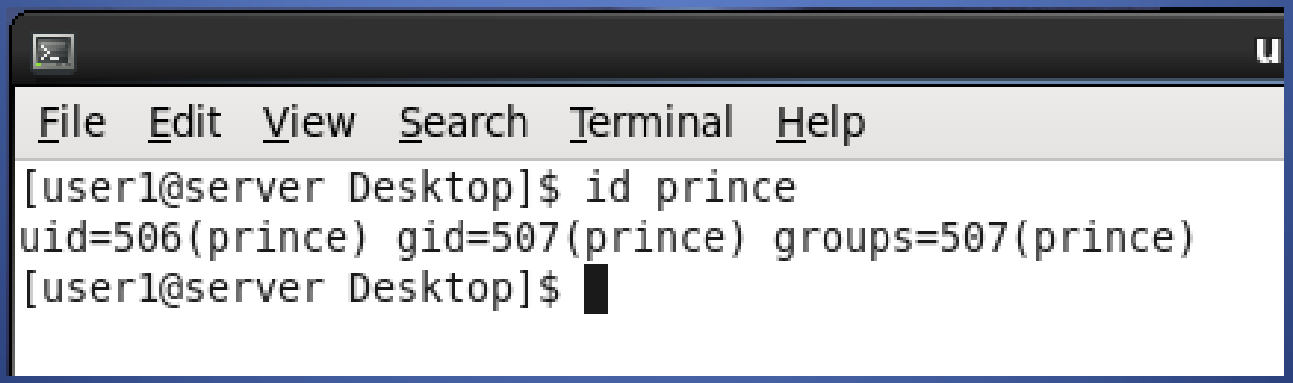


Example: id prince



UID Ranges

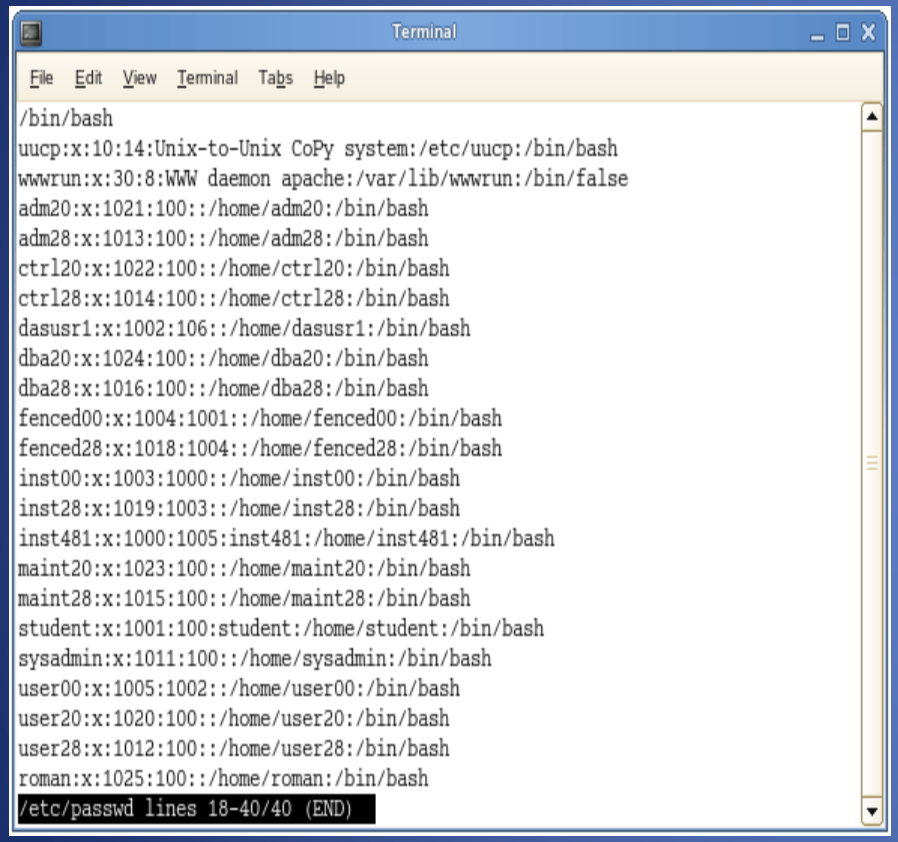
- UID 0 : root, has special privileges
- UID 1 – 499 : system users, non-interactive service accounts
- UID 500 and above : regular users with interactive access to machine

