LPIC-1 102-500 – Lesson 6

107.1 Manage user and group accounts and related system files

User Administration

- One of the most important duties of a System Administrator is the administration of users.

- Only the root user with UID 0 has the right to user administration.

- Every user has a name and a UID which makes them identifiable by the system.

- Users belong to one or more groups which have their own GID.

User categories

- **Superusers**: basically this is the **root** user with a UID of 0. The root user has full administration rights on a system.

- **System users**: these are accounts used by daemons or system services. Their UIDs range between 1 and 999.

- **Regular users**: these are accounts created for use by actual persons. Their UIDs start from 1000 to 65535.

The */etc/passwd* file

- All users in a Linux system are defined to the **/etc/passwd** file that looks like:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
...
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
sshd:x:113:65534::/var/run/sshd:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false
user:x:1000:1000:User
Userides,23,99773377,Boss:/home/user:/bin/bash
george:x:1001:1001:George Papas,,,:/home/george:/bin/csh
```

Fields in the */etc/passwd* file

root:x:0:0:root:/root:/bin/bash # root user
sshd:x:113:65534::/var/run/sshd:/usr/sbin/nologin # system account
user:x:1000:1000:User Userides,,,:/home/user:/bin/bash # normal account

- **username**: a unique name up to 32 characters. It is case sensitive but generally it is recommended to use lowercase latin characters. Dashes (-) and underscores (_) are allowed but the name must start with a letter.

- **password**: This is a legacy field that used to store the hashed password. It is no longer used in modern systems and has been replaced by the **/etc/shadow** file. So here we only have an **x** to designate that the password is saved elsewhere.

- **UID**: it is a numeric value the defines the identity of a user. This value is saved in the inodes to set the owner of a file.

- **GID**: it is a numeric value the defines the identity of the user's primary group.

Fields in the */etc/passwd* file

- **Full name or comments**: Here we can have some comments related to the purpose of the account. In the case of regular users we can have the Full Name and maybe some additional information like the room number, phone or any other comments separated by a comma "**,**". It can be completely omitted.

- **Home Directory**: identifies the personal directory for regular accounts and the working directory in the case of system accounts. On regular users in inherits files and properties from the **/etc/skel** directory, on creation.

- **Shell**: identifies the default shell of a user that starts after login. In many cases of system users it is nullified by **false** or **nologin**. This is a security practice to prevent login for these accounts. It can also be used if we want to disable a user from login to shell.

The */etc/shadow* file

- The **/etc/passwd** file has a global read permission (644) because it needs to be accessible for all users. This posed a security risk because a user could copy the hashed password and try to crack it.

- To solve this issue the Shadow Password subsystem was created to keep the hashed password and some other information under the **/etc/shadow** file which is not globally readable.

Fields in the */etc/shadow* file

root:**6CGpBa7JP$ROaMeOyDIQSKNNrZUFnjhAXgfYwpMqJf7G/tt/**:**15316:0:99999:7:::**
sshd:*****:15214:0:99999:7:::**
user:**$6$04ZF/yDL$Yc2crio0H.A.KlkvhmfnUF/eB0ibQac5mjkXjrjO5/k/**:**15221:0:99999:7:::**

▪ **username**: this filed is identical to the username field in **/etc/passwd**.

▪ **password**: here we store the hashed password Traditionally the encryption algorithm had been the **crypt** (**des**) but in modern systems we use **md5 or sha**. This field can have a value of "*****" or "**!**" when there is no password. In this case the user cannot login. This usually happens for system accounts where a login is not justified.

▪ **Other fields**: they are used by commands like **passwd**, **usermod** and **chage** to set parameters like date of last password change, number of days after the last password change, account expiration date, etc.

The */etc/group* file

- The groups of the system are declared in the **/etc/group** file. A user can be a member of many groups but only one is the primary group. The GID of a user's primary group is declared in the **/etc/passwd** file. This is an example for the **group** file:

