Operating Systems

LS-08. Managing User Accounts on Linux.

Objectives:

- username
- password
- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/gshadow

Managing User Accounts

- Managing user accounts and groups is an essential part of system administration within an organization. But to do this effectively, a good system administrator must first understand what user accounts and groups are and how they work.
- The primary reason for user accounts is to verify the identity of each individual using a computer system. A secondary (but still important) reason for user accounts is to permit the per-individual tailoring of resources and access privileges.
- Resources can include files, directories, and devices. Controlling access to these resources is a large part of a system administrator's daily routine; often the access to a resource is controlled by groups.
- Groups are logical constructs that can be used to cluster user accounts together for a common purpose. For example, if an organization has multiple system administrators, they can all be placed in one system administrator group. The group can then be given permission to access key system resources. In this way, groups can be a powerful tool for managing resources and access.
- User accounts are the method by which an individual is identified and authenticated to the system. User accounts have several different components to them:
  - username,
  - password,
  - resource permissions (access control information).

Username

- From the system's standpoint, the username is the answer to the question, "who are you?" As such, usernames have one major requirement: they must be unique.
- You need is a naming convention for your user accounts should take several factors into account:
  - The size of your organization
  - The structure of your organization
  - The nature of your organization
- Here are some naming conventions that other organizations have used:
  - First name (john, paul, george, etc.)
  - Last name (smith, jones, brown, etc.)
  - First initial, followed by last name (jsmith, pjones, gbrown, etc.)
  - Last name, followed by department code (smith029, jones454, brown191, etc.)

Passwords

- If the username provides an answer to the question, "who are you?", the password is the response to the demand that inevitably follows: "Prove it!”

- The effectiveness of a password-based authentication scheme relies heavily on several aspects of the password:
  - The secrecy of the password
  - The resistance of the password to **guessing** (угадывание)
  - The resistance of the password to a **brute-force** attack

- There are two options available to enforce the use of strong passwords:
  - The system administrator can create passwords for all users.
  - The system administrator can let the users create their own passwords, while verifying that the passwords are acceptably strong.

- Bad Practce:
  - Short Passwords is weak because it is not a problem for brute-force attack.
  - Limited Character Set (only a-z0-9)
  - Recognizable Words (password, 12345)
  - Personal Information (Smit)
  - Simple Word Tricks (p4ssw0rd)
  - The Same Password for Multiple Systems
  - Passwords on Paper
  - Long Term Password Aging

Password Length	Potential Passwords
1	26
2	676
3	17,576
4	456,976
5	11,881,376
6	308,915,776

Managing User Resources & Permissions

- For effectively managing User Resources three points must be considered:
  - Who can access shared data
  - Where users access this data
  - What barriers are in place to prevent abuse of resources

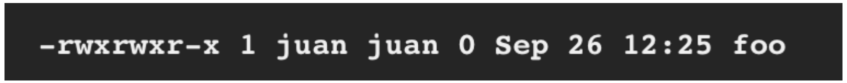
- A user's access to a given application, file, or directory is determined by the permissions applied to that application, file, or directory.

- Shared Groups are used for determined permissions:
  - What groups to create
  - Who to put in a given group
  - What type of permissions should these shared resources have

- One possibility is to mirror your organization's structure when creating groups.

- There are 4 different permissions for files, directories, and applications. Different one-character symbols are used to describe each permission in a directory listing (ls –l):
  - r — Indicates that a given category of user can read a file.
  - w — Indicates that a given category of user can write to a file.
  - x — Indicates that a given category of user can execute the contents of a file.
  - (-) — indicates that no access is permitted.

- Each of the 4 permissions are assigned to three different categories of users:
  - owner — The owner of the file or application.
  - group — The group that owns the file or application.
  - everyone — All users with access to the system.



```
-rwxrwxr-x 1 juan juan 0 Sep 26 12:25 foo
```