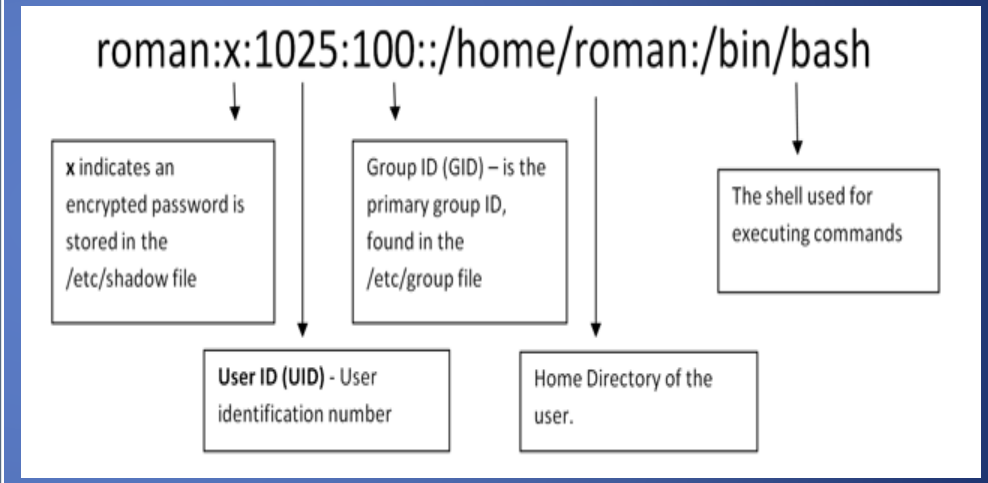


Password File

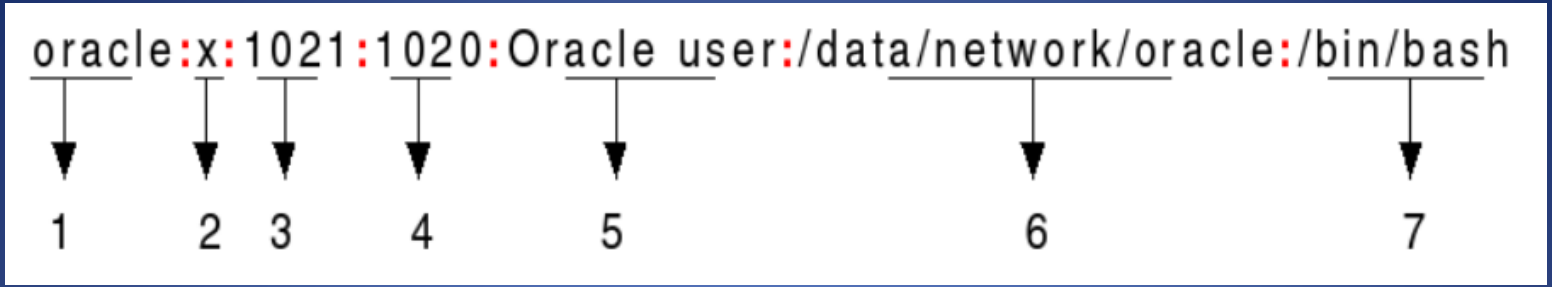
- Passwords stored in /etc/passwd which is a text file
- Passwords migrated to /etc/shadow which supports different password encryption algorithms
- The /etc/passwd file contains seven fields.
- man 5 passwd – see password file format details

