Lab7 - Account management

ADS

Felix Breval, Anthony David, Timothée Van Hove

May 25, 2024

Contents

Task 0: Examine the setup of your own account

- Examine your account by using the command `id` and by looking into the files `/etc/passwd` and `/etc/group` . What is its principal group? What other groups is the account a member of? What is the UID of the account and the GID of the principal group?

Output :

```
$ id
uid=1000(anthony) gid=1000(anthony) groups=1000(anthony),4(adm),24(cdrom),27(sudo),30(dip),46(
    plugdev),
114(lpadmin),125(libvirt),988(sambashare)
```

Answers to questions:

1. **Principal group**: The principal group is `anthony`.
2. **Other groups**: The account belongs to the groups `adm`, `cdrom`, `sudo`, `dip`, `plugdev`, `lpadmin`, `libvirt`, and `sambashare`.
3. **UID of the account**: The UID of the account is `1000`.
4. **GID of the principal group**: The GID of the principal group is `1000`.

- Which skeleton files have been copied?

Output :

```
$ ls -a /etc/skel

.  ..  .bash_logout  .bashrc  .face  .face.icon  .profile
```

Answer to question :

The skeleton files that have been copied from `/etc/skel` are:

- `.bash_logout`
- `.bashrc`
- `.face`
- `.face.icon`
- `.profile`

Task 1: Create user accounts

In this task you will use command-line tools to manage user accounts. Perform the following steps and give in the lab report the commands you used. Use the tools useradd and groupadd .

1. Create the groups `jedi` and `rebels` . Before creating them verify that they do not yet exist.

1. Check that groups do not already exist

Input :

```
$ getent group anthony
anthony:x:1000:
$ getent group jedi
$ getent group rebels
```

To validate the `getent` command, we first test it with a group that we are certain exists.

2. Groups creation

Input :

```
$ sudo groupadd jedi
[sudo] password for anthony:
$ sudo groupadd rebels
[sudo] password for anthony:
```

3. Check that the groups have been created correctly

Input :

```
$ getent group jedi
jedi:x:1001:
$ getent group rebels
rebels:x:1002:
```

This time, the `getent` command returns the group name and its GID.

2. Create the following user accounts with default home directories and login shell (for example account luke should have home directory `/home/luke` and a `bash` shell). Note: For fear of overwriting something the `useradd` tool is very cautious about creating the home directory for an account.

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

   Before creating them verify that they do not yet exist.

   - Account `luke` , assigned to groups `jedi` (principal) and `rebels`.
   - Account `vader` , assigned to group `jedi` (principal).
   - Account `solo` , assigned to group `rebels` (principal).

Questions answers :

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

The `-m` (or `--create-home`) option must be specified for useradd to create a home directory for the user.

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 for users created with `useradd` is `/bin/sh`. To change the default login shell from `/bin/sh` to `/bin/bash`, you can use the `-s` (or `--shell`) option with `useradd`.

Input :

1. Check that the accounts do not exist :

```
$ getent passwd luke
$ getent passwd vader
$ getent passwd solo
```

- `getent`: This command retrieves entries from administrative databases.
- `passwd`: This option specifies that we want to check the password database (users).
- `luke`, `vader`, `solo`: These are the names of the users we are checking. If these users exist, their information will be displayed. Otherwise, the command will return no results.

2. Accounts creation :

```
$ sudo useradd -m -s /bin/bash -g jedi -G rebels luke
$ sudo useradd -m -s /bin/bash -g jedi vader
$ sudo useradd -m -s /bin/bash -g rebels solo
```

- `sudo`: Executes the command with the superuser privileges necessary for creating user accounts.
- `useradd`: The command to add a new user.
- `-m` or `--create-home`: Creates a home directory for the user (e.g., /home/luke).
- `-s /bin/bash` or `--shell /bin/bash`: Sets the default login shell for the user to `/bin/bash`.
- `-g jedi` or `--gid jedi`: Sets the user's primary group to 'jedi'.
- `-G rebels` or `--groups rebels`: Adds the user to additional groups, 'rebels'.
- `luke`, `vader` or 'rebels`: The name of the new user.