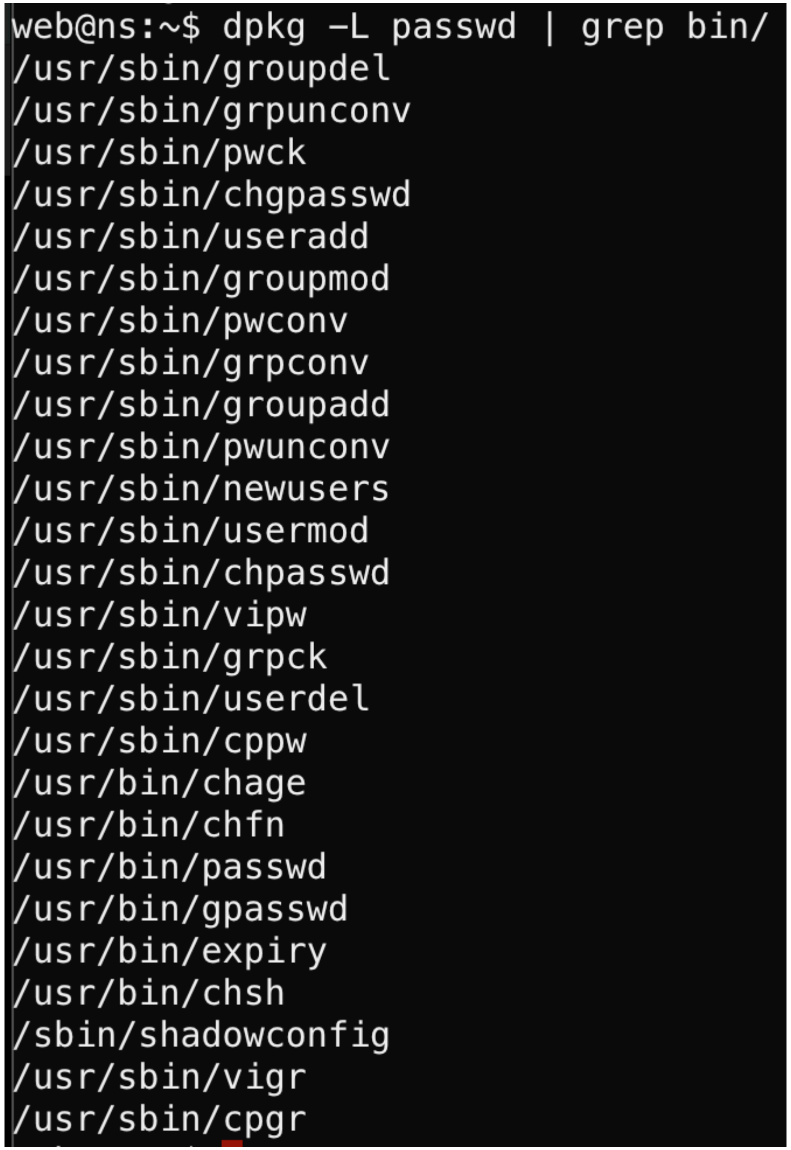
Understanding /etc/passwd File Format

Files Controlling User Accounts and Groups

- On Linux, information about user accounts and groups are stored in several text files within the /etc/ directory.
- When a system administrator creates new user accounts, these files must either be edited manually or applications must be used to make the necessary changes.
- Commands that uses with this files:
  $ dpkg –L passwd | grep bin/

/etc/passwd

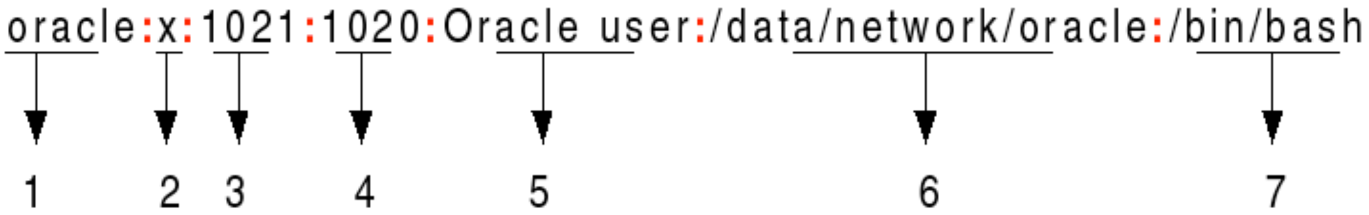
- The /etc/passwd file stores essential information, which is required during login i.e. user account information.
- /etc/passwd is a text file, which contains a list of the system's accounts, giving for each account some useful information like user ID, group ID, home directory, shell, etc.
- It should have general read permission as many utilities like ls use it to map user IDs to user names, but write access only for the superuser/root account.



```
web@ns:~$ dpkg –L passwd | grep bin/
/usr/sbin/groupdel
/usr/sbin/grpunconv
/usr/sbin/pwck
/usr/sbin/chgpasswd
/usr/sbin/useradd
/usr/sbin/groupmod
/usr/sbin/pwconv
/usr/sbin/grpconv
/usr/sbin/groupadd
/usr/sbin/pwunconv
/usr/sbin/newusers
/usr/sbin/usermod
/usr/sbin/chpasswd
/usr/sbin/vipw
/usr/sbin/grpck
/usr/sbin/userdel
/usr/sbin/cppw
/usr/bin/chage
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/expiry
/usr/bin/chsh
/sbin/shadowconfig
/usr/sbin/vigr
/usr/sbin/cpgr
```

Understanding /etc/passwd File Format

The /etc/passwd contains one entry per line for each user (or user account) of the system. All fields are separated by a colon (:) symbol. Total seven fields as follows. Generally, passwd file entry looks as follows:



```
oracle:x:1021:1020:Oracle user:/data/network/oracle:/bin/bash
```

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. Please note that 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 allow 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.