/etc/passwd





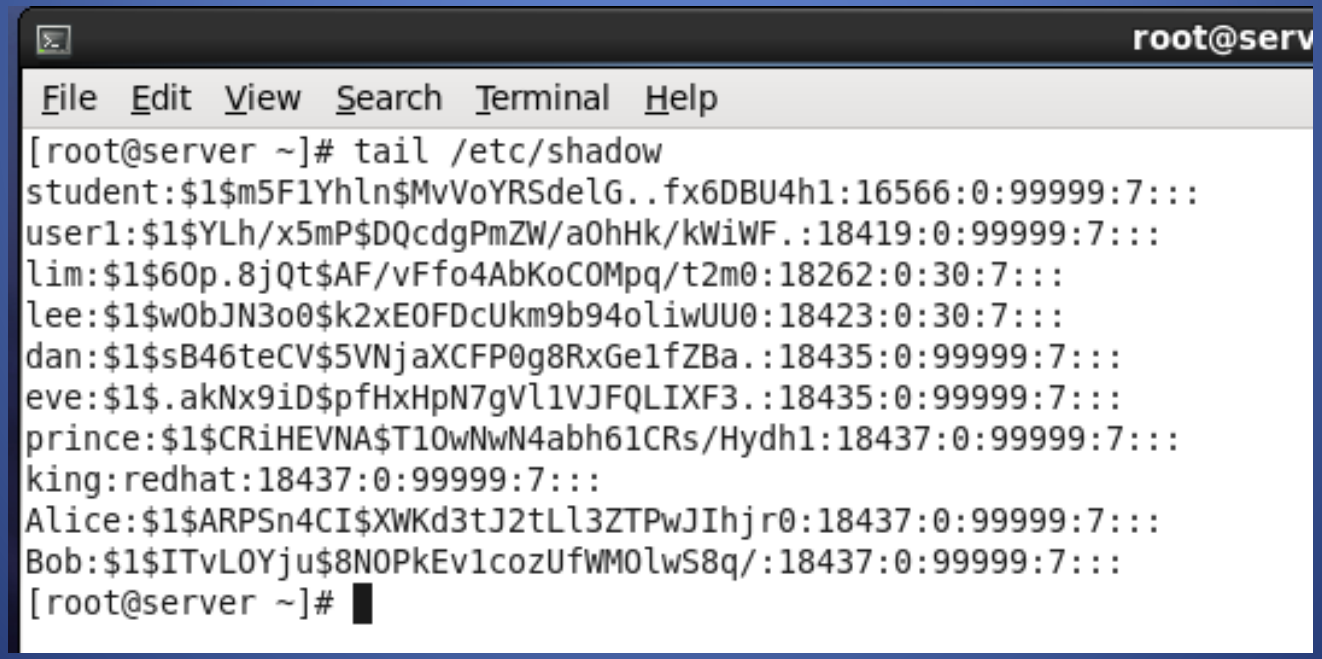
File containing user accounts - /etc/passwd



1. **Username**: It is used when user logs in. It should be between 1 and 32 characters in length.
2. **Password**: An x character indicates that encrypted password is stored in /etc/shadow file. You need to use the passwd command to computes the hash of a password typed at the CLI or to store/update the hash of the password in /etc/shadow file.
3. **User ID (UID)**: Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
4. **Group ID (GID)**: The primary group ID (stored in /etc/group file)
5. **User ID Info**: The comment field. It allows you to add extra information about the users such as user's full name, phone number etc. This field use by finger command.
6. **Home directory**: The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
7. **Command/shell**: The absolute path of a command or shell (/bin/bash). Typically, this is a shell. Please note that it does not have to be a shell.

/etc/shadow

- The /etc/shadow file is referenced when someone logs in. The file contains a mapping of a user name to a password. For example list of the fields, see the man page: shadow(5)



/etc/shadow

username:passwd:last:may:must:warn:expire:disable:reserved

- User login name
- salt and hashed password OR a status exception value
  - "idsalt$encrypted", where "$id" is the hashing algorithm used (On linux, "1" stands for MD5, "2" is Blowfish, "5" is SHA-256 and "6" is SHA-512, etc. Other Unix may have different values).
  - "NP" or "!" or null - No password, the account has no password.
  - "LK" or "*" - the account is Locked, user will be unable to log-in
  - "!!" - the password has expired
- Days since epoch (since Jan 1, 1970) of last password change
- Days until password may be changed
- Days before password must be changed
- Days to warn users for password expiration
- Days since Jan 1, 1970 that the account has been disabled