3. Set a password for the account `luke`.

Input :

```
$ sudo passwd luke
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
```

- `passwd`: The command to set or change a user's password.

4. Test the account `luke` . Verify that the user can log in and create files. Verify that the user cannot access sensitive system information such as the file `/etc/shadow`.

```
$ su - luke
Mot de passe :
$ touch /home/luke/testfile
$ ls -l /home/luke/testfile
-rw-r--r-- 1 luke jedi 0 mai   15 23:39 /home/luke/testfile
$ cat /etc/shadow
cat: /etc/shadow: Permission non accordée
```

- `su - luke`: Switches the user to luke and starts a login session with his environment.
- `touch /home/luke/testfile`: Creates an empty file named `testfile` in luke's home directory
- `ls -l /home/luke/testfile`: Displays the details of the `testfile`, thus verifying its creation.
- `cat /etc/shadow`: Attempts to display the contents of the `/etc/shadow` file, which should fail for a non-privileged user.

5. Use `su` to change your account to that of `vader`. Test if the user vader has access to the files in the home directory of user `luke` .

Input :

```
$ sudo su - vader
vader@anthoony-1-2:~$ ls -l /home/luke
total 0
-rw-r--r-- 1 luke jedi 0 mai   15 23:39 testfile
```

Vader has access to Luke's home folder.

Task 2: Change group membership

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

First, verify that `leia` does not already exist:

```
getent passwd leia
```

Let's create `leia` without specifying a principal group:

```
sudo useradd -m -s /bin/bash leia
```

`leia` will be assigned to the default group, typically one with the same name as the username or the default group defined in the system settings. Let's check the principal group assigned to `leia` with `id leia`:

```
uid=1004(leia) gid=1005(leia) groups=1005(leia)
```

2. Make `leia` a Member of the Group `rebels` (as a Secondary Group).

```
sudo usermod -aG rebels leia
```

`-aG`: append the user to the specified supplementary groups.

Verify that `leia` has been added to `rebels` with `id leia`:

```
uid=1004(leia) gid=1005(leia) groups=1005(leia),1004(rebels)
```

3. Make `leia` Leave the Group `rebels` and Join the Group `jedi` Instead:

Remove `leia` from `rebels`:

```
sudo gpasswd -d leia rebels
```

output:

```
Removing user leia from group rebels
```

Add `leia` to `jedi`:

```
sudo usermod -aG jedi leia
```

Confirm that `leia` is now in the `jedi` group and not in `rebels` with `id leia`:

```
uid=1004(leia) gid=1005(leia) groups=1005(leia),1003(jedi)
```

4. Make `leia` Leave Any Secondary Group:

This command sets the user's group list to be empty, meaning `leia` will only be in her primary group:

```
sudo usermod -G "" leia
```

Verify this with `id leia`:

```
uid=1004(leia) gid=1005(leia) groups=1005(leia)
```

Task 3: Give a user sudo rights

To give a user access to `sudo` one must normally manually edit the file `/etc/sudoers` by using the `visudo` command and list all the users there. In many Linux distributions (among them Ubuntu) though touching the file is not necessary. Out of the box the file `/etc/sudoers` is configured to give sudo access to all users that are members of the group named `sudo` . Instead of modifying the `/etc/sudoers` file, one can simply make users members of the `sudo` group.

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

From `/etc/sudoers`:

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

This line means that any user in the `sudo` group can execute any command from any terminal, acting as any user.

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).

The line should be modified to include `NOPASSWD:`:

```
%sudo   ALL=(ALL:ALL) NOPASSWD:ALL
```

Perform the following steps and give in the lab report the commands you used.