More security today

Your password is stored in /etc/shadow file

Your encrypted password is not stored in /etc/passwd file. It is stored in /etc/shadow file. In the good old days there was no great problem with this general read permission. Everybody could read the encrypted passwords, but the hardware was too slow to crack a well-chosen password, and moreover, the basic assumption used to be that of a friendly user-community.

Almost, all modern Linux / UNIX line operating systems use some sort of the shadow password suite, where /etc/passwd has asterisks (*) instead of encrypted passwords, and the encrypted passwords are in /etc/shadow which is readable by the superuser only.

Tasks for /etc/passwd

See User List

/etc/passwd is only used for local users. To see list of all users, simply use the cat command:

```
$ cat /etc/passwd
```

To search for a username called tom, use the grep command:

```
$ grep tom /etc/passwd
```

OR

```
$ grep -w '^tom' /etc/passwd
```

Sample outputs:

```
tom:x:1000:1000:Vasja Pupkin:/home/vasja:/bin/bash
```

See /etc/passwd file permission

The permission on the /etc/passwd file should be read only to users (-rw-r–r–) and the owner must be root:

```
$ ls -l /etc/passwd
```

Sample outputs:

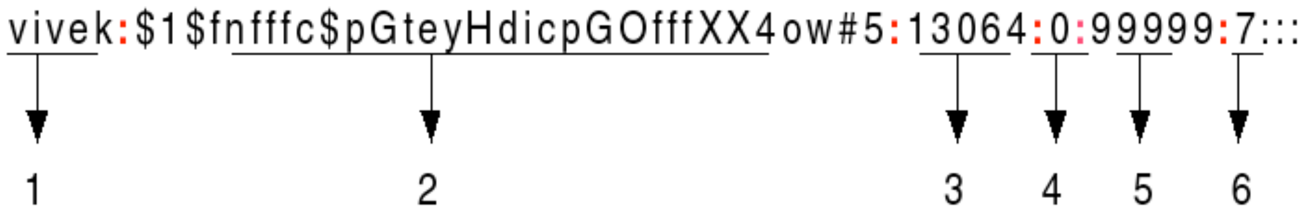
```
-rw-r--r-- 1 root root 2659 Sep 17 01:46 /etc/passwd
```

Understanding /etc/shadow File Format

The /etc/shadow file stores actual password in encrypted format (more like the hash of the password) for user's account with additional properties related to user password.

Basically, it stores secure user account information.

All fields are separated by a colon (:) symbol. It contains one entry per line for each user listed in /etc/passwd file. Generally, shadow file entry looks as follows:

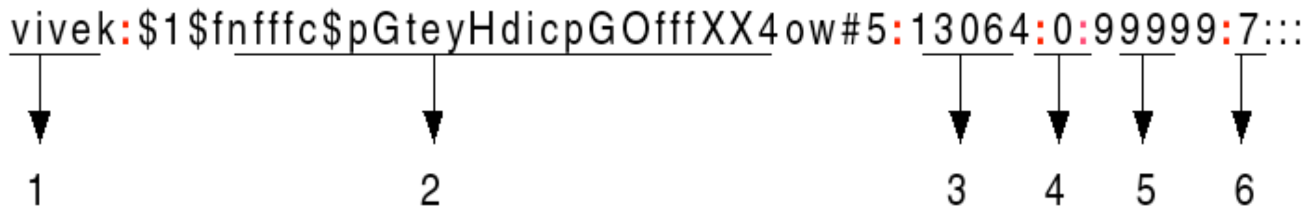


```
vivek:$1$fnfffc$pGteyHdicpGOfffXX4ow#5:13064:0:99999:7:::
```