Password Aging Policies

- The **chage** command is used to modify password aging. It can:
  - Modify user password expiry information
  - View user account aging information
  - Change no. of days between password changes and the date of the last password change.

chage

```
chage [-m mindays] [-M maxdays] [-d lastday] [-I inactive] [-E expiredate] [-W warndays]
user
```

- [root@stationX~] # chage [options] username
- Common options used with the **chage** command:
  - **-m** days — minimum days between password changes
  - **-M** days — maximum days between password changes
  - **-d** days — set the day when the password was last changed.
  - **-E** date — expire the account on this date (yyyy- mm- dd format)

Examples:

`chage -d 0 <username>` — force a password update on next login

`chage -l <username>` — list username current settings

Default Password Policy

- Stored in /etc/login.defs file
- Affect every user that registered in the system
- If you want to setup different rule for different user, use chage command

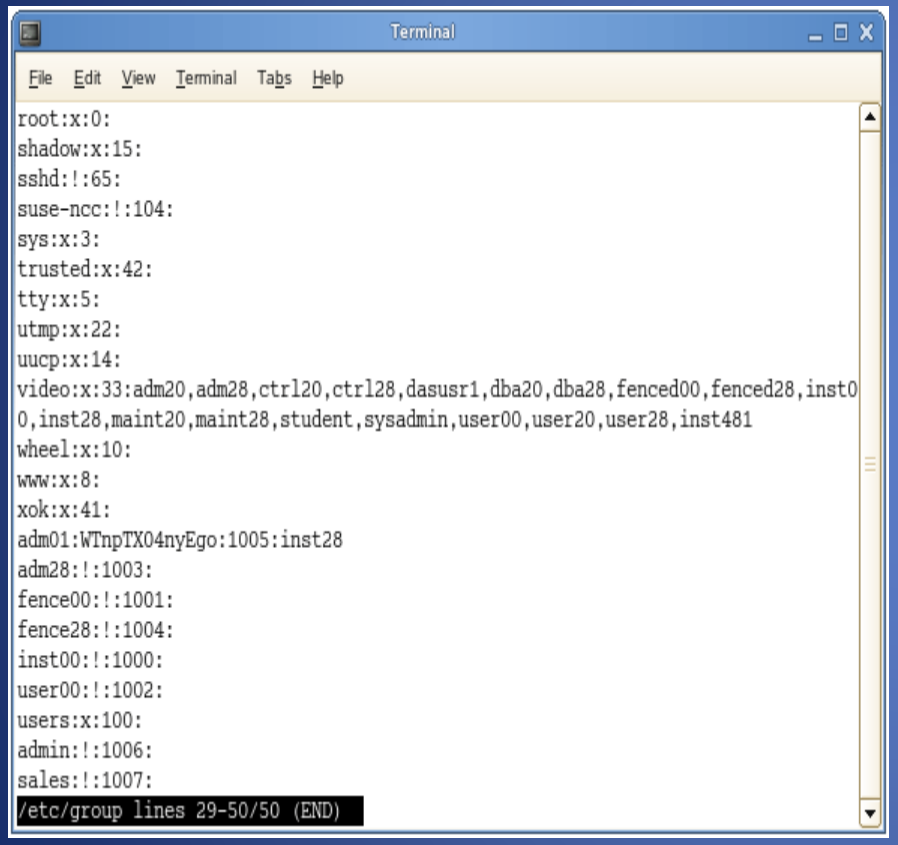
```
# Password aging controls:
#
# PASS_MAX_DAYS Maximum number of days a password may be used.
# PASS_MIN_DAYS Minimum number of days allowed between password changes.
# PASS_MIN_LEN Minimum acceptable password length.
# PASS_WARN_AGE Number of days warning given before a password expires.
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7
```

