ADS Lab 07 - Account Management

Authors: Samuel Roland, Antoine Leresche, Nicolas Carbonara
Date: 2025-05-25

Task 0: Examine the setup of your own account

> Note
>
> What is its principal group ?

The principal has the same name as the account name, which is in my case `a2va`.

> Note
>
> What other groups is the account a member of?

`proj_a` and `proj_b`

> Note
>
> What is the UID of the account and the GID of the principal group?

The UID is `1000` and the GID is `1000`

> Note
>
> Which skeleton files have been copied?

On the Ubuntu Server VM, the folder `/etc/skel` contains the following files:

- `.bash_logout` -> Used to clear the console when you logout and is executed by bash(1).
- `.bashrc` -> basic bash configuration for a user.
- `.profile` -> configures the PATH for the user if `.bash_profile` or `.bash_login` are not present.

Task 1: Create user accounts

```
getent group jedi rebels # checking if the 2 groups do exist
sudo groupadd jedi
sudo groupadd rebels
```

> Note
>
> What option do you need to specify to have `useradd` create a home directory?

There are two ways to do it: Either use the `-d` flag with the path to the home directory or use the `-m` flag without the path (it will infer the home directory path from the username).

> Note
>
> What is the default login shell for users created with useradd ? What command should we use to change the default login shell from `/bin/sh` to `/bin/bash` ?

The default login shell is `/bin/sh`. To change it to `/bin/bash`, we can use the `chsh` command.

> ⓘ Note
>
> Create the following user accounts with default home directories and login shell

```
id luke vader solo # checking if the accounts exist
sudo useradd -s /bin/bash -m -g jedi -G rebels luke
sudo passwd luke
sudo useradd -s /bin/bash -m -g jedi vader
sudo useradd -s /bin/bash -m -g rebels solo
```

Task 2: Change group membership

Perform the following steps and give in the lab report the commands you used.
Use the tool `usermod`.

> ⓘ Note
>
> Create the account leia without assigning it a principal group. After it was created, which principal group did it get assigned?

```
$ sudo useradd -m leia
$ cat /etc/passwd
...
leia:x:1001:1001::/home/leia:/bin/sh
$ groups leia
leia : leia
```

It also created a new group with the same name as the username.

> ⓘ Note
>
> Make leia member of the group rebels (as secondary group).

```
$ sudo usermod -a -G rebels leia

# Alternative (prefered for consistency with removing)
# $ sudo gpasswd -a leia rebels

$ groups leia
leia : leia rebels
```

> ⓘ Note
>
> Make leia leave the group rebels and join the group jedi instead.

```
# Leave the group rebels
# We can use -r to remove a supplementary group given after -G
$ sudo usermod -r -G rebels leia

# Alternative way
# $ sudo gpasswd -d leia rebels
# Removing user leia from group rebels

# Join the group jedi
$ sudo usermod -a -G jedi leia

# Alternative
# $ sudo gpasswd -a leia jedi
# Adding user leia to group jedi

$ groups leia
leia : leia jedi
```

> ⓘ Note
>
> Make leia leave any secondary group.

```
$ sudo usermod -G "" leia # setting the list of supplementary groups to empty
$ groups leia
leia : leia
```

Task 3: Give a user sudo rights

> ⓘ Note
>
> a. Which line in `/etc/sudoers` gives the members of the group sudo the right to execute any command?

The line `%sudo    ALL=(ALL:ALL) ALL` is denoted by a comment saying that it gives the members of the sudo group the right to execute any command.

> ⓘ Note
>
> b. How would you have to modify this line so that users can use sudo without typing a password (this is in general not recommended, but can be handy sometimes).

It could be modified like this: `%sudo ALL=(ALL:ALL) NOPASSWD: ALL`

```
sudo usermod -aG sudo luke # make luke part of the sudo group
su luke # login in another session to see the change applied
sudo cat /etc/sudoers # now luke can use sudo
sudo usermod -rG sudo luke # remove luke from the sudo group
```

Task 4: Remove a user account

> Perform the following steps and give in the lab report the commands you used. Use the tool `userdel`.

> ⓘ Note
>
> Remove the account `leia`, but do not delete the home directory yet.

```
$ sudo userdel leia
```

> Note
>
> Inspect the home directory (look at the file metadata). What has changed?

```
$ sudo ls -la leia/
total 20
drwxr-x--- 2 leia leia 4096 May 19 14:07 .
drwxr-xr-x 4 root root 4096 May 19 14:07 ..
-rw-r--r-- 1 leia leia  220 Mar 31  2024 .bash_logout
-rw-r--r-- 1 leia leia 3771 Mar 31  2024 .bashrc
-rw-r--r-- 1 leia leia  807 Mar 31  2024 .profile

$ sudo ls -la leia/
total 20
drwxr-x--- 2 1001 1001 4096 May 19 14:07 .
drwxr-xr-x 4 root root 4096 May 19 14:07 ..
-rw-r--r-- 1 1001 1001  220 Mar 31  2024 .bash_logout
-rw-r--r-- 1 1001 1001 3771 Mar 31  2024 .bashrc
-rw-r--r-- 1 1001 1001  807 Mar 31  2024 .profile
```

The files retain the UID of their owner but as the owner doesn't exist anymore, it doesn't display its name `leia` so we see the UID directly. The permissions have not changed.

> Note
>
> Suppose the user leia has created other files on the system, but you do not know where they are. How would you systematically scan the whole system to find them?

I would use the `find` command with the UID of said account/user.

```
$ sudo find / -uid 1001
/home/leia
/home/leia/.bashrc
/home/leia/.profile
/home/leia/.bash_logout
find: ‘/proc/2305/task/2305/fd/6’: No such file or directory
find: ‘/proc/2305/task/2305/fdinfo/6’: No such file or directory
find: ‘/proc/2305/fd/5’: No such file or directory
find: ‘/proc/2305/fdinfo/5’: No such file or directory
```

Note that the sudo is important otherwise a lot of "Permission denied" error mesage will be printed burying the information we are looking for.
Possible work around to avoid using `sudo` would be to redirect the stderr to `/dev/null` with adding that `2>/dev/null` at the end of the command.

> Note
>
> Remove the home directory manually.

```
$ pwd
/home
$ ls
leia  syseria
$ sudo rm -rf /home/leia/
$ ls
syseria
```