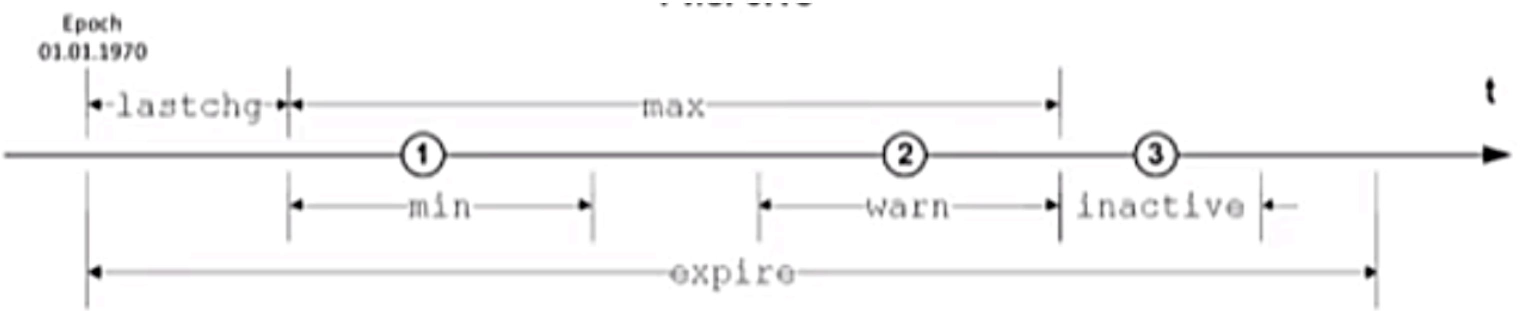
1.Username : It is your login name.
2.Password : It is your encrypted password. The password should be minimum 8-12 characters long including special characters, digits, lower case alphabetic and more. Usually password format is set to $id$salt$hashed, The $id is the algorithm used On GNU/Linux as follows:

1. **$1$** is MD5
2. **$2a$** is Blowfish
3. **$2y$** is Blowfish
4. **$5$** is SHA-256
5. **$6$** is SHA-512

Understanding /etc/shadow File Format



```
vivek:$1$fnfffc$pGteyHdicpGOfffXX4ow#5:13064:0:99999:7:::
```



3. **Last password change (lastchanged)** : 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
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.

Understanding /etc/shadow File Format

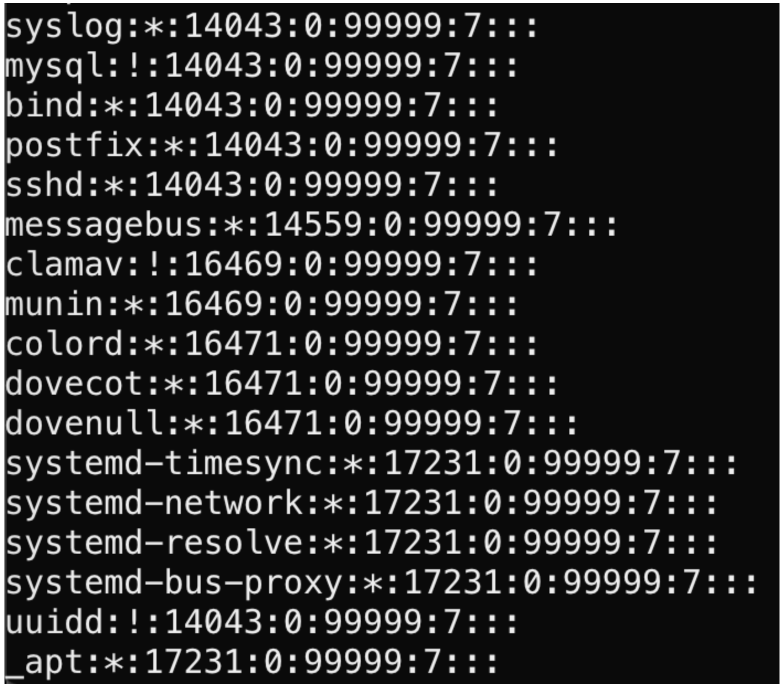
The last 6 fields provides password aging and account lockout features. You need to use the chage command to setup password aging.

According to man page of shadow – the password field must be filled. The encrypted password consists of 13 to 24 characters from the 64 character alphabet a through z, A through Z, 0 through 9, \. and /.

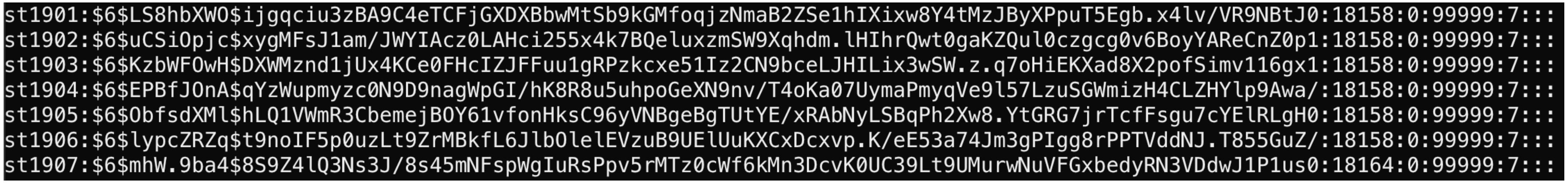
Optionally it can start with a “$” character. This means the encrypted password was generated using another (not DES) algorithm.