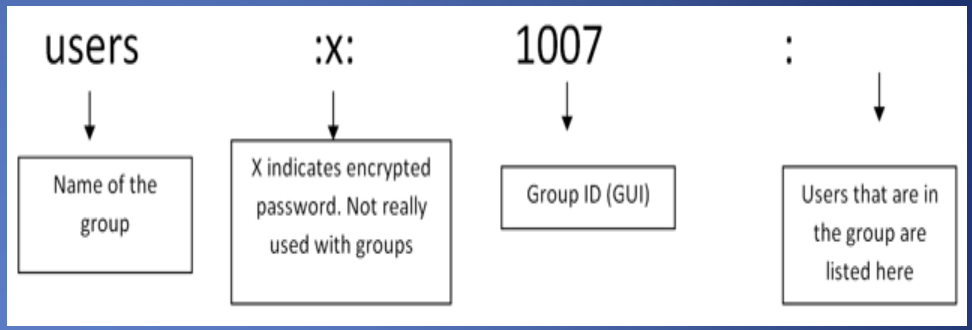
Managing Groups

- A **group** is a collection of users. The main purpose of the **group** is to define a set of privileges like read, write, or execute permission for a given resource that can be shared among the users within the **group**
- Group have name and number (GID)
- Defined in /etc/group
- Every user has one Primary Group
- Normally primary group owns new files created by user
- Normally new user Primary Group name is same as user name and user is the only member.
- The /etc/group file contains four fields.

```
sales:x:506:dan,eve
```

/etc/group





1.group_name: It is the name of group.
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 commas.

File containing group accounts - /etc/group

Supplementary Groups

- Users may be member of zero or more supplementary groups
- Listed in the last field of the group's entry in /etc/group file
- Help users access other files and resources on the system

Example:

/etc/group → sales:x:506:dan,eve

sales group has 2 members – dan & eve

Group Add/Delete

- Creation Group
  - **groupadd [-g** uid] auxgroup
- Add users to group (either):
  - # usermod –a G auxgroup username
  - # gpasswd –a username auxgroup
  - # vigr
- Rename – **groupmod**
- Delete group **- groupdel**

Group Add

- New groups may be created by editing the file **/etc/group** with vigr (editor) or by using **groupadd.** The basic syntax for **groupadd** is very simple :

```
# groupadd groupname
# groupadd –g 201 groupname
```

Example: groupadd –g 505 finance

- GID 0 (zero) is reserved for the root group.
- GID 1– 499 are reserved for the system and application use.
- GID 500+ allocated for the user's group.

Modifying User/Group Account

- To change fields in a user's /etc/passwd entry you can:
  - Edit the file by hand with text editor (eg. nano)
  - Use **usermod** [option] username

Graphical User Interface (GUI)

- **system-config-users** is a graphical tool that can manage both users and groups in these databases.

usermod Command

- Command: usermod options
  - **-e** date
    Set date on which the account will expire and be disabled.
  - **-g** group
    Change the primary group
  - **-G** group,[. . .]
    A comma separated list of All supplementary groups for the user.
  - **- a –G** group
    Append the group to the user's list if auxiliary groups.

- When adding user access to a supplementary group with the **–G** option, you must list all groups to which a user belongs, or include the **–a** option to append the user

system-config-users

Typical User Login Process

- UNIX stores hashed passwords in a password files /etc/shadow and allows only root to have read access
  - *Each user can change his/her password through /etc/passwd program through using setuid*
- When a user logs in through the login process, login process asks for a login name and a password.
- It then hashed the user-entered password and compare it with the respective hashed passwords stored in /etc/shadow.
- Storing hashed passwords is to prevent exposure of the original password.

Command Summary

groupadd - create a new group.
groupdel - delete a group.
groupmod - modify a group.

useradd - create a new user or update default new user information.
userdel - delete a user account and related files.
usermod - modify a user account.

chgrp - changes the group ownership of files.
chown - change the owner of file(s) to another user.

passwd - set a user's pass word.
chage - used to change the time the user's password will expire.

Summary

- Manage user and group accounts in Linux
- Manage password policy