```
root:x:0:
daemon:x:1:
bin:x:2:
adm:x:4:theodotos,george
tty:x:5:
admin:x:121:theodotos
theodotos:x:1000:
sambashare:x:122:theodotos
winbindd_priv:x:123:
mysql:x:124:
george:x:1001:
```
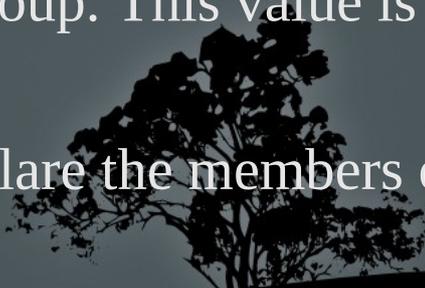
Fields in the */etc/group* file

root:**x**:**0**: # root group
nogroup:**x**:**65534**: # system group
plugdev:**x**:**46**:**user,george** # system group
user:**x**:**1000**:: # personal group of a regular user

- **group name**: a unique group name. Groups controls the access to various services, files and resources. There are personal group that are created when a new user is created and they usually have the same name as the user.

- **password**: Here we used to have the hashed password for a group. This password can be used to give access to a non member on a resource controlled by a group. In modern systems it has been replaced by **/etc/gshadow** and we just have an **x**

- **GID**: A numeric value that declares the ID of the group. This value is saved in the inodes to set the group of a file.

- **group members**: it is a comma separated list to declare the members of a group.

The */etc/gshadow* file

- The **/etc/group** file has a global read permission (644) because it needs to be accessible by all users. This is a security risk for the same reasons as **/etc/passwd**.

- To solve this problem we use the **Shadow Group** subsystem to keep the hashed passwords in the **/etc/gshadow** file which is not globally readable.

Fields in the */etc/gshadow* file

root:*****::
sshd:**!**::
sales:**$6$04ZF/yDL$Yc2crio0H.A.KlkvhmfnUF/eB0ibQac5mjkXjrjO5/k/**::

- **group name**: the name in this field is identical yo the group name in
  /etc/group

- **password)** the hashed password is saved in this field. Just like the **shadow**
  file we use **md5** or **sha** for the encryptiom. A "*****" or "**!**" value means
  there is no password set. The passwords in these groups are rarely used
  and for different reasons than the password of a user.

- **Other fields**: used to declare the admin and members of the team.

Adding users with `useradd`

- The **useradd** command is the basic command for adding users in the system.

- # **useradd user1** # add the user '**user1**' in the system. We should later use the command **passwd** to set the password.

- # **useradd -m user1** # add user **user1** in the system and at the same time create a home directory based on the **/etc/skel** template.

- # **useradd -m -c "User Userides" -s /bin/bash user1** # add user **user1**, create a home directory and set the default shell of the user to **bash**.

- # **useradd -D** # show the default settings for useradd. These are configured in the **/etc/login.defs** configuration file.

Adding users with `useradd`

Options:

- **-m** # create a home directory under **/home** with the same name as the user.

- **-d** # set a different home directory that the default. It does not imply creation of that directory unless it is combined with **-m**.

- **-c** # add the comment field. In the case of a normal user that can be the username.

- **-e** # expiration date in the form YYYY-MM-DD

- **-f** # number of days until the actual deactivation of the account past the expiration date.

- **-r** # create a system user (UID less than 1000).

- **-D** # show the default settings from **/etc/login/def**.

Set the password with `passwd`

- The **passwd** command sets the access password of a user. A new user will remain disabled unless we set a password.

- It is recommended to use a password with suficient complexity. This should be specified by the organization's security policy.