For example if it starts with “$1$” it means the MD5-based algorithm was used.

Please note that a password field which starts with a exclamation mark (!)or asterisk (*) means that the password is locked. The remaining characters on the line represent the password field before the password was locked.



```
syslog:*:14043:0:99999:7:::
mysql:!:14043:0:99999:7:::
bind:*:14043:0:99999:7:::
postfix:*:14043:0:99999:7:::
sshd:*:14043:0:99999:7:::
messagebus:*:14559:0:99999:7:::
clamav:!:16469:0:99999:7:::
munin:*:16469:0:99999:7:::
colord:*:16471:0:99999:7:::
dovecot:*:16471:0:99999:7:::
dovenull:*:16471:0:99999:7:::
systemd-timesync:*:17231:0:99999:7:::
systemd-network:*:17231:0:99999:7:::
systemd-resolve:*:17231:0:99999:7:::
systemd-bus-proxy:*:17231:0:99999:7:::
uuidd:!:14043:0:99999:7:::
_apt:*:17231:0:99999:7:::
```



```
st1901:$6$LS8hbXWO$ijgqciu3zBA9C4eTCFjGXDXBbwMtSb9kGMfoqjzNmaB2ZSe1hIXixw8Y4tMzJByXPpuT5Egb.x4lv/VR9NBtJ0:18158:0:99999:7:::
st1902:$6$uCSiOpjc$xygMFsJ1am/JWYIAcz0LAHci255x4k7BQeluxzmSW9Xqhdm.lHIhrQwt0gaKZQul0czgcg0v6BoyYAReCnZ0p1:18158:0:99999:7:::
st1903:$6$KzbWFOwH$DXWMznd1jUx4KCe0FHcIZJFFuu1gRPzkcxe51Iz2CN9bceLJHILix3wSW.z.q7oHiEKXad8X2pofSimv116gx1:18158:0:99999:7:::
st1904:$6$EPBfJOnA$qYzWupmyzc0N9D9nagWpGI/hK8R8u5uhpoGeXN9nv/T4oKa07UymaPmyqVe9l57LzuSGWmizH4CLZHYlp9Awa/:18158:0:99999:7:::
st1905:$6$ObfsdXMl$hLQ1VWmR3CbemejBOY61vfonHksC96yVNBgeBgTUtYE/xRAbNyLSBqPh2Xw8.YtGRG7jrTcfFsgu7cYElRLgH0:18158:0:99999:7:::
st1906:$6$lypcZRZq$t9noIF5p0uzLt9ZrMBkfL6JlbOlelEVzuB9UElUuKXCxDcxvp.K/eE53a74Jm3gPIgg8rPPTVddNJ.T855GuZ/:18158:0:99999:7:::
st1907:$6$mhW.9ba4$8S9Z4lQ3Ns3J/8s45mNFspWgIuRsPpv5rMTz0cWf6kMn3DcvK0UC39Lt9UMurwNuVFGxbedyRN3VDdwJ1P1us0:18164:0:99999:7:::
```

Tasks for /etc/shadow

How do I change the password?

Use the following syntax to change your own password:

```
$ passwd
```

How do I change the password for other users?

You must be root to change the password for all other users:

```
# passwd userNameHere
```

OR

```
$ sudo passwd userNameHere
```

How do I verify integrity of password files?

Use the pwck command verifies the integrity of the users and authentication information.

The user is prompted to delete entries that are improperly formatted or which have other uncorrectable errors. The syntax is:

```
# pwck -r /etc/passwd

# pwck -r /etc/shadow

# pwck [options] /etc/shadow
```

