Linux - User management

First step of the administrator

In case several people have access to a system, it is necessary for the administrator to manage the users. To do so, the usual commands and the files to be configured must be known.

You should know the following important files:

- the file **/etc/passwd**
- the file **/etc/group**

The file /etc/passwd

The file **/etc/passwd** contains all information regarding the user (login, passwords, etc.). Only the superuser (root) must be able to change it. It is therefore necessary to change the rights of this file so that it can only be ready by the other users.

This file has a special format which makes it possible to mark each user, and each of its lines has the following format:

account_name : password : user_number : group_number : comment : directory : start_program
Seven fields are specified separated by the character ":":

- the **account name** of the user
- the **password** of the user (encoded, of course)
- the **integer** identifying **the user** for the operating system (UID=User ID, user identification)
- the **integer** identifying the **group** of the user (GID=Group ID, group identification)
- the **comment** in which the information on the user or simply its real name can be found
- the **connection directory**, which is directory which opens upon connection to the system
- the **command** is the one that is executed **after connection** to the system (often, this is the command interpreter)

Here is an example of a ***passwd*** file:

```
root:x:0:0:root:/root:/bin/bash
```

bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/bash
news:x:9:13:News system:/etc/news:/bin/bash
uucp:x:10:14::/var/lib/uucp/taylor_config:/bin/bash
cquoi:x:500:100:Cool.....:/home/cquoi:/bin/bash

It is important to know that the passwords located in this file are encrypted. It is therefore useless to edit and replace the field ***password*** by directly typing the password, which would only cause the account to be blocked.
Once a user connects, the login program compares the password typed in by the user (after encrypting it) with the password stored in the passwd file. If they do not match, the connection can not be established.

To prohibit use, it is sufficient to replace the encrypted password by a star: "*****".
Access to an account may be opened by leaving the field ***password*** open. Any person who wishes to connect via the account can then do so.
To be able to modify the password of an account with the command ***passwd***, you must either be the system administrator or the account owner (the system will than require that you enter the old password before asking you to enter the new password twice).
UID: (unique) identifier of each user account. Numbers beween 0 and 99 are frequently reserved for the machine's own accounts. Numbers higher than 100 are reserved for user accounts.
GID: group identifier. The default group (called **group**) has the number 50. This identifier is used in connection with access rights to the files. This question will not concern you if your system has more than one user group. (In that case, you must pay attention to the file ***/etc/group***.
From the shell, it is possible to modify the command interpreter. To do so, use the command ***chsh*** or ***passwd -s***. Linux will look for program you have specified in the file ***/etc/shells***. Only commands that are present in this file will be accepted and will replace the current value of the field ***start_program***. These restrictions do not apply to the superuser account.
Make sure that the access rights to the file ***/etc/shells*** are the same as for the file ***/etc/passwd***
The superuser may not necessarily be called **root**. To change this, just replace the root account name by the desired name.
A privileged account is an account whose identifier (UID, User ID) is zero.

The file /etc/group

The file **/etc/group** contains a list of the users who belong to the different groups. As a matter of fact, whenever a large number of users may have access to the system, they are frequently placed in different groups, each of which has it own access rights to the files and directories.
It has different fields that are separated by ":":
groupe_name : special_field : group_number : member1, member2

The special field is frequently blank.
The group number is the number which makes the link between the /etc/group and /etc/passwd files.
Here is an example of a */etc/group* file:

root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:
tty:x:5:
disk:x:6:
lp:x:7:
wwwadmin:x:8:
kmem:x:9:
wheel:x:10:
mail:x:12:cyrus
news:x:13:news

- When the *ls* command is used with the option -l, the group number is displayed with the number of the user to whom the file (or the directory) belongs. This unique number corresponds to a unique group name (often 8 characters max.).
- The same user can appear in several groups. When he connects to the system he belongs to a group specified in the /etc/passwd (in the GID field). He can change this using the *newgrp* command. The file access rights are then defined.
- File protections must prevent the modification of files by non-privileged users.
- To add a group, the administrator can change the */etc/group* file using a text editor. He can also use the **addgroup** or **groupadd** command (not always present). In the first instance, he will only have to add the line(s) relating to the groups. For example, the line:
  admin : : 56 : ccm

- To add a user to a group, just edit the file */etc/group* and add the name at the end of the line by separating the names of the members by a comma.
- To delete a group, edit the */etc/group* file and delete the corresponding line. Please note, do not forget to change the the numbers (GID) of the deleted group in the /etc/passwd file, if users belonged to it. It is also important to search the files and directories of this group to change this (otherwise, the files and directories may become inaccessible).

Customizing the shell

Use the file **/etc/profile** to configure the shell. This pertains to all users.
First of all, you will find the variables of the shell such as OPENWINHOME, PATH, etc. Afterwards, the type of terminal and the TERM variable are defined. One part is reserved for the shell prompt, and a last one makes it possible to define the colors of the command */s*.
Upon startup of Linux, it is recommendable to have the numeric pad illuminated, which is not the case by default.
You therefore have to add the following lines to the file */etc/profile*:

```
INITTY=/dev/tty[1-7]
for tty in $INITTY;
do setleds -D +num < $tty
done
```

Upon connection to the shell, the first thing that appears is the **prompt**, which can be configured at the user's discretion.
In case the administrator wishes a prompt which reads: "**Hello#**", just edit the file/**etc/profile**. This file contains a variable called **PS1**. All lines pertaining to this variable must then be preceded by a number sign: **#**. The line **PS1='Hello#'** must be added.
Just save an log on again. Some changes will be noted.
Tip: leave a blank space after the prompt to improve readability.

It is also possible to use variables in the prompt (for example, to display the time or name of the machine, etc.):

\d	to add the date (English format)

\t	to add the time(HH:MM:SS)
\u	to add the user name
\r	to return to the line
\w	to add the full path of the current directory
\W	to add the current directory
\h	to add the name of the machine

The color may also be changed. To do so, use the variable **PS1** as follows:
PS1='\[\033[*num_color*m]*desired_prompt*\033[0m]'
The color number is shown in the list below:

Black	0;30
Red	0;31
Green	0;32
Brown	0;33
Blue	0;34
Violet	0;35
Cyan	0;36
Light Gray	0;37
Gray	1;30
Pink	1;31
Light Green	1;32
Light Brown	1;33
Light Blue	1;34
Light Violet	1;35
Light Cyan	1;36
White	1;37

Here is an example which shows the time followed by the user name in red:

PS1='\t \[\033[0;31m]\u\033[0m]'

Source: http://en.kioskea.net/contents/328-linux-user-management