- **# passwd user1** # set or reset the password of user **user1** from the **root** user.

- **$ passwd** # change the password of the active user (as set by the **$USER** variable). Normal users can only change their own password and the have to enter the old password first.

Set the password with `passwd`

Options:

- **-d** # delete user password (deactivates account).
- **-e** # force expiration of the account on the next login and change the password after a successful login.
- **-i DAYS** # deactivate the account after the expiration of the account the days set in DAYS pass without changing the password.
- **-n DAYS** # minimum number of days where no password change is allowed.
- **-m DAYS** # maximum number of days allowed before you are required to change the password.
- **-l** # lock an account by adding '!' in front of the hashed password
- **-u** # unlock account.
- **-w DAYS** # set password expiration warning after the days defined in DAYS, before the expiration of the account.
- **-x DAYS** # set the maximum number of days where a password must be expired.

Modify users with `usermod`

- The **usermod** command is used to change the information and settings of the existing users.

- **# usermod -g users user1** # change the primary group of **user1** to **users**.

- **# usermod -a -G admin user1** # add **user1** to the supplementary group **admin**.

- **# usermod -L user1** # lock account.

- **# usermod -U user1** # unlock account.

Modify users with `usermod`

- **-a** # add user to supplementary group. Always combined with **-G**.
- **-d** # change the home directory name in **/etc/passwd**. It can be combined with **-m** to relocate the existing home directory.
- **-c** # change comment.
- **-e** # set the expiration date in the form YYYY-MM-DD
- **-f DAYS** # deactivate the account after the expiration of the account the days set in DAYS pass without changing the password.
- **-g** # change the user's primary group.
- **-G group1,gpoup2,group3** # add the user to these supplementary group. Combined with **-a** if we want to keep the existing supplementary groups.
- **-l** # change username.
- **-L, -U** # lock, unlock account.
- **-s** # reconfigure the user's default shell.

Delete users with `userdel`

- The **userdel** command is used to delete users from the system.

- # **userdel user1** # delete the user from the system but keep the home directory.

- # **userdel -r user1** # delete the user from the system along with the home directory.

- # **userdel -f user1** # delete users from the system even if they are already connected!

Create group with `groupadd`

- The **groupadd** command is used to create groups on the system.

- # **groupadd sales** # create a normal user group (GID > 999).

- # **groupadd -r httpd** # create system group (GID < 1000).

- # **groupadd -g 42 httpd** # create a group with the GID set to 42.

Set a group password with `gpasswd`

- The **gpasswd** command is used to set the group password for the group but also to make changes in the group (man gpasswd for more info).

- This password can be used by non-members to access a directory which belongs to the group.

- `#` **`gpasswd httpd`** `# set group password.`

Modify a group with `groupmod`

- The **groupmod** command is used to modify existing groups in the system.

- `#` **`groupmod -g 50 httpd`** `# change GID to 50.`

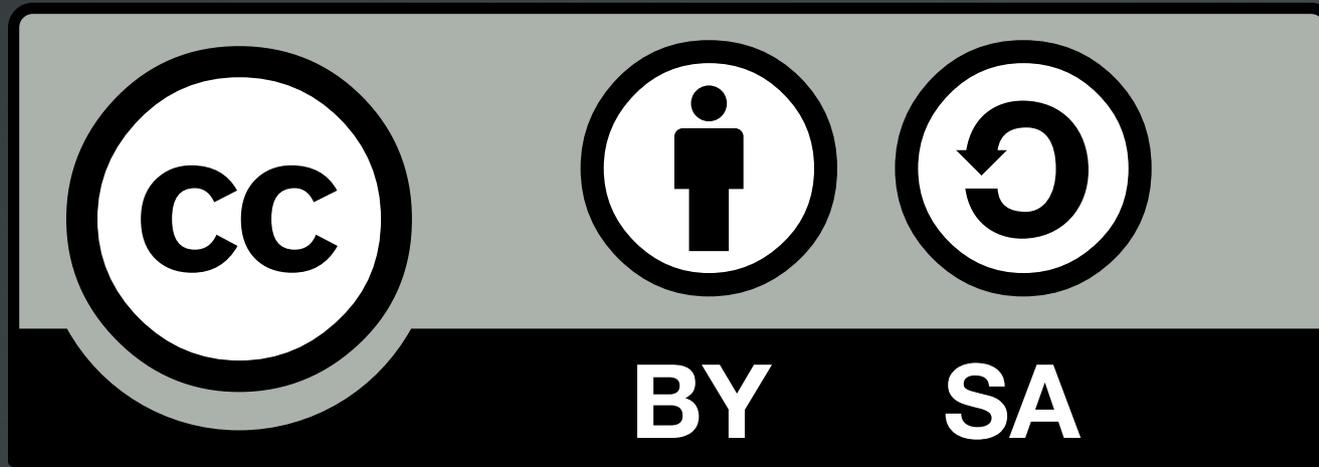- `#` **`groupmod -n apache httpd`** `#` `change group name from` **`httpd`** `to` **`apache`**`.`

Delete a group with `groupdel`

- The **groupdel** command is used to delete a group from the system.

- # **groupdel httpd** # delete the **httpd** group.

License