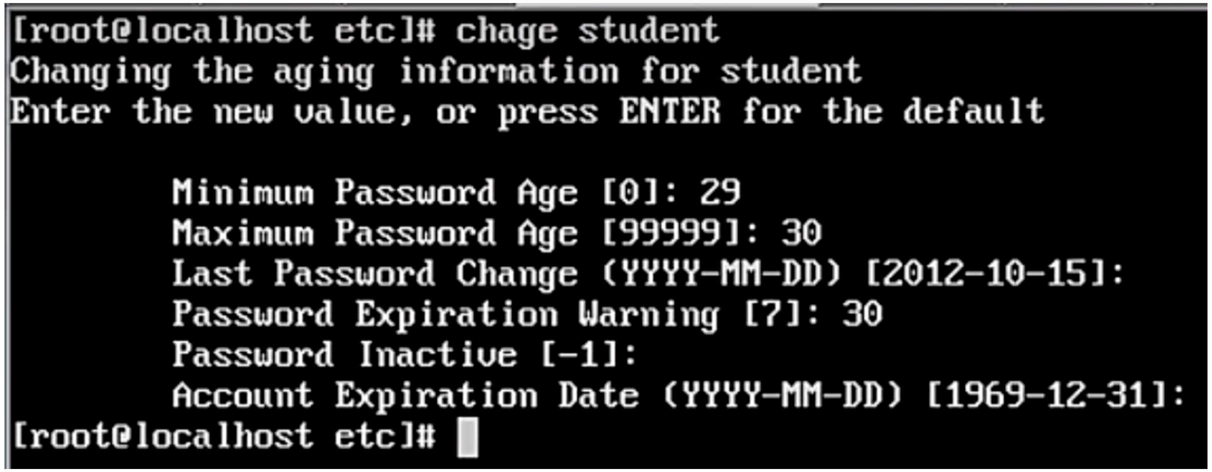
Tasks for /etc/shadow

How do I change or set password ageing information?

To change user password expiry information use the **chage** command on Linux.

The syntax is (again you must be root) as follows chage [options] username:

```
        # chage stud
OR
        $ chage -l student
```



```
[root@localhost etc]# chage student
Changing the aging information for student
Enter the new value, or press ENTER for the default

        Minimum Password Age [0]: 29
        Maximum Password Age [99999]: 30
        Last Password Change (YYYY-MM-DD) [2012-10-15]:
        Password Expiration Warning [7]: 30
        Password Inactive [-1]:
        Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
[root@localhost etc]# █
```

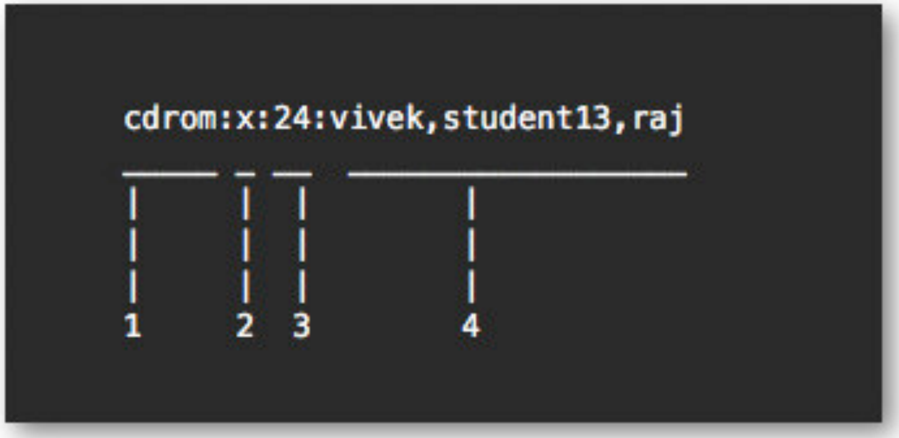
The chage options are as follows:

-d, --lastday LAST_DAY	set date of last password change to LAST_DAY
-E, --expiredate EXPIRE_DATE	set account expiration date to EXPIRE_DATE
-h, --help	display this help message and exit
-I, --inactive INACTIVE	set password inactive after expiration to INACTIVE
-l, --list	show account aging information
-m, --mindays MIN_DAYS	set min number of days before password change to MIN_DAYS
-M, --maxdays MAX_DAYS	set max number of days before password change to MAX_DAYS
-R, --root CHROOT_DIR	directory to chroot into
-W, --warndays WARN_DAYS	set expiration warning days to WARN_DAYS

Understanding /etc/group File Format

The /etc/group is a text file which defines the groups to which users belong under Linux and UNIX operating system. Under Unix / Linux multiple users can be categorized into groups. Unix file system permissions are organized into three classes, user, group, and others. The use of groups allows additional abilities to be delegated in an organized fashion, such as access to disks, printers, and other peripherals. This method, amongst others, also enables the Superuser to delegate some administrative tasks to normal users.