1. Give the account `luke` sudo rights.

To give `luke` `sudo` rights, we can add him to the `sudo` group:

```
sudo usermod -aG sudo luke
```

2. Test the new rights. Verify that `luke` can read the file /etc/shadow using `sudo`.

1. Switch to `luke`:

```
sudo su - luke
```

2. Test `sudo` access:

```
sudo cat /etc/shadow
```

output:

```
root:!:19760:0:99999:7:::
daemon:*:19576:0:99999:7:::
# ( ... )
luke:$y$j9T$Bvsrz4hYTyCbBtHfQyuT9.$NSMCTpEfvqJd2YbFHIBD2Uhaou0YAUUdO7xvrz.0Jz5
    :19868:0:99999:7:::
vader:!:19868:0:99999:7:::
solo:!:19868:0:99999:7:::
leia:!:19868:0:99999:7:::
```

3. Remove sudo rights from the account `luke`.

Let's remove him from the `sudo` group:

```
sudo gpasswd -d luke sudo
```

And verify with `id luke`:

```
uid=1001(luke) gid=1003(jedi) groups=1003(jedi),1004(rebels)
```

Task 4 : Remove a user account

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

1. Remove the account `leia`, but do not delete the home directory yet.

We can use the `userdel` command to ensure the home directory is not removed. This will delete `leia` from the system but will not remove her home directory located at `/home/leia`:

```
sudo userdel leia
```

Let's check the deletion with `id leia`:

```
id: ‘’leia: no such user
```

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

```
sudo ls -la /home/leia
```

output:

```
total 40
drwxr-x--- 3 1004 1005  4096 Mai 25 17:05 .
drwxr-xr-x 8 root root  4096 Mai 25 17:05 ..
-rw-r--r-- 1 1004 1005   220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 1004 1005  3771 Jan  6  2022 .bashrc
drwxr-xr-x 2 1004 1005  4096 Aug  7  2023 .config
-rw-r--r-- 1 1004 1005 14965 Apr 29  2022 .face
lrwxrwxrwx 1 1004 1005     5 Feb  7 11:11 .face.icon -> .face
-rw-r--r-- 1 1004 1005   807 Jan  6  2022 .profile
```

Within `/home/leia` we now see the `UID` and `GID` of a username and group name, because the system no longer recognizes the UID and GID associated with the removed user. If we compare with a user that still exists with `ls -la /home/luke`, we would see the names and not the IDs:

```
total 40
drwxr-x--- 3 luke jedi  4096 Mai 25 17:30 .
drwxr-xr-x 8 root root  4096 Mai 25 17:05 ..
-rw-r--r-- 1 luke jedi   220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 luke jedi  3771 Jan  6  2022 .bashrc
drwxr-xr-x 2 luke jedi  4096 Aug  7  2023 .config
-rw-r--r-- 1 luke jedi 14965 Apr 29  2022 .face
lrwxrwxrwx 1 luke jedi     5 Feb  7 11:11 .face.icon -> .face
-rw-r--r-- 1 luke jedi   807 Jan  6  2022 .profile
-rw-r--r-- 1 luke jedi     0 Mai 25 17:30 .sudo_as_admin_successful
-rw-r--r-- 1 luke jedi     0 Mai 25 17:01 testfile.txt
```

3. 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?

Leia had the UID 1004, so if she has created files elsewhere on the system, we could find them by scanning for items with her UID with `sudo find / -user 1004`. This command searches the entire filesystem for files owned by the user with UID 1004:

```
find: ‘/run/user/1000/’doc: Permission denied
/home/leia
/home/leia/.config
/home/leia/.config/korgacrc
/home/leia/.face.icon
/home/leia/.face
/home/leia/.bashrc
/home/leia/.bash_logout
/home/leia/.profile
find: ‘/proc’/10843: No such file or directory
find: ‘/proc/11000/task/11000/fd’/6: No such file or directory
find: ‘/proc/11000/task/11000/fdinfo’/6: No such file or directory
find: ‘/proc/11000/fd’/5: No such file or directory
find: ‘/proc/11000/fdinfo’/5: No such file or directory
```

This command searches the entire filesystem for files owned by the user with UID 1002. If you don't know the UID, and it's not practical to look it up since the user has been deleted, scanning might be challenging unless you have logs or other references.

4. Remove the home directory manually.

To remove `leia`'s home directory manually, we can use the `rm` command:

```
sudo rm -r /home/leia
```

And finally check the result with `sudo ls -la /home/leia`:

```
ls: cannot access '/home/leia': No such file or directory
```