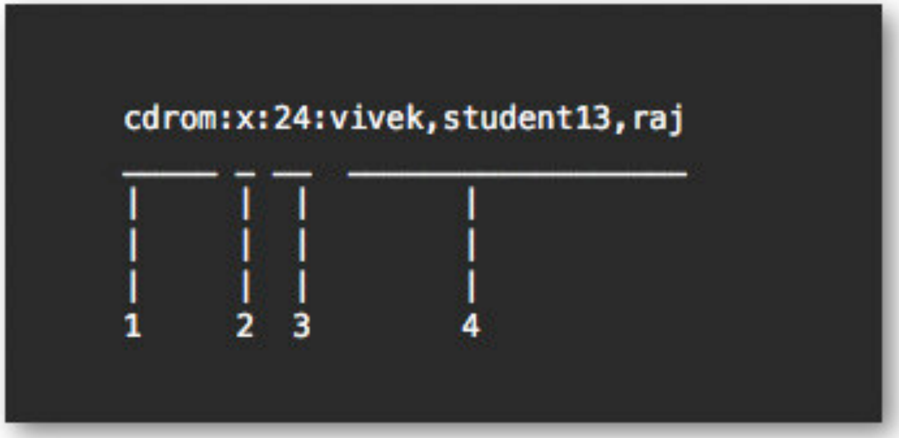
It stores group information or defines the user groups i.e. it defines the groups to which users belong. There is one entry per line, and each line has the following format (all fields are separated by a colon (:)



```
cdrom:x:24:vivek,student13,raj
_____ _ __  __________________
|     | |        |
|     | |        |
|     | |        |
1     2 3        4
```

Understanding /etc/group File Format

It stores group information or defines the user groups i.e. it defines the groups to which users belong. There is one entry per line, and each line has the following format (all fields are separated by a colon (:)



```
cdrom:x:24:vivek,student13,raj
_____ _ __ ___________________
|     |  |        |
|     |  |        |
|     |  |        |
1     2  3        4
```

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.

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.

Tasks for /etc/group

Task: View Current Groups Settings

Type any one of the following command:

```
$ less /etc/group
```

OR use the more command:

```
$ more /etc/group
```

OR use the cat command:

```
$ more /etc/group
```

Task: Find Out the Groups a User Is In

Type the following groups command:

```
$ groups {username}
$ groups
$ groups vivek
```

Sample outputs:

Task: Print user / group Identity

Use the id command to display information about the given user.

Display only the group ID, enter:

Use the id command:

```
$ id -g
$ id -g user
$ id -g vivek
```

OR

```
$ id -gn vivek
```

Display only the group ID and the supplementary groups, enter:

```
$ id -G
$ id -G user
$ id -G vivek
```

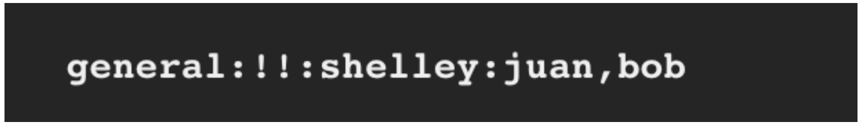
OR

```
$ id -Gn vivek
```

Understanding /etc/gshadow File Format

The /etc/gshadow file is readable only by the root user and contains an encrypted password for each group, as well as group membership and administrator information.

Just as in the /etc/group file, each group's information is on a separate line. Each of these lines is a colon delimited list including the following information:



```
general:!!:shelley:juan,bob
```

Group name — The name of the group. Used by various utility programs as a human-readable identifier for the group.

Encrypted password — The encrypted password for the group.

If **password** set, non-members of the group can join the group by typing the password for that group using the **newgrp** command.

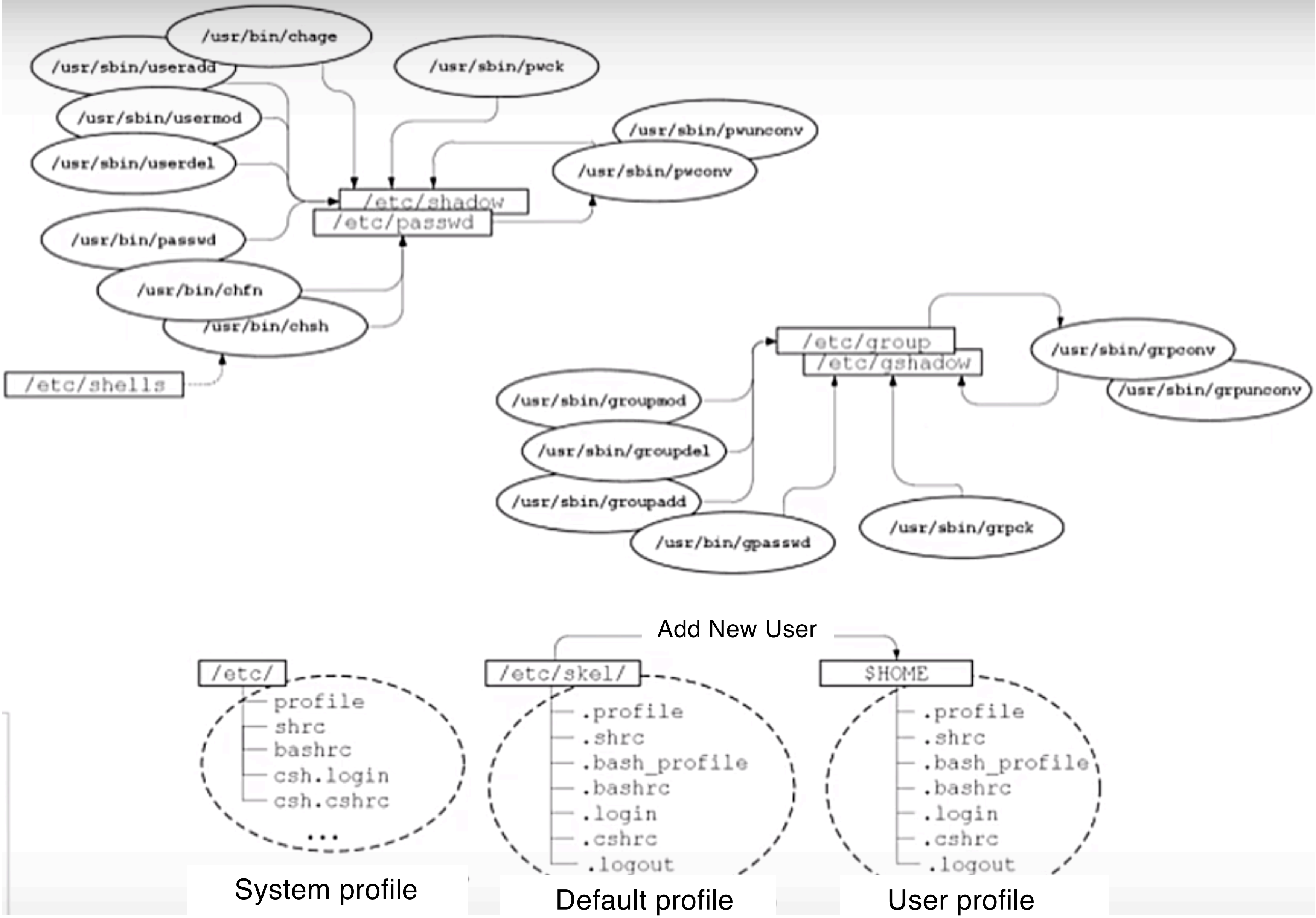
If the value of this field is **!**, then no user is allowed to access the group using the **newgrp** command. A value of **!!** is treated the same as a value of ! — however, it also indicates that a password has never been set before.

If the value is **null**, only group members can log into the group.

Group administrators — Group members listed here (in a comma delimited list) can add or remove group members using the **gpasswd** command.

Group members — Group members listed here (in a comma delimited list) are regular, non-administrative members of the group.

Resume for Linux Protection



System profile

Default profile

User profile

Resume for Linux Protection

```
unix etc # ls -l passwd shadow group
-rw-r--r--  1 root root  705 Sep 23 15:36 group
-rw-r--r--  1 root root 1895 Sep 24 18:20 passwd
-rw-------  1 root root  634 Sep 24 18:22 shadow
unix etc #
```

```
unix root # more /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
...
guest:x:405:100:guest:/dev/null:/dev/null
nobody:x:65534:65534:nobody:/:/bin/false
girtsf:x:1000:100::/home/girtsf:/bin/bash
dima:x:1001:100::/home/dima:/bin/bash
guntis:x:1002:100::/home/guntis:/bin/bash
students:x:1003:100::/home/students:/bin/bash
unix root #
```

```
unix root # more /etc/shadow
root:$1$VlYbWsrd$GUs2cptio.rKlGHgAMBzr.:12684:0:::::
halt:*:9797:0:::::
...
guest:*:9797:0:::::
nobody:*:9797:0:::::
girtsf:$1$u6UEWKT2$w5K28n2iAB2wNWtyPLycP1:12684:0:99999:7:::
dima:$1$BQCdIBdV$xzzlj4s8XT6L9cLAmcoV50:12684:0:99999:7:::
guntis:$1$fiJF/0BT$Py9JiQQL6icajjQVyMZ7//:12684:0:99999:7:::
students:$1$wueon8yh$nLpUpNOKr8yTYaEnEK6OJ1:12685:0:99999:7:::
unix root #
```

Only “wheel” group have *su* to root; See on /etc/pam.d/

```
unix root # more /etc/group
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:girtsf
disk::6:root,adm
lp::7:lp
mem::8:
kmem::9:
wheel::10:root,girtsf
floppy::11:root
mail::12:mail
...
users::100:games,girtsf
nofiles:x:200:
qmail:x:201:
postfix:x:207:
postdrop:x:208:
smmsp:x:209:smmsp
slocate::245:
portage::250:portage
utmp:x:406:
nogroup::65533:
nobody::65534:
unix root #
```

The End